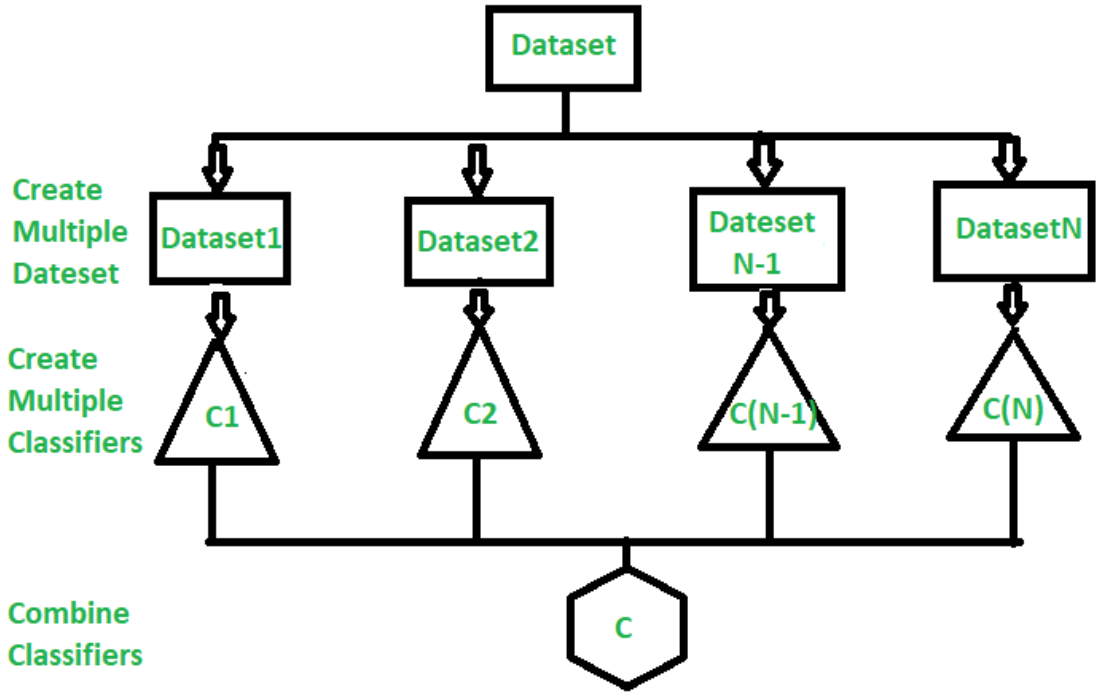
- **Random Forest Classification Algorithm (Ensemble learning)**

