Random Forest

DATARANCH.org
VISUALIZE | ANALYZE | CAPITALIZE

What is a Random Forest?

Random forest is a commonly-used machine learning algorithm used for both classification and regression tasks. It combines the output of multiple decision trees to reach a single result.

Random forest is a flexible, easy-to-use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also less prone to overfitting compared to individual decision trees.

Key Terms

Ensemble learning: Ensemble learning refers to the technique of combining multiple models to improve overall performance. Random Forest is an ensemble learning method that combines multiple decision trees.

Hyperparameters: Random Forest has several hyperparameters that can be tuned to optimize its performance, such as the number of decision trees in the ensemble, the maximum depth of each tree, and the number of features considered at each split.

Key Terms

Bagging: Bagging stands for Bootstrap Aggregating. It is a technique used in Random Forest where subsets of the training data are randomly sampled (with replacement) to train each decision tree.

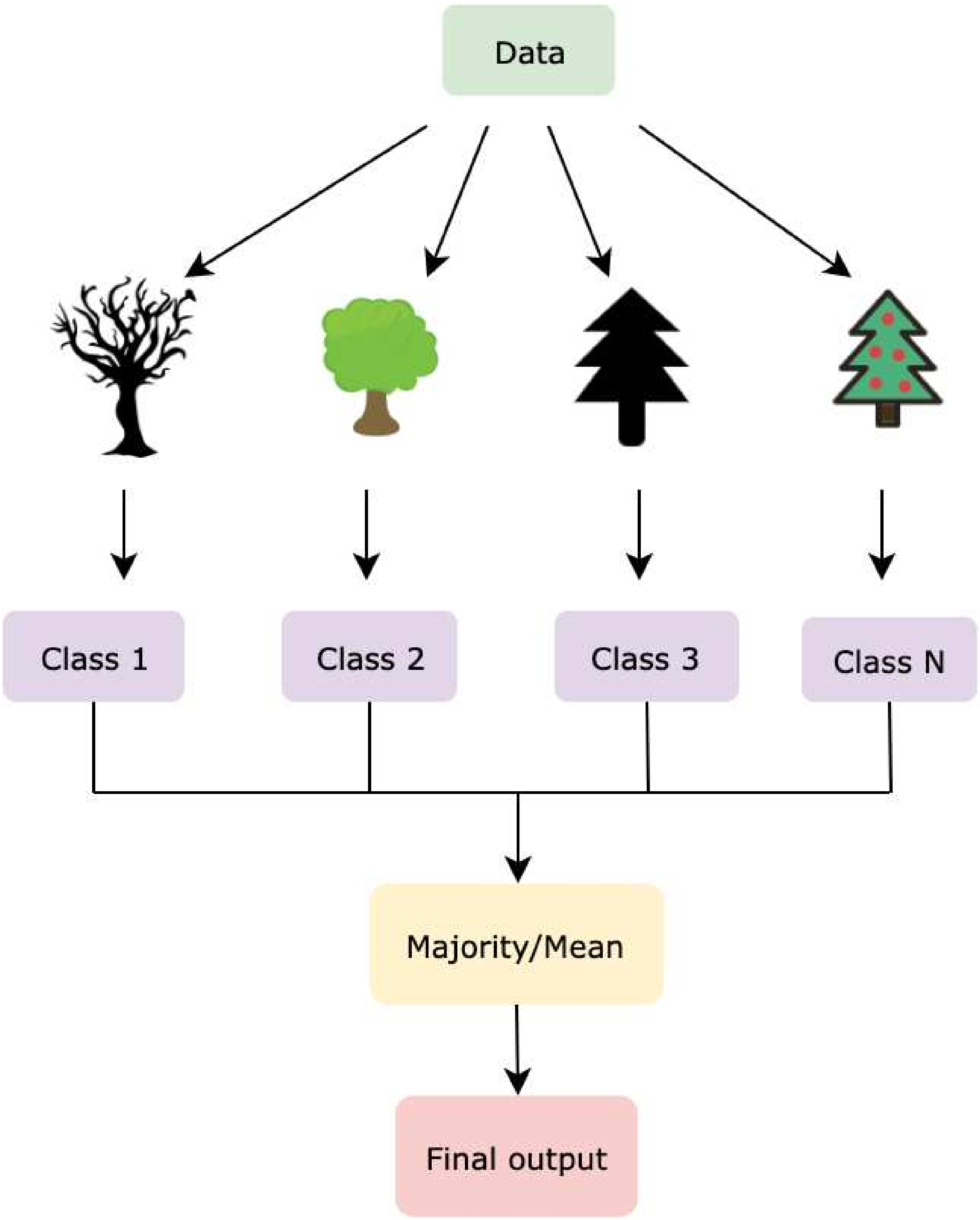
Out-of-sample error: This refers to the error rate or mean squared error of the model when making predictions on unseen data. It is a measure of how well the model generalizes to new data and is used to evaluate the performance of Random Forest models.

