**Basic Machine Learning Algorithm, For in Plant Breeding**

**Machine learning** is a subset of artificial intelligence that focuses on developing algorithms and models that allow computers to learn from data and make predictions or decisions without being explicitly programmed. It involves the use of statistical techniques to enable machines to improve their performance on a specific task over time.

**supervised learning**, the algorithm is trained on a labeled dataset, where each input is associated with a corresponding output or target. The model learns to map the input data to the correct output by generalizing from the labeled examples. **Unsupervised learning**, the algorithm is given unlabeled data and must find patterns or structures within it without explicit guidance. Clustering and dimensionality reduction are common tasks in unsupervised learning.

Machine learning has the potential to revolutionize plant breeding by helping breeders make more informed decisions and accelerate the breeding process. Here are some ways in which machine learning is relevant to plant breeding:

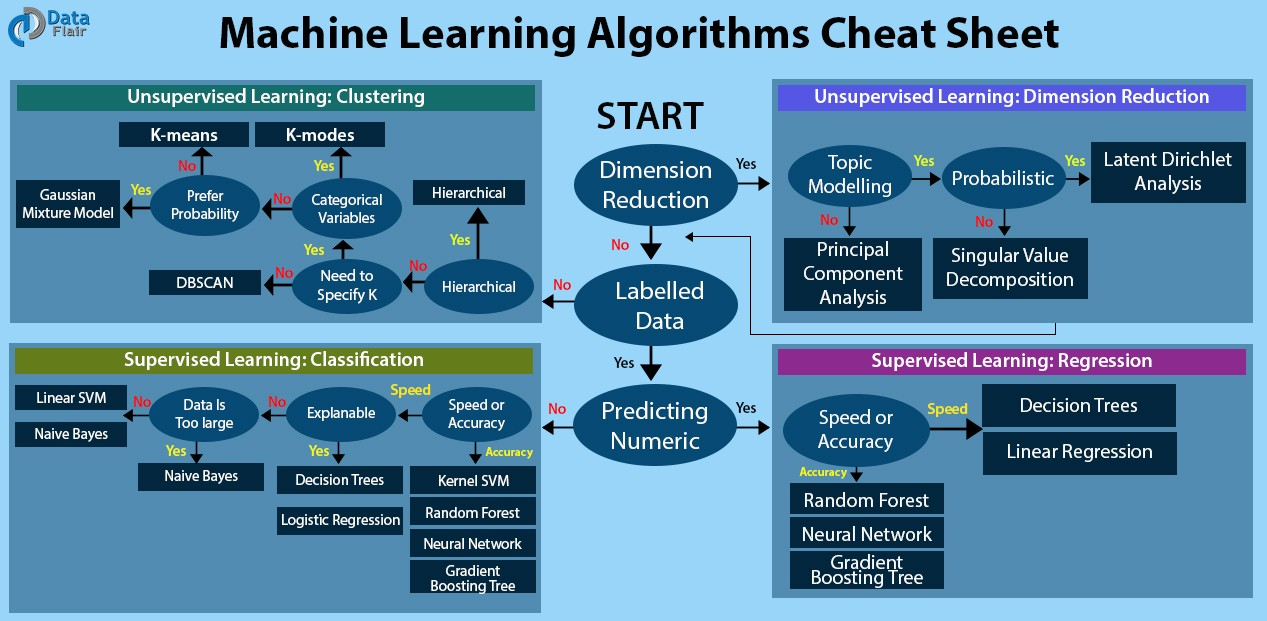
**Genomic Selection:** Machine learning models can analyze genomic data to predict the performance of plants based on their genetic makeup. This enables breeders to select plants with desired traits more efficiently.