Machine Learning

Dr Jagendra Singh

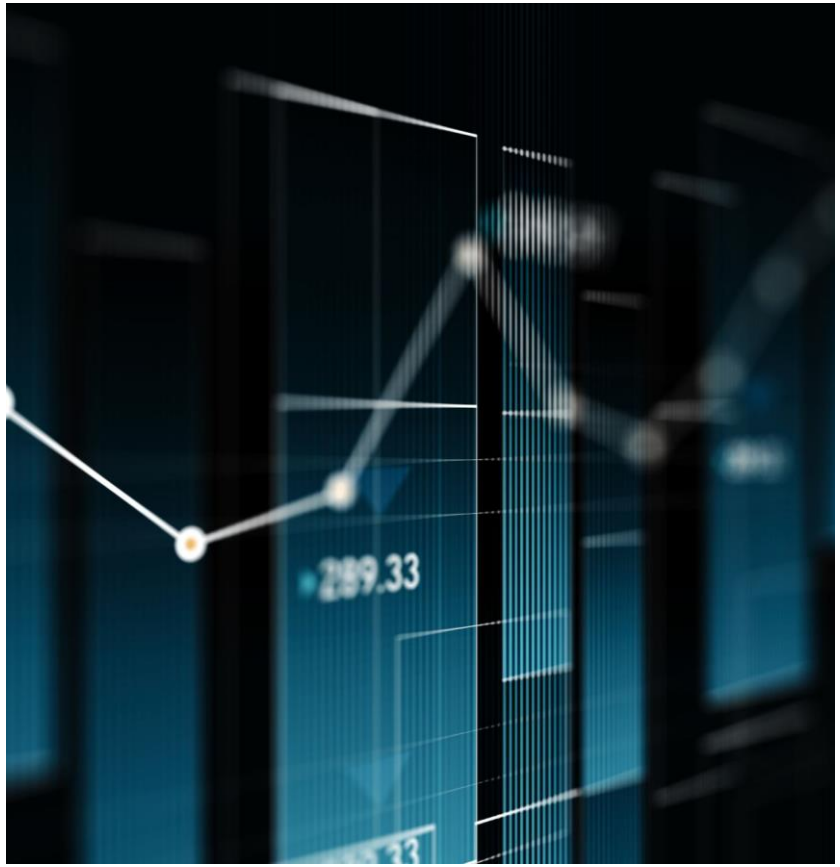
ENSEMBLE CLASSIFIER

- Ensemble learning helps improve machine learning results by combining several models.
- This approach allows the production of better predictive performance compared to a single model.
- Basic idea is to learn a set of classifiers (experts) and to allow them to vote.
- Improvement in predictive accuracy.
- But It is difficult to understand an ensemble of classifiers.



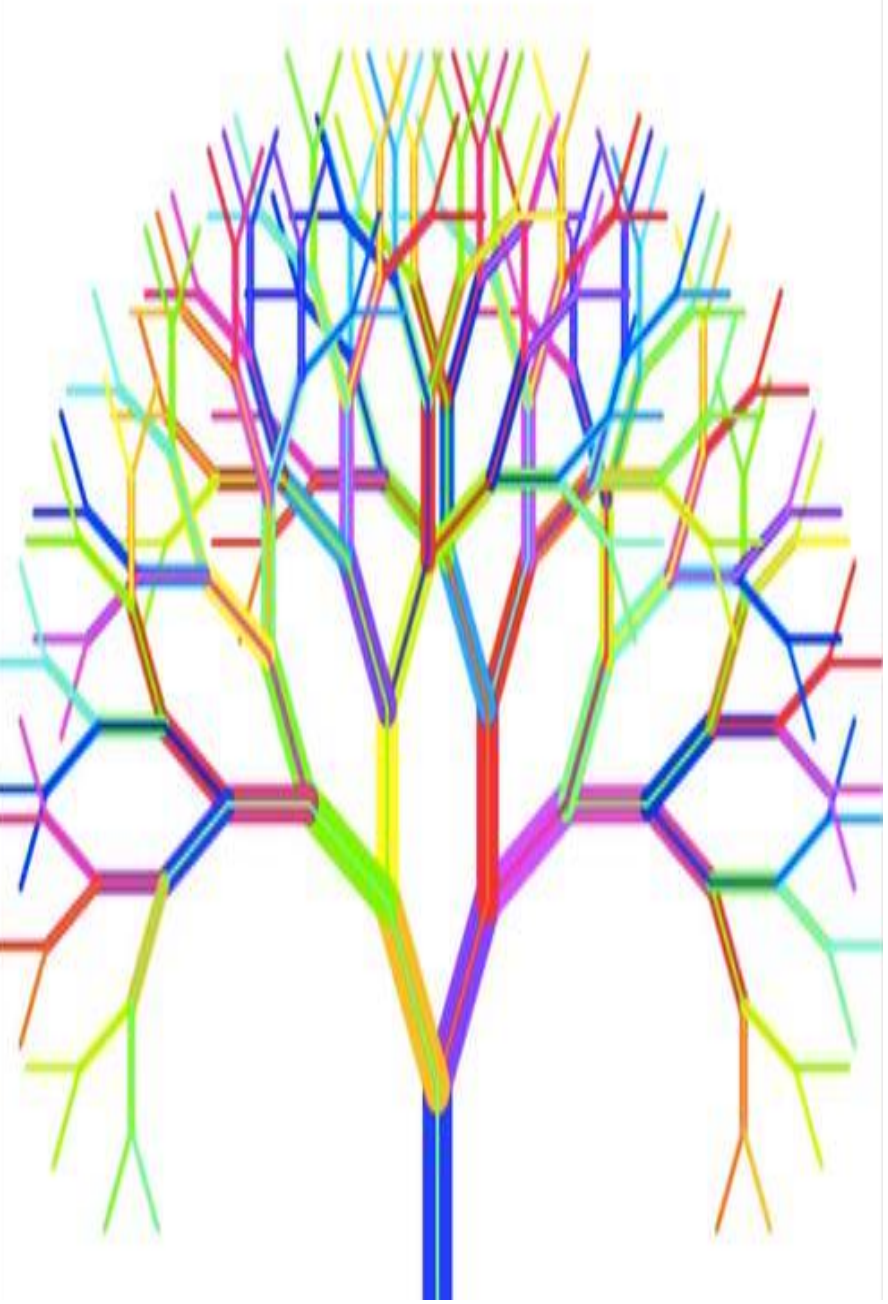
WHY DO ENSEMBLES WORK?

