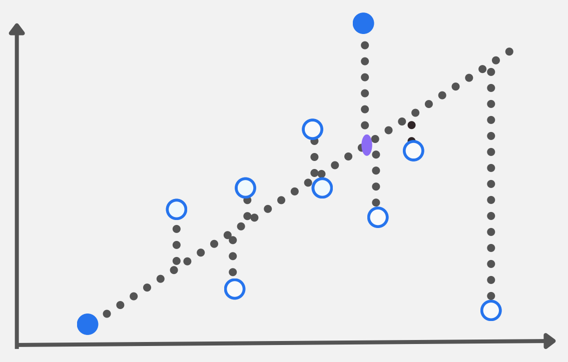
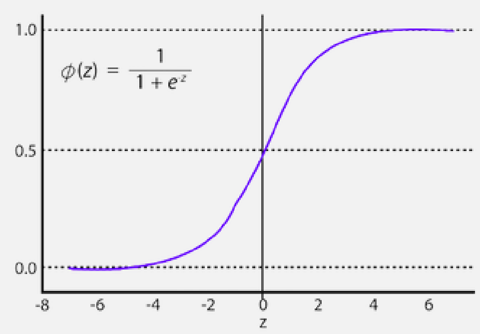
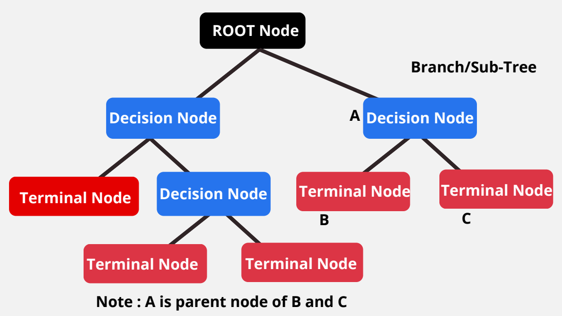
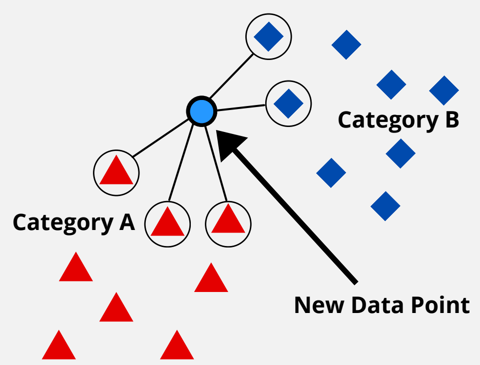
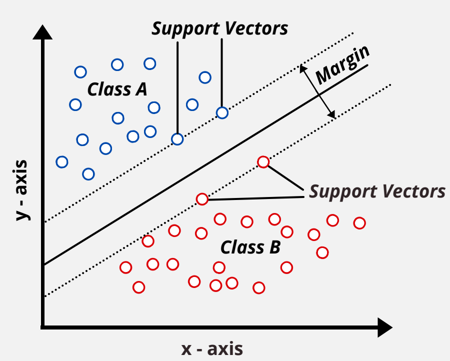
Hyperparameter Tuning in Machine Learning

Representation	Algorithm Name	Hyperparameter
	Linear Regression	Regularization parameter (alpha for Ridge/Lasso regression)
	Logistic Regression	C (Inverse of regularization strength), penalty (L1, L2)
	Decision Tree	Max_depth, min_samples_split, min_samples_leaf, criterion
	K-Nearest Neighbors	n_neighbors, weights, metric
	Support Vector Machines	C, kernel, gamma, degree (for polynomial kernel)

What is Hyperparameter Tuning

- Hyperparameter tuning is an essential process in machine learning that involves adjusting the parameters of an algorithm that are not learned from the data but are set prior to the training process.

- These parameters, known as hyperparameters, govern the overall behavior of a machine learning model.

- Here's why hyperparameter tuning is useful and when to use it:

Why Hyperparameter Tuning Is Useful

- **Performance Improvement**: Proper tuning can significantly improve model performance. It can help in finding the right balance between underfitting and overfitting.
- **Model Generalization**: Tuned models are often more generalizable to unseen data.
- **Algorithm Optimization**: Different algorithms have different hyperparameters, and tuning them can help in harnessing the full potential of the algorithm.
- **Resource Efficiency**: By optimizing hyperparameters, one can achieve better results, often with less computational resources.

When to Use Hyperparameter Tuning

- After selecting an appropriate algorithm for the problem.
- When the default hyperparameters do not yield satisfactory results.
- As part of a systematic model development process to ensure robustness.
- When there is sufficient data to validate the impact of different hyperparameters.

Analytics Vidhya

Without Hyperparameter Tuning

- Models may underfit or overfit the data.
- Model performance may not be optimal.
- The algorithm may not work as efficiently as it could.
- The model might not generalize well to new, unseen data.

With Hyperparameter Tuning

- Model accuracy typically improves.
- The risk of overfitting or underfitting is reduced.
- The model is better tailored to the specifics of the problem and data.
- Computational resources can be used more effectively.

Hyperparameter Tuning in Algorithms

Linear Regression:

- Common hyperparameters include regularization parameters in ridge and lasso regression. Tuning these can control the complexity of the model and prevent overfitting.

Logistic Regression:

- Similar to linear regression, it involves tuning regularization parameters. Additionally, the decision threshold can be tuned for classification problems to balance precision and recall.

Decision Tree:

- Hyperparameters like the depth of the tree, minimum number of samples required to split a node, and minimum number of samples required at a leaf node can be tuned to prevent the tree from being too complex or too simple.

K-Nearest Neighbor (KNN):

- The number of neighbors (k) is a critical hyperparameter that can be tuned. Also, the distance metric (like Euclidean or Manhattan) and the weights given to neighbors can be optimized.

Support Vector Machines (SVM):

Key hyperparameters include the penalty parameter C, the kernel type (linear, polynomial, RBF, etc.), and the kernel's specific parameters (like degree of the polynomial, gamma in the RBF kernel).

Tuning Methods

- **Grid Search**: Exhaustive search over a specified parameter grid.
- **Random Search**: Randomized search over parameters, which is often faster and can yield good results.

- **Bayesian Optimization**: Uses a probabilistic model to guide the search for the best hyperparameters.

- **Gradient-based Optimization**: For differentiable hyperparameters, this method can be used.

- **Evolutionary Algorithms**: Inspired by natural evolution, these algorithms search the hyperparameter space iteratively.