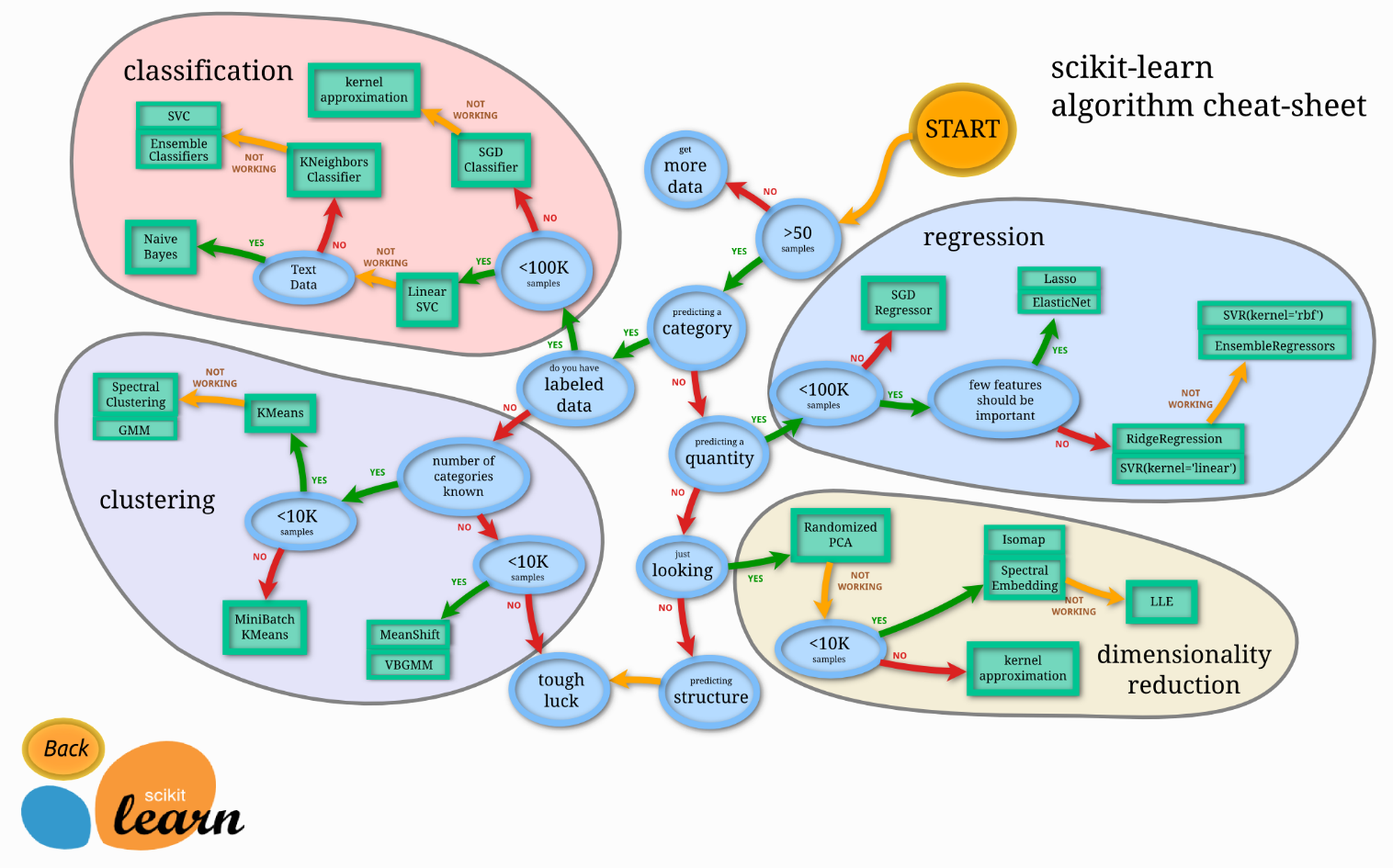
**Phenotype Prediction:** Machine learning can be used to predict plant phenotypes (observable traits) based on genetic information, environmental factors, and other variables.This helps in identifying plants with desirable characteristics early in the breeding process.

**Crop Monitoring:** Machine learning algorithms can analyze remote sensing data to monitor crop health, detect diseases, and assess environmental conditions.This information aids breeders in making decisions to improve crop yield and resilience.