- **Statistical Problem –**
  The Statistical Problem arises when the hypothesis space is too large for the amount of available data.
- **Computational Problem –**
  The Computational Problem arises when the learning algorithm cannot guarantees finding the best hypothesis.
- **Representational Problem –**
  The Representational Problem arises when the hypothesis space does not contain any good approximation of the target classes.



MAIN CHALLENGE FOR DEVELOPING ENSEMBLE MODELS?

- The main challenge is not to obtain highly accurate base models, but rather to obtain base models which make different kinds of errors.
- For example, if ensembles are used for classification, high accuracies can be accomplished if different base models misclassify different training examples, even if the base classifier accuracy is low.
- Methods for Independently Constructing Ensembles -
  - Majority Vote
  - Bagging and Random Forest
  - Randomness Injection
  - Feature-Selection Ensembles
  - Error-Correcting Output Coding



RANDOM FOREST CLASSIFIER

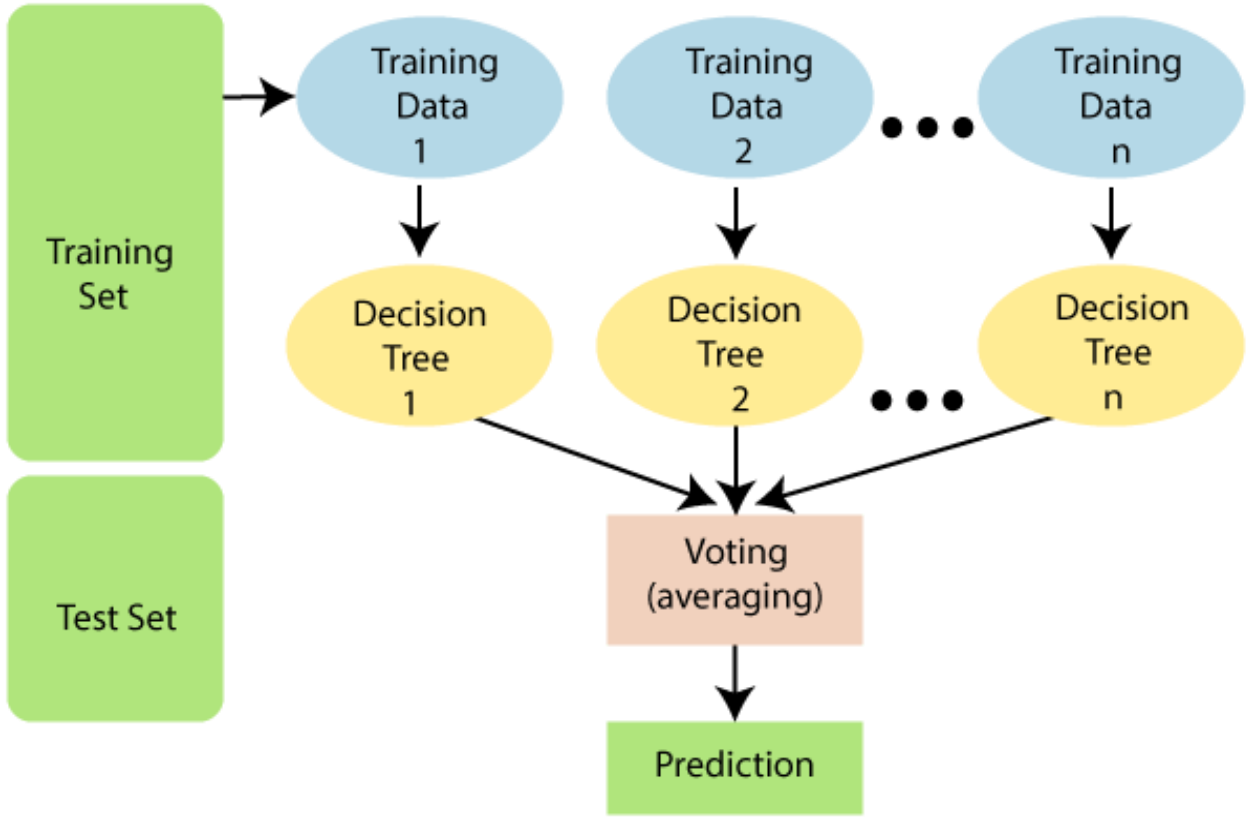
- Random Forest is a popular machine learning algorithm
  - that belongs to the supervised learning technique.
- It can be used for both Classification and Regression problems in ML.
- It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*
- It contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

