Ensemble Learning (Part I)

Introduction

Ensemble Learning: Definition

- In machine learning, ensemble learning methods are techniques used to combine multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.
- It works on the hypothesis that certain models do well in one aspect of the data, while others do well in modeling another. The idea is to leverage the strengths of different models and reduce their individual weaknesses.
- "No Free Lunch" Theorem
  - No single algorithm wins all the time!
- When combing multiple independent and diverse decisions each of which is at least more accurate than random guessing, random errors cancel each other out, correct decisions are reinforced.

Real life example- Survey Prediction

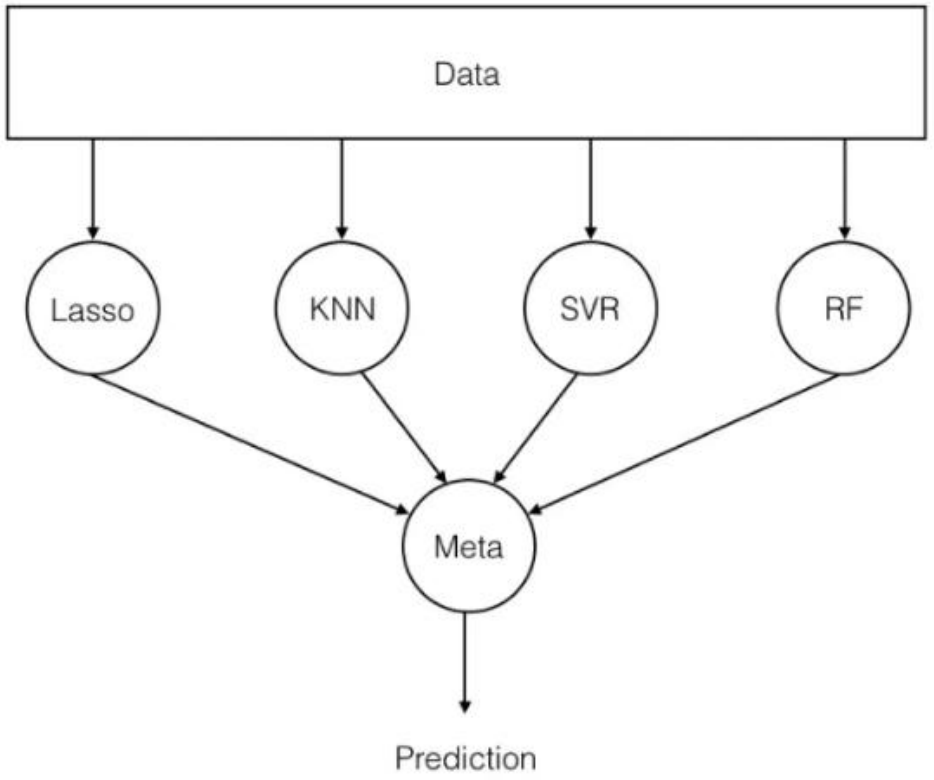
- Let's take a real example to build the intuition.
- Suppose, you want to invest in a company XYZ. You are not sure about its performance though.
- So, you look for advice on whether the stock price will increase by more than 6% per annum or not?
- You decide to approach various experts having diverse domain experience:

- Employee of Company XYZ:
  - In the past, he has been right 70% times.
- Financial Advisor of Company XYZ:
  - In the past, he has been right 75% times.
- Stock Market Trader:
  - In the past, he has been right 70% times.
- Employee of a competitor:
  - In the past, he has been right 60% times.
- Market Research team in the same segment:
  - In the past, he has been right 75% times.
- Social Media Expert:
  - In the past, he has been right 65% times.

Real life example- Survey Prediction

- Given the broad spectrum of access you have, you can probably combine all the information and make an informed decision.
- In a scenario when all the 6 experts/teams verify that it's a good decision(assuming all the predictions are independent of each other), you will get a combined accuracy rate of 1 - (30% . 25% . 30% . 40% . 25% . 35%) = 1 - 0.07875 = 99.92125%
- The assumption used here that all the predictions are completely independent is slightly extreme as they are expected to be correlated. However, you can see how we can be so sure by combining various forecasts together.
- Well, Ensemble learning is **no** different.