**Data-Driven Decision Making**: By leveraging historical breeding data, machine learning can assist breeders in making data-driven decisions on which plant varieties to cross or select for breeding programs.

|  |  |  |  |
| --- | --- | --- | --- |
| **Supervised** | | **Unsupervised** | |
| **Regression** | **Classification** | **Clustering** | **Dimension Reduction** |
| 1. Linear Regression | 1. Logistic Regression | 1. KNN | 1. PCA |
| 2. Multivarte linear regression | 2. KNN | 2. K means |  |
| 3. Decision trees | 3. Navies Bayes | 3. Hierarchical |  |
| 4. Random Forest | 4. SVM |  |  |
| 5. Neural Networks | 5. Decision trees |  |  |
|  | 6. Random Forest |  |  |
|  | 7. Neural Networks |  |  |





**Steps in development of Machine Learning model**

1. **Define the Problem:** Clearly articulate the problem you are trying to solve and define the goal of your machine learning model.
2. **Collect and Prepare Data:** Gather relevant data for your problem. This may involve collecting new data, using existing datasets, or a combination of both. Clean and preprocess the data to handle missing values, outliers, and ensure it is in a suitable format for analysis.
3. **Exploratory Data Analysis (EDA):** Perform exploratory data analysis to understand the characteristics of the data. Visualize data distributions, correlations, and relationships between variables to gain insights.
4. **Feature Engineering:** Select and engineer features that are relevant to the problem. Transform and scale features to improve model performance.
5. **Split the Data:** Divide the dataset into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune hyperparameters, and the test set is used to evaluate the final model's performance.
6. **Choose a Model:** Select a machine learning model or algorithm based on the nature of the problem (classification, regression, clustering, etc.) and the characteristics of the data.
7. **Train the Model:** Train the model using the training dataset. Adjust model parameters to minimize the difference between predicted and actual outcomes.
8. **Validate and Tune Hyperparameters:** Evaluate the model's performance on the validation set.

Fine-tune hyperparameters to optimize the model's performance. This may involve techniques like grid search or random search.

1. **Evaluate on Test Set:** Assess the final model's performance on the test set to ensure it generalizes well to new, unseen data.
2. **Interpret the Model:** Understand the model's decision-making process. For example, interpret feature importance or coefficients in linear models.
3. **Deploy the Model:** Integrate the model into a production environment if it meets the performance requirements. Implement monitoring and maintenance procedures for ongoing model performance.
4. **Iterate and Improve:** Based on feedback, new data, or changes in the problem, iterate on the model to improve performance.

Continuous monitoring and improvement are crucial for maintaining the model's relevance over time.

**Explanatory data Analysis**

**Univariate Bivariate Multivariate Analysis**

Univariate: only one feature will be studied; histogram, classify the data (e.g. tall short medium)

Bivariate: At least two features are studied. Ex: height vs weight

Multivariate: more than two variables/features are studied Ex: height, weight, age, sex, DOB

**Z-Score statistics**

The formulae used to convert normal distribution to standard normal distribution

Standard normal distribution (mean = 0 SD =1)

**Bias and Variance**

Linear regression is a model gives the best fit line

* *Underfitting*: The error is high for the trained data as well as test data.
* *Overfitting*: The accuracy for the training data is high and for the test data low accuracy
* *Bias:* The error of the training data
* *Variance*: The error of the test data

*In underfitting high bias and high variance*

*In overfitting low bias and high variance*

For Classification

|  |  |  |
| --- | --- | --- |
| **Model 1** | **Model 2** | **Model 3** |
| Training error 1% | Training error 25% | Training error >10% |
| Test error 20% | Test error 25% | Test error >10% |
| Overfitting | Under fitting | Good model |

In general model should be low bias and low variance

**What is hyper parameter tuning**

Hyperparameters are configuration settings for a model that are not learned from the data but are set prior to the training process. They are external to the model and are used to guide the learning process. Examples of hyperparameters include the learning rate, the number of hidden layers in a neural network, the number of trees in a random forest, and regularization parameters.

Unlike model parameters, which are learned from the training data during the training process (e.g., coefficients in linear regression), hyperparameters are set by the data scientist or machine learning engineer and play a crucial role in determining the model's performance.

Hyperparameter tuning, also known as hyperparameter optimization, refers to the process of finding the best set of hyperparameters for a machine learning model to achieve optimal performance on a specific task or dataset. The goal is to systematically search through a hyperparameter space to identify the combination of hyperparameters that results in the highest model performance.

There are several techniques for hyperparameter tuning:

**Grid Search:** A predefined set of hyperparameter values is specified, and the model is trained and evaluated for each combination of hyperparameters in the grid.This method is simple but can be computationally expensive, especially for large hyperparameter spaces.