Key Terms

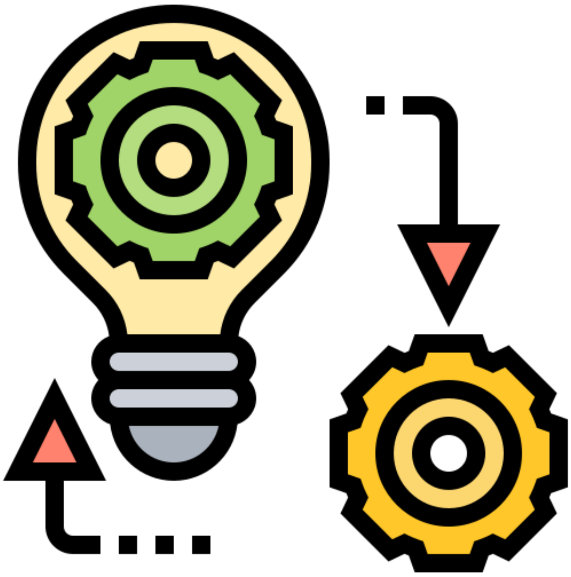
Out-of-bag (OOB) samples: During the training process, some samples in the training set are not included in the bootstrap samples used to train a particular decision tree. These "out-of-bag" samples can be used to estimate the model's performance without the need for a separate validation set.

Voting: Random Forest combines the predictions of all decision trees in the ensemble to make a final prediction. For classification tasks, it uses majority voting, and for regression tasks, it uses averaging.

Example


Data
Class 1
Class 2
Class 3
Class N
Majority/Mean
Final output

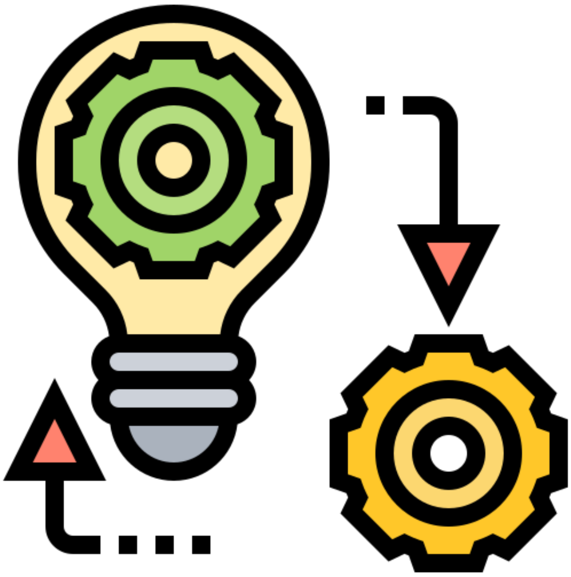
How Random Forest work?



Step 1. Data preparation: Random Forest requires a dataset with input features and corresponding target variables. The data is typically split into a training set and a test set.

Step 2. Random sampling: Random Forest performs random sampling with replacement from the training set to create multiple subsets, known as bootstrap samples. Each subset contains a random selection of data points, allowing some instances to be repeated while others may be left out.