Basic Ensemble Structure



- Use multiple learning algorithms (classifiers)
- Combine the decisions
- Can be more accurate than the individual classifiers
- Generate a group of base-learners
- Different learners use different
  - Algorithms
  - Hyperparameters
  - Representations (Modalities)
  - Training sets

Why Ensembles?

- There are two main reasons to use an ensemble over a single model, and they are related; they are:
  - Performance: An ensemble can make better predictions and achieve better performance than any single contributing model.
  - Robustness: An ensemble reduces the spread or dispersion of the predictions and model performance.

Bias-variance tradeoff

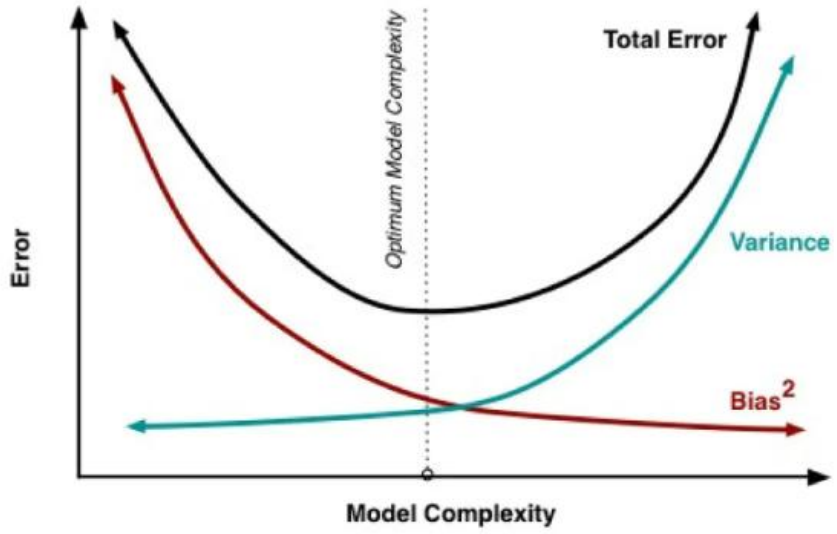
- Machine learning models are designed to generalize from training data to new, unseen data. However, there is a fundamental trade-off between the bias and variance of a model that can affect its ability to generalize.
- Bias of a model is the error in predicting the training dataset, while variance is the change in model with respect to change in training dataset.
- A model with high bias tends to underfit the training data, i.e. the model is too simple to capture the underlying patterns in the data. In contrast, a model with high variance is prone to memorize the noise in the training data rather than the underlying patterns. In other words, it overfit the training data.
- Therefore, we look for a model which is both low bias and low variance, however it is difficult to construct a single model that simultaneously achieves both of them.

Model Error

The error emerging from any machine model can be broken down into three components mathematically. Following are

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

- Normally, as you increase the complexity of your model, you will see a reduction in error due to lower bias in the model. However, this only happens till a particular point.
- As you continue to make your model more complex, you end up over-fitting your model and hence your model will start suffering from high variance.
- A champion model should maintain a balance between these two types of errors. This is known as the trade-off management of bias-variance errors. Ensemble learning is one way to execute this trade off analysis.