**Random Search:** Random combinations of hyperparameters are sampled from a predefined space. While less exhaustive than grid search, random search often finds good hyperparameter combinations more efficiently.

**Different Machine Learning model/Algorithms**

1. **Linear Regression:**

Linear regression is a supervised learning algorithm used for predicting a continuous outcome variable (dependent variable) based on one or more predictor variables (independent variables). The goal of linear regression is to find the best-fit line that minimizes the difference between the predicted values and the actual values. The equation for a simple linear regression with one independent variable is typically represented as:

*Best fit line* y= mx +c m= slope c = intercept

Gradient descent: The cost function is calculated for different *m* values and plotted against a plot in cost function vs m values.

We need to get the global minima to obtain the better fit line by convergence theorem.

Ridge regression is used to overcome overfitting (Low bias and high variance)

1. **Ridge regression**

Cost functions + lambda × (slope)2 : The steeper slope is penalised to reduce the overfitting

Lamba value can be any positive value

1. **Lasso Regression**

Cost functions + lambda × |slope| (magnitude of slope = m1+m2+m3+…..)

Lasso also help in feature selection, the feature where the slope values very less that feature will be removed

1. **Multivariate Linear Regression**

Multivariate linear regression is an extension of simple linear regression that involves predicting a dependent variable (target) based on multiple independent variables (features). In the context of multivariate linear regression, the model assumes a linear relationship between the dependent variable and multiple independent variables. The goal is to find the best-fit hyperplane that minimizes the difference between the predicted values and the actual values.

Multivariate linear regression is particularly useful when dealing with datasets where the dependent variable is influenced by multiple factors. It allows for a more nuanced understanding of how each independent variable contributes to the overall prediction.

**3. Logistic Regression**

Logistic Regression is a statistical method and a popular machine learning algorithm used for binary classification tasks. Despite its name, logistic regression is used for classification, not regression. It models the probability that an instance belongs to a particular class. The algorithm is well-suited for problems where the dependent variable is categorical with two possible outcomes, such as spam or not spam, fraudulent or non-fraudulent, or whether a customer will buy a product (yes/no).

In logistic regression, the logistic function (also known as the sigmoid function) is used to transform a linear combination of input features into a value between 0 and 1. The logistic regression model predicts the probability that an instance belongs to the positive class (class 1), and a threshold is applied to classify the instance into one of the two classes based on this probability.

Logistic regression is favoured for its simplicity, interpretability, and efficiency. It provides insights into the impact of each feature on the probability of the positive class. However, it assumes a linear relationship between the features and the log-odds of the outcome, which may limit its effectiveness in capturing complex relationships in the data.

Logistic regression mainly used binary classification

**4. Decision trees**

Decision Trees are a versatile and intuitive machine learning algorithm used for both classification and regression tasks. The algorithm makes decisions by recursively partitioning the data into subsets based on feature values, creating a tree-like structure. Each internal node in the tree represents a decision based on a specific feature, and each leaf node corresponds to a predicted outcome.

**General overview of how Decision Trees work:**

**Root Node**: The feature that provides the best split is chosen as the root node. The "best split" is determined based on criteria such as Gini impurity for classification or mean squared error for regression.

**Internal Nodes**: The data is split at each internal node based on the chosen feature, creating branches for different values of that feature. The process continues recursively for each subset.

**Leaf Nodes**: The process stops when a predefined stopping criterion is met or when further splits do not significantly improve the model. At this point, the final nodes are called leaf nodes, and each leaf node represents a predicted outcome.

**Prediction:** To make predictions for a new instance, it traverses the tree from the root to a leaf node, following the decision rules at each node.

**5. Random Forest**

Random Forest is an ensemble learning algorithm that operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It belongs to the family of bagging methods (Bootstrap Aggregating).

Here's how Random Forest works:

**Bootstrapping**: Random Forest builds multiple decision trees by bootstrapping (sampling with replacement) from the original dataset. Each tree is trained on a different subset of the data.

**Random Feature Selection**: For each tree, a random subset of features is selected for splitting at each node. This introduces diversity among the trees and prevents them from being highly correlated.

**Tree Building:** Each decision tree is built using the selected subset of data and features. The trees are grown until a certain condition is met (e.g., reaching a maximum depth or minimum number of samples per leaf).