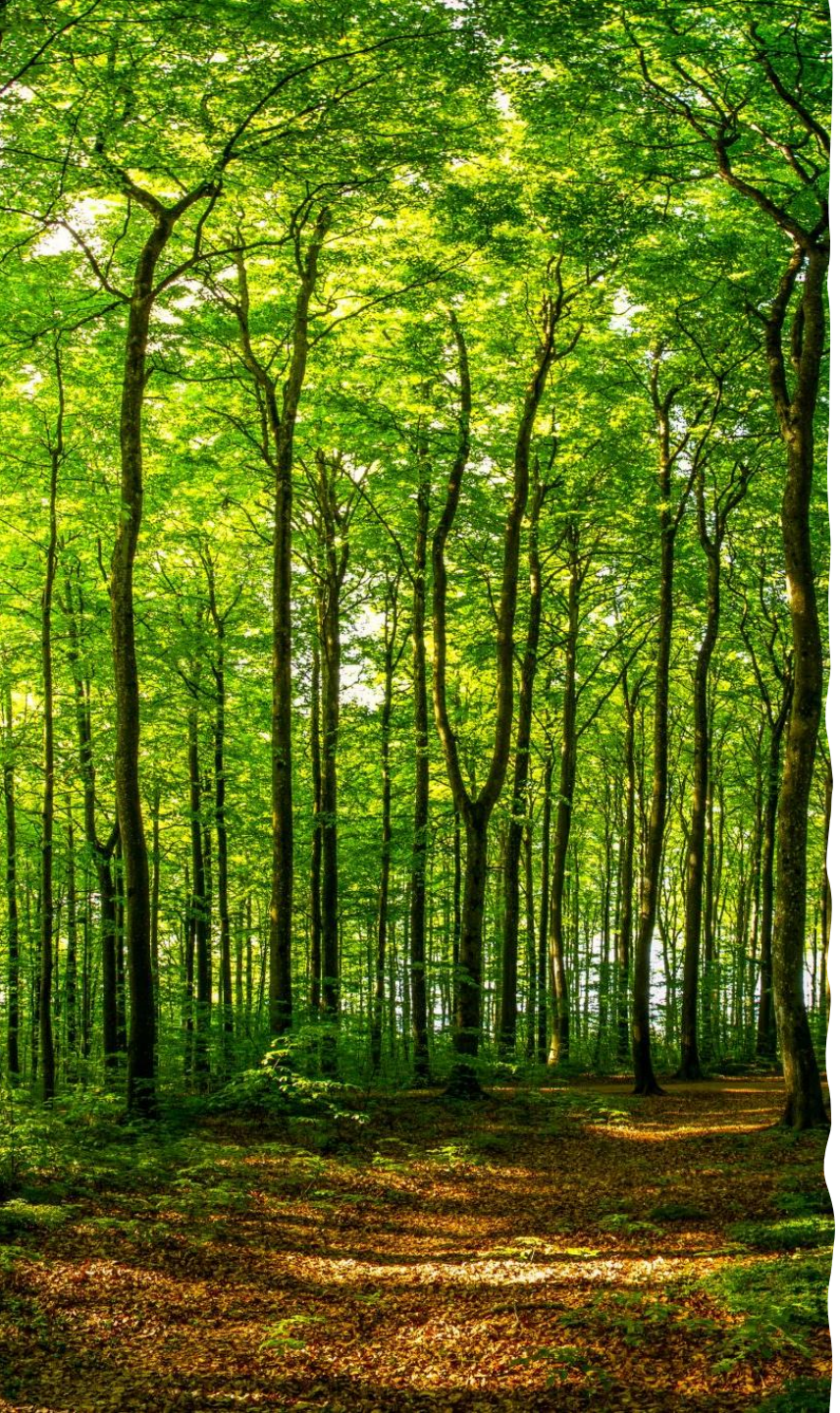
RANDOM FOREST CLASSIFIER

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The diagram explains the working of the Random Forest algorithm:





ASSUMPTIONS FOR RANDOM FOREST

- Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not.
- But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:
- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

FEATURES OF RANDOM FOREST ALGORITHM