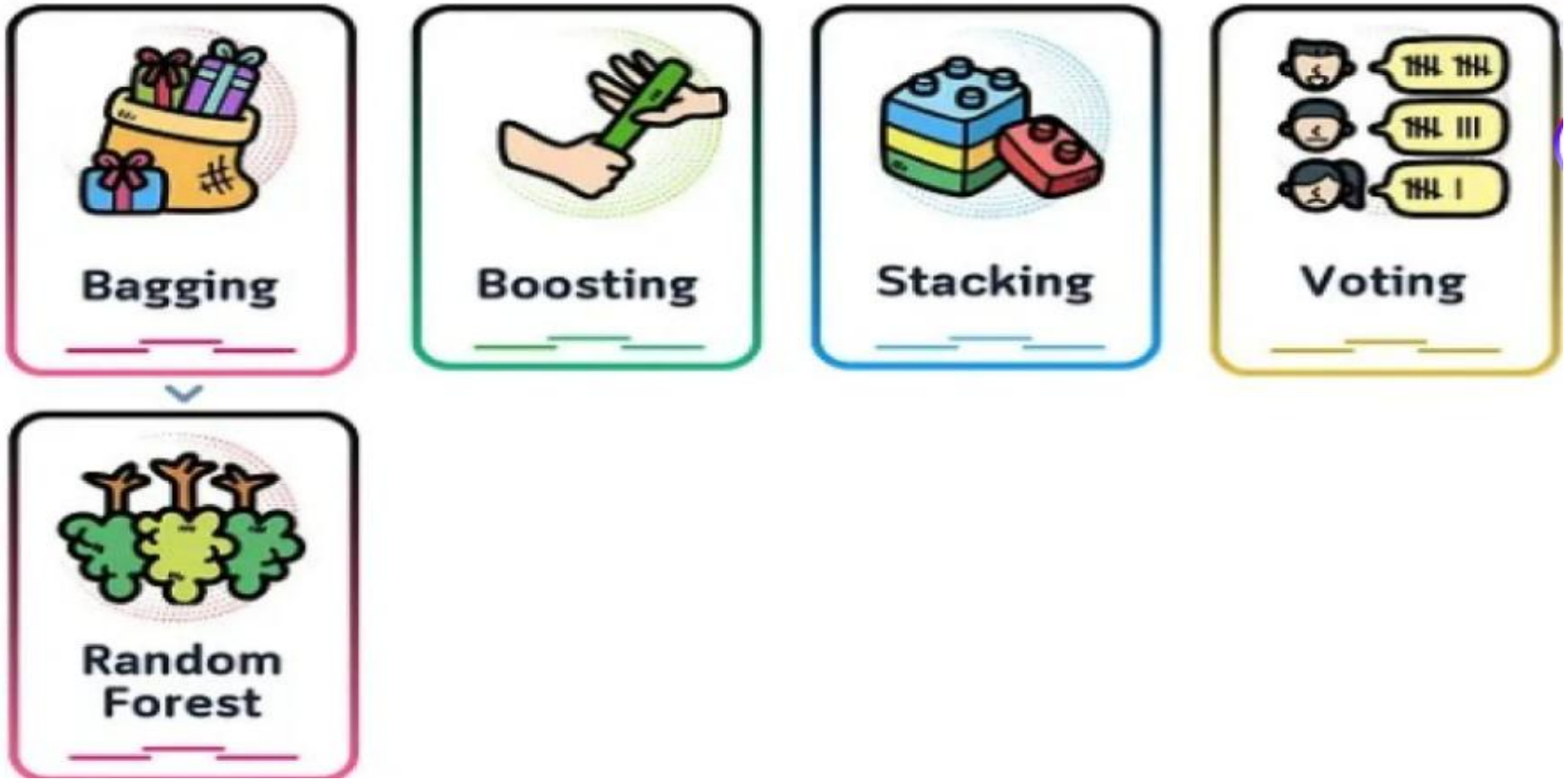


The Netflix Prize: A fun Example

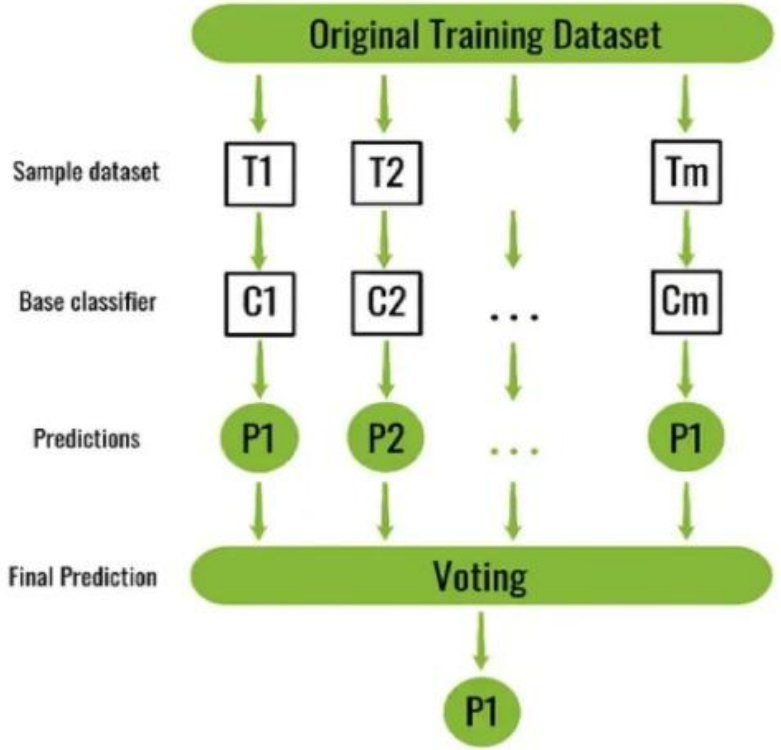
Ensemble learning has been successfully applied in various real-world scenarios, one such example is the Netflix Prize Challenge.