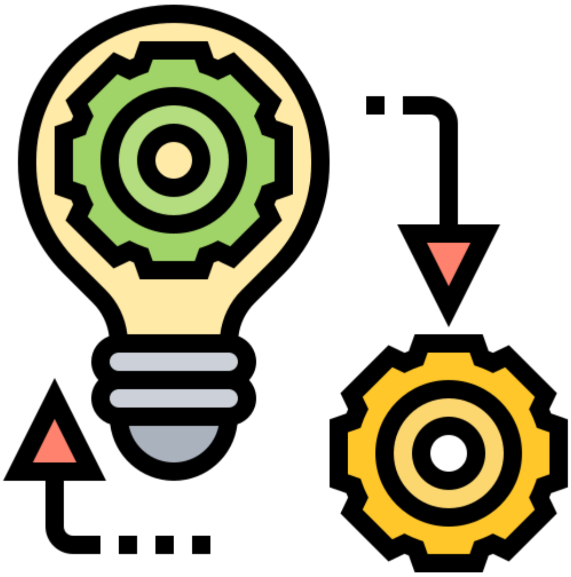
How Random Forest work?



Step 3. Decision tree construction: For each bootstrap sample, a decision tree is constructed. The decision tree is built using a process called recursive partitioning, where the data is split based on feature values to create branches that maximize the separation of the target variables.

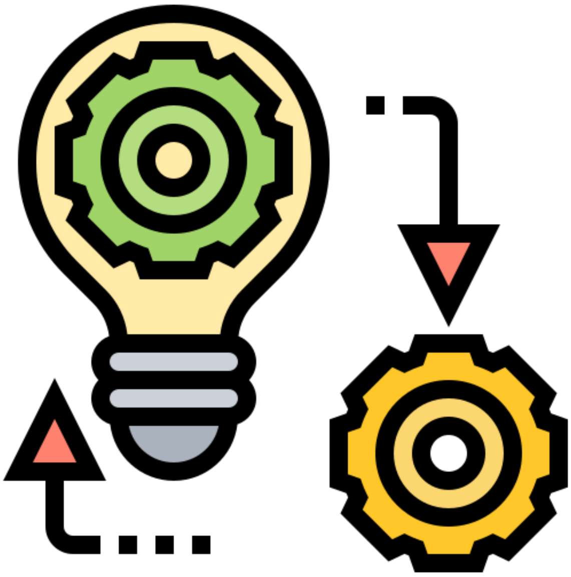
Step 4. Feature selection: At each node of the decision tree, a random subset of features is considered for splitting. This random feature selection introduces diversity among the decision trees and helps to reduce correlation.

How Random Forest work?



Step 5. Voting for predictions: When making predictions, Random Forest combines the outputs of all decision trees. For classification tasks, it uses majority voting, where the predicted class with the highest number of votes is selected. For regression tasks, it averages the predicted values from all decision trees.

How Random Forest work?



Step 6. Hyperparameter tuning: Random Forest has several hyperparameters that can be adjusted to optimize performance, such as the number of decision trees in the ensemble, the maximum depth of each tree, and the number of features considered at each split. These hyperparameters can be fine-tuned using techniques like cross-validation.

Advantages of Random Forest

1. Accuracy: Random Forest generally produces highly accurate predictions. By combining multiple decision trees and aggregating their outputs, it reduces the impact of individual tree errors and achieves robust results.

2. Robustness to outliers and noise: Random Forest is resilient to outliers and noisy data. It combines the predictions of multiple decision trees, the impact of outliers is dampened, resulting in more robust predictions.

3. Handling of high-dimensional data: Random Forest can effectively handle datasets with a large number of input features. It automatically selects subsets of features at each node, allowing it to handle high-dimensional data without much manual feature engineering.

Advantages of Random Forest

4. Handling of missing data: Random Forest can handle missing values in the dataset. It uses the available features to make predictions and does not require imputation or removal of instances with missing values.