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of overfitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

NEED OF RANDOM FOREST?

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

HOW DOES THE ALGORITHM WORK?

Random Forest works in two-phase:

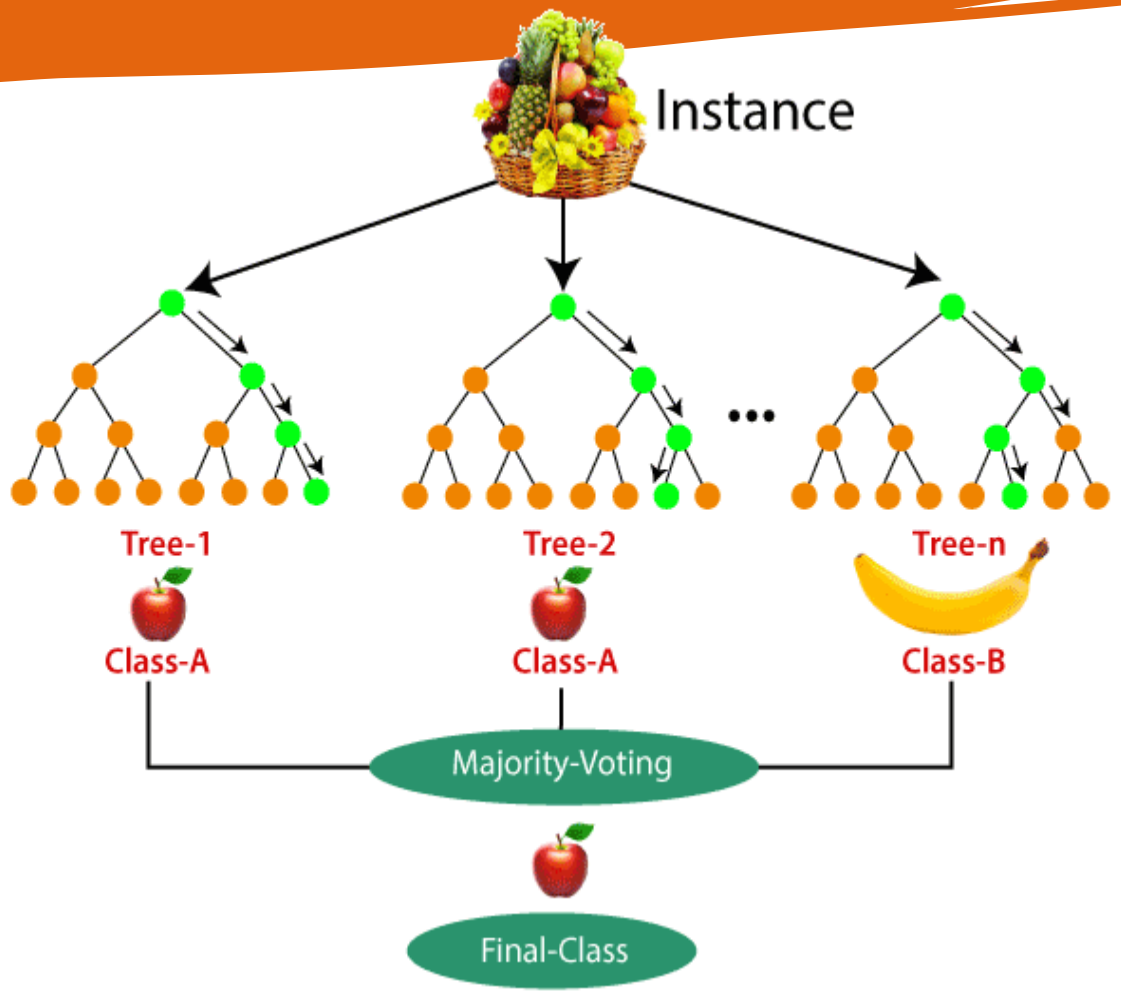
First is to create the random forest by combining N decision tree, and Second is to make predictions for each tree created in the first phase.