- In 2006, Netflix launched a competition to improve their recommendation algorithm, which recommends movies and TV shows to users based on their viewing history.
- The goal was to reduce the root mean square error (RMSE) of their existing algorithm by 10%. The competition attracted more than 40,000 teams from around the world, and the winning team, BellKor's Pragmatic Chaos, used an ensemble of multiple machine learning models to achieve a 10.06% improvement in RMSE.
- The winning solution was a blend of 107 different models, each using different combinations of features and algorithms. By combining the predictions of these individual models, the ensemble was able to achieve a much lower RMSE than any individual model.

Types of Ensemble methods



Voting Classifiers



Introduction to Voting classifiers

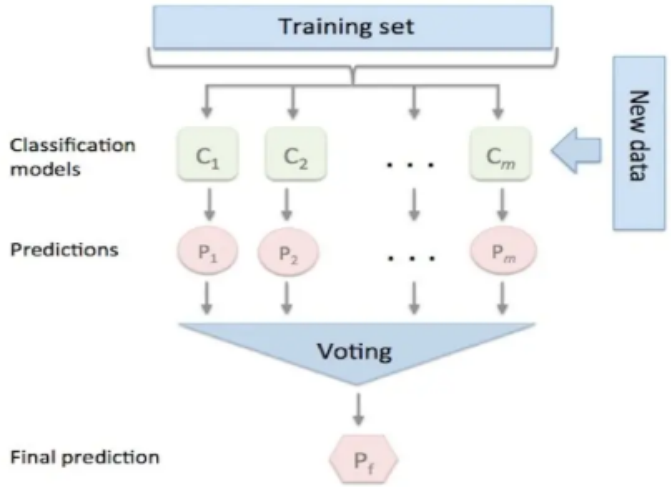
- A Voting Classifier is an ensemble machine learning technique that combines the predictions from multiple individual classifiers (also known as base classifiers or estimators) to make a final prediction.

- It's a type of model averaging approach where each base classifier contributes its prediction, and the final prediction is determined by a majority vote (for classification) or an average (for regression).

- The Voting Classifier can be used for both binary and multiclass classification tasks.

Hard Voting

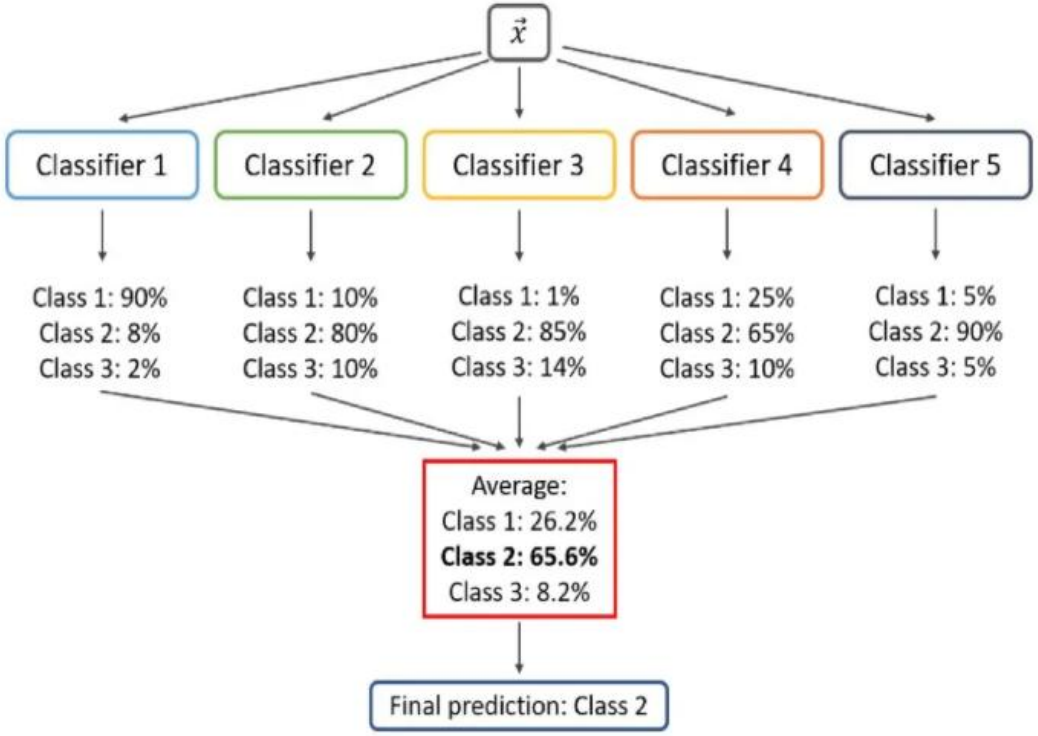
In hard voting, each base classifier's prediction is **treated as a vote**, and the **final prediction is the majority vote** among the predictions of the individual classifiers. This is commonly used for **classification tasks**.