5. Resistance to overfitting: Random Forest is less prone to overfitting compared to individual decision trees. The ensemble averaging of multiple trees helps to reduce variance and generalize well to unseen data. It achieves a good balance between bias and variance, leading to more reliable predictions.

Advantages of Random Forest

6. Out-of-bag evaluation: Random Forest provides an internal estimate of the model's performance using out-of-bag samples. This allows for a quick assessment of the model's accuracy without the need for a separate validation set.

7. Versatility: Random Forest can be applied to both classification and regression tasks. It can handle categorical and numerical features, as well as a mix of both, without requiring extensive data preprocessing.

Disadvantages of Random Forest

1. Interpretability: Random Forest, as an ensemble method, can be less interpretable compared to individual decision trees. It may be challenging to understand the specific decision-making process behind the ensemble's predictions, especially when dealing with a large number of trees.

2. Computational complexity: Random Forest can be computationally intensive, especially when dealing with a large number of decision trees or high-dimensional datasets. The training and prediction times can be longer compared to simpler algorithms, particularly if the dataset is large.

Disadvantages of Random Forest

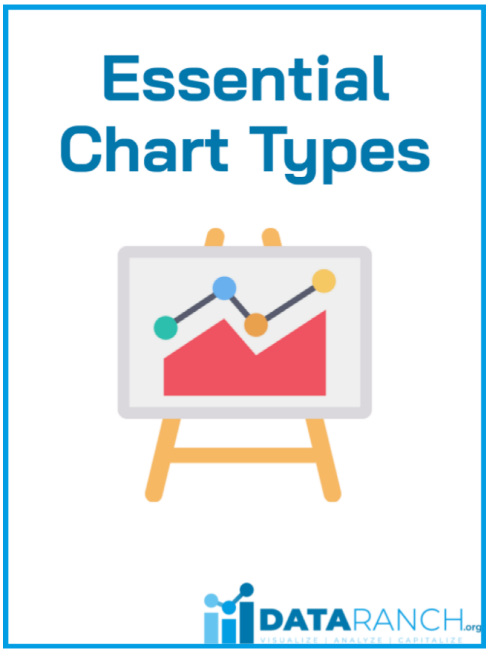
3. Overfitting in noisy datasets: While Random Forest is generally robust to noise and outliers, it can still be susceptible to overfitting in extremely noisy datasets. If the noise dominates the signal, the ensemble of decision trees may still capture and amplify the noise, leading to reduced performance.

4. Sensitivity to correlated features: Random Forest may struggle to differentiate the importance of highly correlated features. In such cases, the algorithm might assign similar levels of importance to correlated features, reducing the effectiveness of feature selection.

Follow **#DataRanch** on LinkedIn for more...

Data Analysis Steps

DATARANCH.org



Essential Chart Types

DATARANCH.org

Data Cleaning Steps

DATARANCH.org

Common data fallacies to watch out for...

DATARANCH.org

Data Wrangling Steps

DATARANCH.org

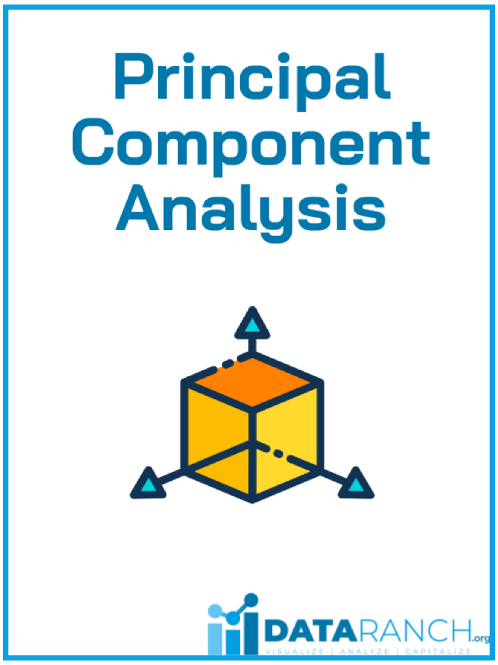
DATARANCH.org

VISUALIZE | ANALYZE | CAPITALIZE

Follow **#DataRanch** on LinkedIn for more...