- Step-1: Select random K data points from the training set.
- Step-2: Build the decision trees associated with the selected data points (Subsets).
- Step-3: Choose the number N for decision trees that you want to build.
- Step-4: Repeat Step 1 & 2.
- Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

EXAMPLE:

Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier.

The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision.



APPLICATIONS OF RANDOM FOREST

There are mainly four sectors where Random forest mostly used:

- Banking: Banking sector mostly uses this algorithm for the identification of loan risk.
- Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.
- Land Use: We can identify the areas of similar land use by this algorithm.
- Marketing: Marketing trends can be identified using this algorithm.

APPLICATIONS OF RANDOM FOREST

Apart from these this classifier has been applied across a number of industries, allowing them to make better business decisions. Some use cases include:

- Finance: It is a preferred algorithm over others as it reduces time spent on data management and pre-processing tasks. It can be used to evaluate customers with high credit risk, to detect fraud, and option pricing problems.
- Healthcare: The random forest algorithm has applications within computational biology (link resides outside IBM) (PDF, 737 KB), allowing doctors to tackle problems such as gene expression classification, biomarker discovery, and sequence annotation. As a result, doctors can make estimates around drug responses to specific medications.
- E-commerce: It can be used for recommendation engines for cross-sell purposes

ADVANTAGES OF RANDOM FOREST

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

KEY BENEFITS OF RANDOM FOREST

- Reduced risk of overfitting: Decision trees run the risk of overfitting as they tend to tightly fit all the samples within training data.
- Provides flexibility: Since random forest can handle both regression and classification tasks with a high degree of accuracy, it is a popular method among data scientists.
- Feature bagging: also makes the random forest classifier an effective tool for estimating missing values as it maintains accuracy when a portion of the data is missing.

KEY BENEFITS OF RANDOM FOREST

- Easy to determine feature importance: Random forest makes it easy to evaluate variable importance, or contribution, to the model.
- There are a few ways to evaluate feature importance. Gini importance and mean decrease in impurity (MDI) are usually used to measure how much the model's accuracy decreases when a given variable is excluded.
- However, permutation importance, also known as mean decrease accuracy (MDA), is another importance measure. MDA identifies the average decrease in accuracy by randomly permutating the feature values in oob samples.

DISADVANTAGES OF RANDOM FOREST

Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

- Time-consuming process: Since random forest algorithms can handle large data sets, they can be provide more accurate predictions, but can be slow to process data as they are computing data for each individual decision tree.
- Requires more resources: Since random forests process larger data sets, they'll require more resources to store that data.
- More complex: The prediction of a single decision tree is easier to interpret when compared to a forest of them.

PYTHON IMPLEMENTATION OF RANDOM FOREST ALGORITHM

- We will implement the Random Forest Algorithm tree using Python.
- For this, we will use the same dataset "user_data.csv", which we have used in previous classification models.
- By using the same dataset, we can compare the Random Forest classifier with other classification models such as Decision tree Classifier,
- KNN,
- SVM,
- Logistic Regression, etc.

THANK YOU