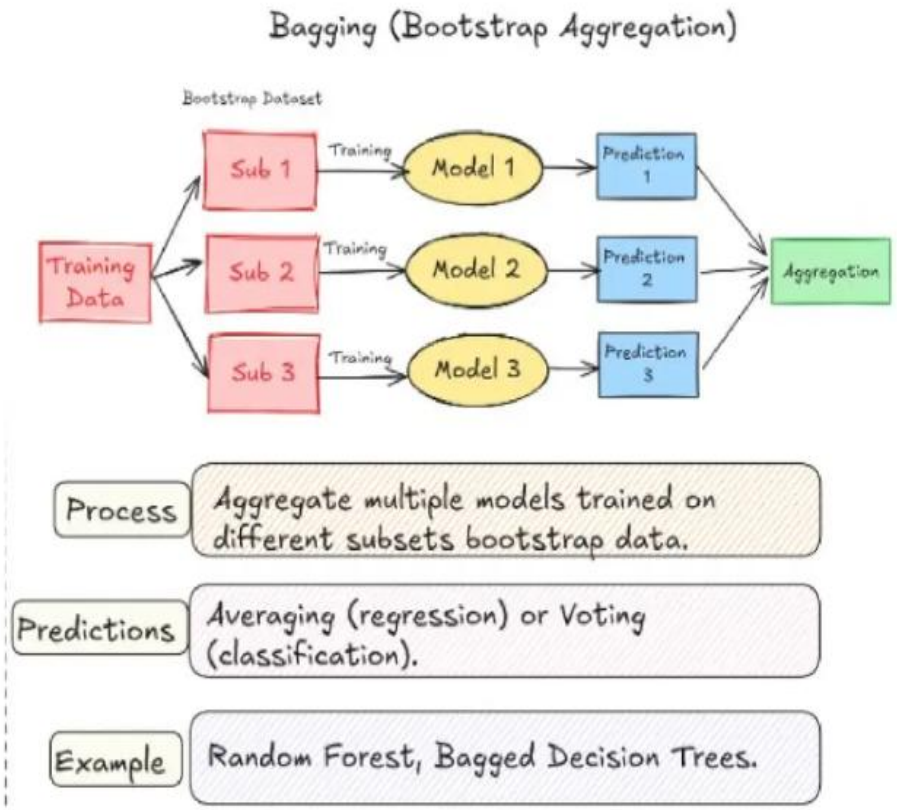
Soft Voting

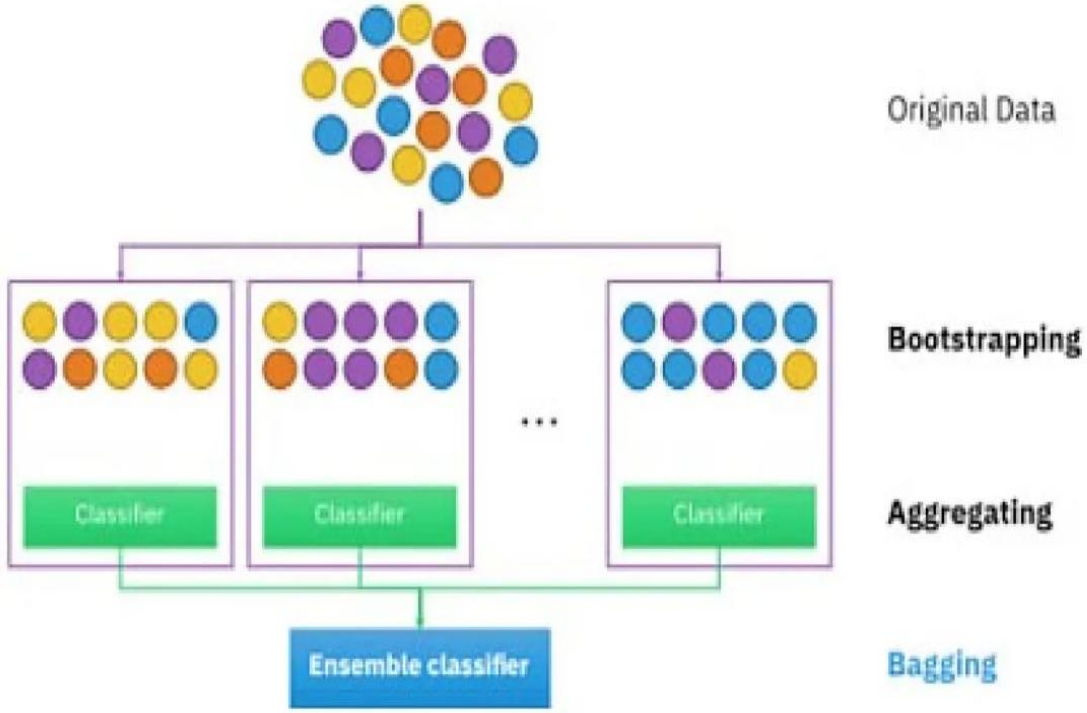
Soft Voting: In soft voting, each base classifier's predicted **probabilities for each class are averaged**, and the **class with the highest average probability is chosen as the final prediction**.

Soft voting often produces **better results than hard voting** because it takes into account the **confidence levels** of the classifiers.



Bagging


Bagging (Bootstrap Aggregation)
Bootstrap Dataset
Training Data
Sub 1
Sub 2
Sub 3
Training
Model 1
Model 2
Model 3
Prediction 1
Prediction 2
Prediction 3