**Voting or Averaging**: For classification tasks, each tree "votes" for a class, and the class with the most votes becomes the predicted class. For regression tasks, the predictions of individual trees are averaged to obtain the final prediction.

**6. Neural Networks**

Neural Networks, particularly artificial neural networks, are a class of supervised machine learning models inspired by the structure and functioning of the human brain. They are powerful models capable of learning complex relationships and patterns in data. Neural networks have gained significant popularity in various applications, including image and speech recognition, natural language processing, and many others.

**Brief overview of Neural Networks in the context of supervised machine learning:**

**Architecture:** Neural networks consist of layers of interconnected nodes, also called neurons or units. The basic building blocks include an input layer, one or more hidden layers, and an output layer.The connections between nodes have weights, which are parameters learned during the training process.

**Activation Function:** Each node applies an activation function to the weighted sum of its inputs. Common activation functions include the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU).

**Feedforward Propagation:** During training, input data pass through the network in a process called feedforward propagation. The output of the network is compared to the actual target values, and the difference is used to compute a loss or cost function.

**Backpropagation:** The backpropagation algorithm is employed to minimize the loss by adjusting the weights of the connections through gradient descent or a similar optimization algorithm. This involves computing the gradient of the loss with respect to the weights and updating the weights accordingly.

**Training:** The training process iteratively adjusts the network's parameters to reduce the difference between predicted and actual outcomes. This involves multiple epochs (passes through the entire dataset) until convergence or a predefined stopping criterion.

**Types of Neural Networks:**There are various architectures of neural networks designed for specific tasks. For example:

Feedforward Neural Networks (FNN): Standard architecture with information flowing in one direction from input to output.

Convolutional Neural Networks (CNN): Especially effective for image processing, utilizing convolutional layers.

Recurrent Neural Networks (RNN): Suitable for sequential data, such as time series or natural language, incorporating recurrent connections.

Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU): Specialized RNN architectures addressing the vanishing gradient problem.

**Supervised Learning:** Neural networks are commonly used for supervised learning tasks, where the model is trained on labeled datasets, and the goal is to predict the target variable based on input features.

**What is ensemble learning:**

Ensemble learning is a machine learning paradigm where multiple models (learners) are combined to improve overall predictive performance and robustness. The idea is that by aggregating the predictions of multiple models, the ensemble can often achieve better results than any individual model. Ensemble methods are commonly used in both classification and regression tasks. Some popular ensemble learning algorithms include:

**Bagging (Bootstrap Aggregating)**:

Example Algorithm: Random Forest

Description: Bagging involves training multiple instances of the same learning algorithm on different subsets of the training data (sampled with replacement) and then averaging the predictions (for regression) or taking a majority vote (for classification) to make the final prediction.

**Boosting**:

Example Algorithms: AdaBoost, Gradient Boosting (e.g., XGBoost, LightGBM)

Description: Boosting focuses on sequentially training models, where each subsequent model corrects the errors made by the previous ones. It assigns higher weights to misclassified instances, so subsequent models pay more attention to them. The final prediction is a weighted sum of the individual models.

**Stacking**:

Description: Stacking involves training multiple models and combining their predictions through another model called a meta-model. The meta-model learns to weigh the predictions of the base models based on their performance. Stacking can be more complex than bagging and boosting but has the potential for improved performance.

**Voting Classifiers/Reggressors**:

Description: This involves combining the predictions of multiple models and making a final prediction based on a "voting" mechanism. In a majority vote for classification or an average for regression, the final prediction is determined by the consensus of the base models.

Ensemble methods are powerful tools in machine learning because they can reduce overfitting, enhance generalization, and provide more robust predictions, especially when individual models may have limitations or make errors in specific scenarios. The diversity among the base models is key to the success of ensemble learning, as it allows for complementary strengths and weaknesses.

Ensemble learning is widely used in practice and has contributed to the success of various machine learning applications, including image recognition, natural language processing, and predictive modeling. Each ensemble method has its strengths and is suitable for different types of problems, so the choice of method often depends on the characteristics of the data and the specific task at hand.

**7. Logistic Regression**

**8. KNN**

**9. SVM**

**10. Navie**

**11. KNN**

**12. K means**

**13. Hierarchical s Bayes**

**14. PCA**