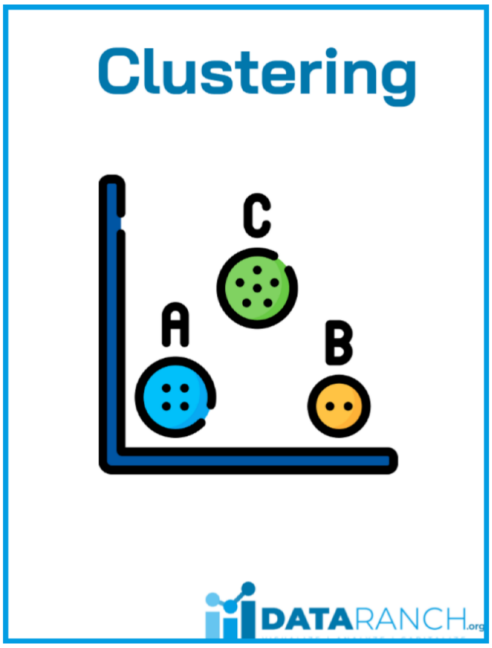
What is Unsupervised Learning?

DATARANCH.org



Principal Component Analysis

DATARANCH.org



Clustering

DATARANCH.org

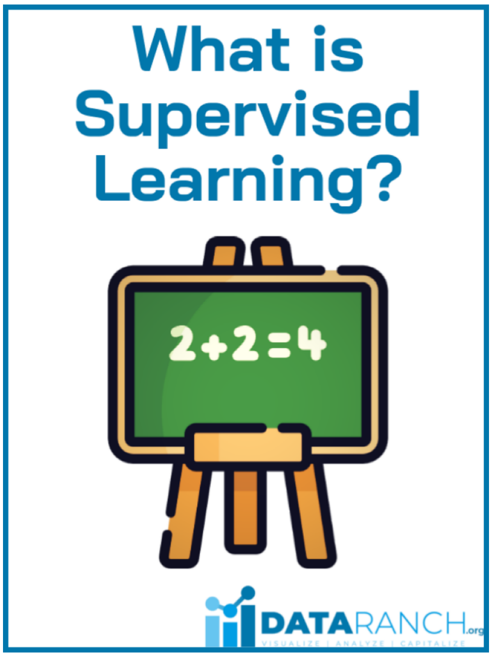
t-Distributed Stochastic Neighbour Embedding (t-SNE)

DATARANCH.org

DATARANCH.org

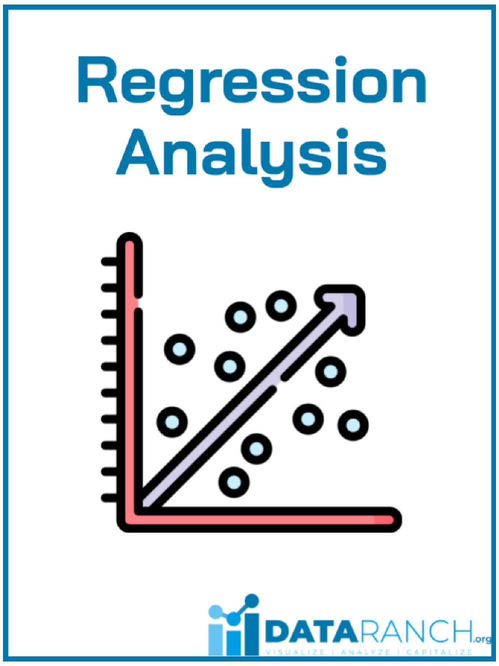
VISUALIZE | ANALYZE | CAPITALIZE

Follow **#DataRanch** on LinkedIn for more...



What is Supervised Learning?

Logistic Regression



Regression Analysis

Decision Trees

DATARANCH.org
VISUALIZE | ANALYZE | CAPITALIZE

DATARANCH.org

VISUALIZE | ANALYZE | CAPITALIZE

✉ info@dataranch.org

in linkedin.com/company/dataranch