Aggregation
Process
Aggregate multiple models trained on different subsets bootstrap data.
Predictions
Averaging (regression) or Voting (classification).
Example
Random Forest, Bagged Decision Trees.


Original Data
Classifier
Classifier
Classifier
Ensemble classifier
Bootstrapping
Aggregating
Bagging

Bagging

Purpose: **Reduce variance and prevent overfitting**

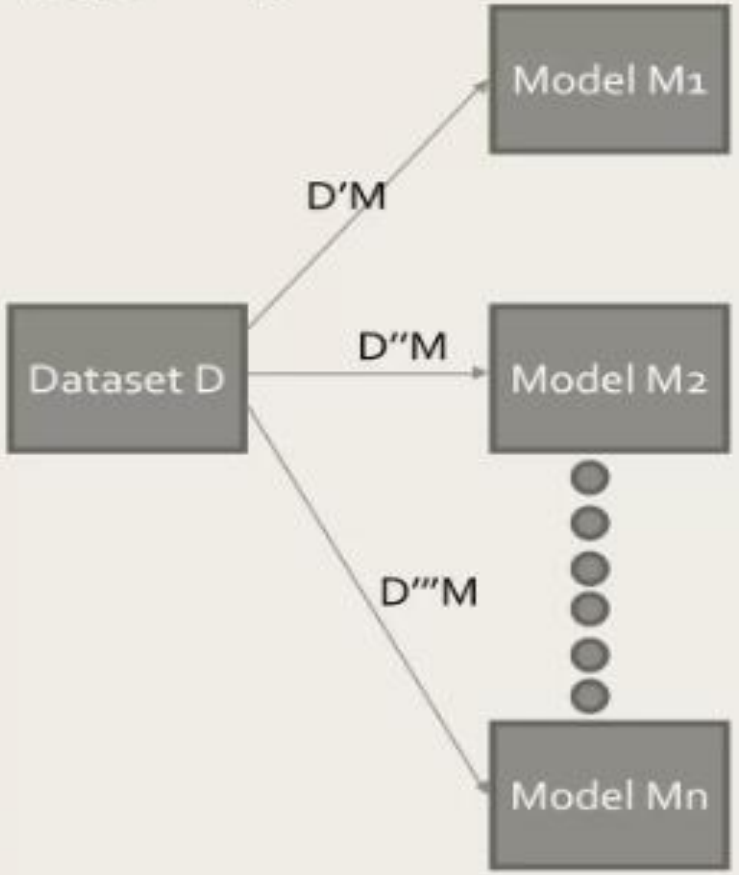
How it works: Bagging trains **multiple models** independently on different subsets of the training data. These subsets are **created by random sampling with replacement** (bootstrap sampling).

