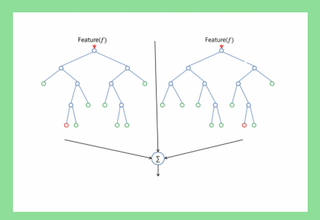
Digit Recognition using RANDOM FOREST Classifier

1. Finding the Problem - Application
   - To Predict
     - Digit recognition
2. Collecting Dataset
   - Input: Pixel value of Input Image
     Output: 10 Digits (0 to 9)
   - MNIST Digit Dataset
3. Load & Summarize Dataset
   - Load Dataset from the directory & Summarize the details such as data shape & length
     - Data
       - Shape
         - (42000, 785)
     - Target
       - [0 1 2 ... 8 9 8]
4. Segregating Dataset into X & Y
   - iloc - It helps us select a value that belongs to a particular row or column
     - SYNTAX: dataset.iloc[:,start_col:end_col]
     - X = dataset.iloc[:,1:]
     - Y = dataset.iloc[:,0]
5. Splitting Dataset to Train & Test
   - Useful for validation
     - train_test_split(X, Y, test_size = 0.25, random_state = 0)
6. Algorithm
   - Random Forest - Classifier
     - 
     - Definition
       - It consists of a large number of individual decision trees that operate as an ensemble
       - Random forest spits out a class prediction and the class with the most votes becomes our model's prediction
     - HyperParameter
       - n_estimators
         - The number of trees the algorithm builds before taking the maximum voting or taking the averages of predictions
       - max_features
         - The maximum number of features random forest considers to split a node
       - min_sample
         - The minimum number of leafs required to split an internal node
       - n_jobs
         - how many processors it is allowed to use.
         - 1 - limit
         - "-1" - no limit
       - random_state
         - Produce the same results when it has a definite value
       - oob_score
         - Random forest cross-validation method
         - In this sampling, about one-third of the data is not used to train the model and can be used to evaluate its performance - out-of-bag samples
7. Training
   - Training our Model for Pre-processed Dataset
     - model.fit(X_train, y_train)
8. Evaluate/Validation
   - Obtaining the accuracy of the Model
     - accuracy_score
9. Prediction
   - Observing how our model is classifying our new data