After training, the predictions from all models are averaged (for regression) or voted on (for classification).

Example: Random Forest, which is an ensemble of decision trees.

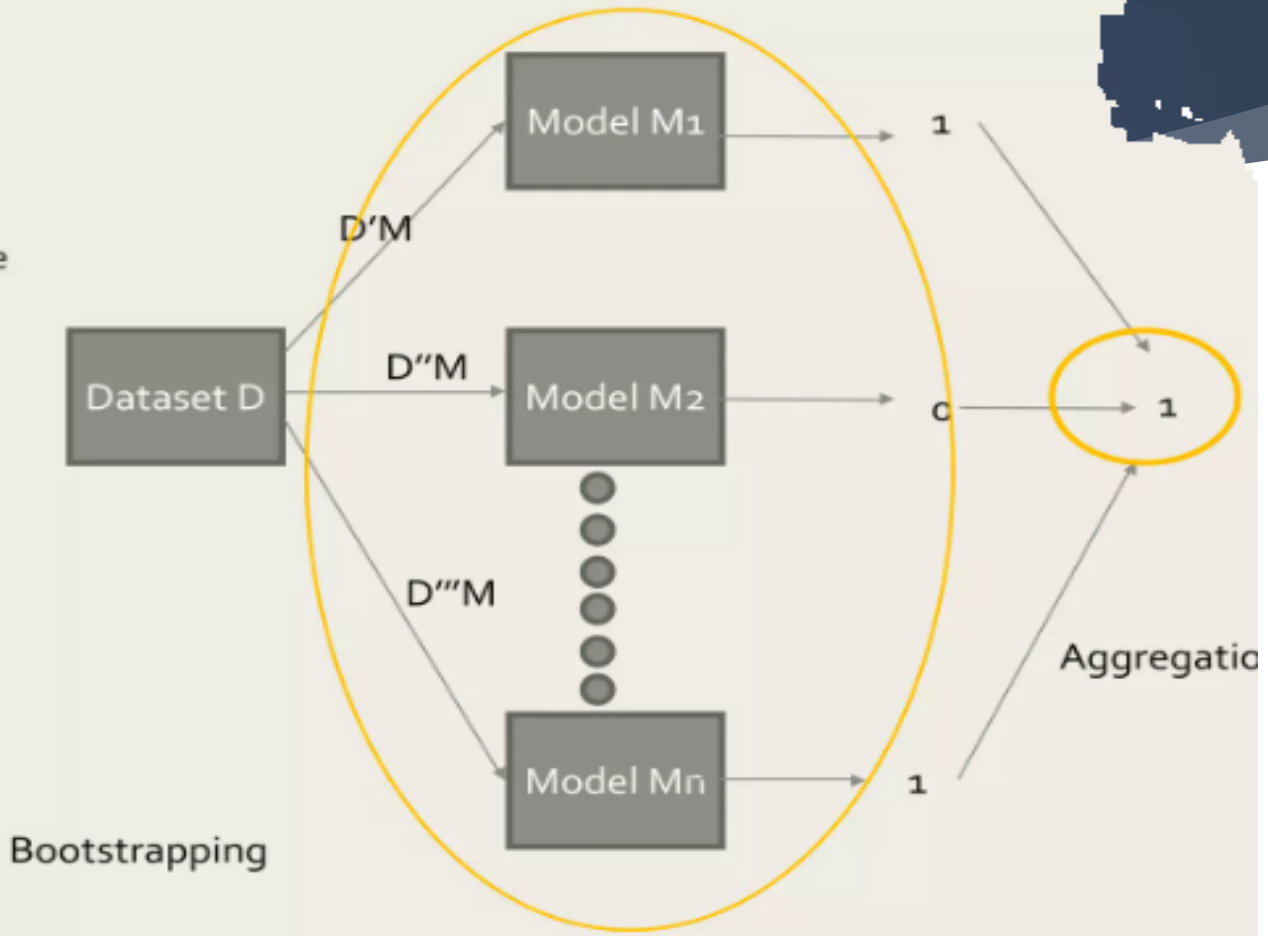
.....Working with bagging

- Consider dataset D.
- It has many rows and columns
- Consider models or base learners M(M1,M2,...,Mn) for dataset D
- For each model we provide dataset D'M,D''M,Etc.
- Suppose we have n records we select sample of n records and provide a particular record to model 1
- Similarly for next model we use **row sampling with replacement.**
- For example in model M1 if there is data (A,B) ,then for model M2(B,C) where B is reptive
- After training is done we give new test data to predict.
- Now we consider this method in binary classifier model



.....solving with test data in bagging

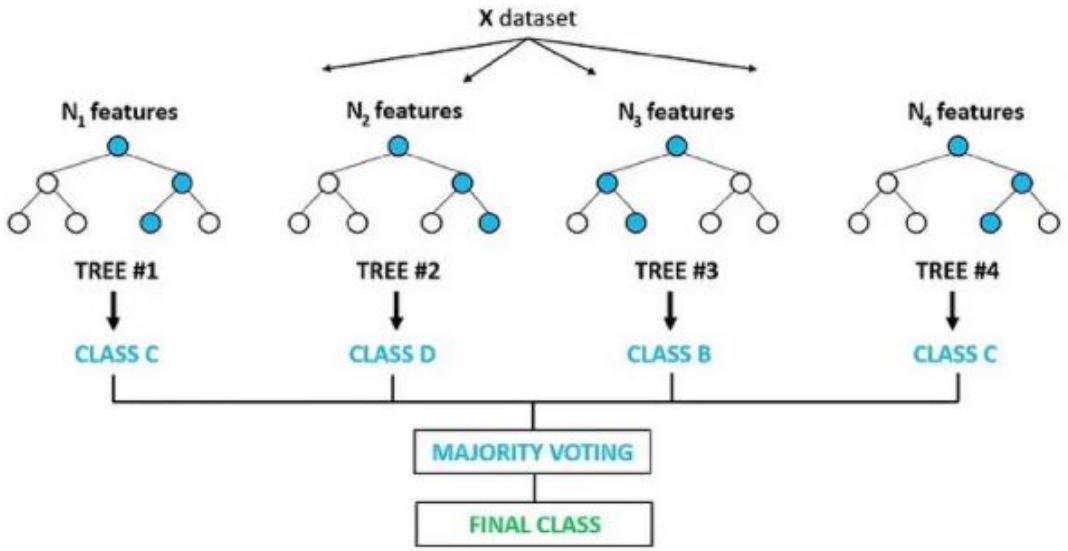
- Suppose we give new test data and made them to pass
- The models gives their values as 1 or 0 as we consider binary classifier
- In the given dataset by voting classifier the majority (1) is taken as O/P



Random Forest

- Random Forest is a popular ensemble learning algorithm, which is an extension of the vanilla bagging algorithm.
- The first algorithm for random decision forests was created in 1995 by Tin Kam Ho. In this algorithm, he introduced the idea of random feature selection for a high cardinality of feature space, which is the key difference between vanilla bagging and random forest.
- The algorithm for random forests is similar to that of bagging methods. However, in random forests, a subset of pre-decided length is formed from original feature space for each of the bootstrapped dataset.
- A decision tree is formed for each dataset and corresponding feature space, leading to a prediction from each. Final prediction is made following the same rules as of bagging, i.e. taking mean for regression and mode for classification.

Random Forest Classifier



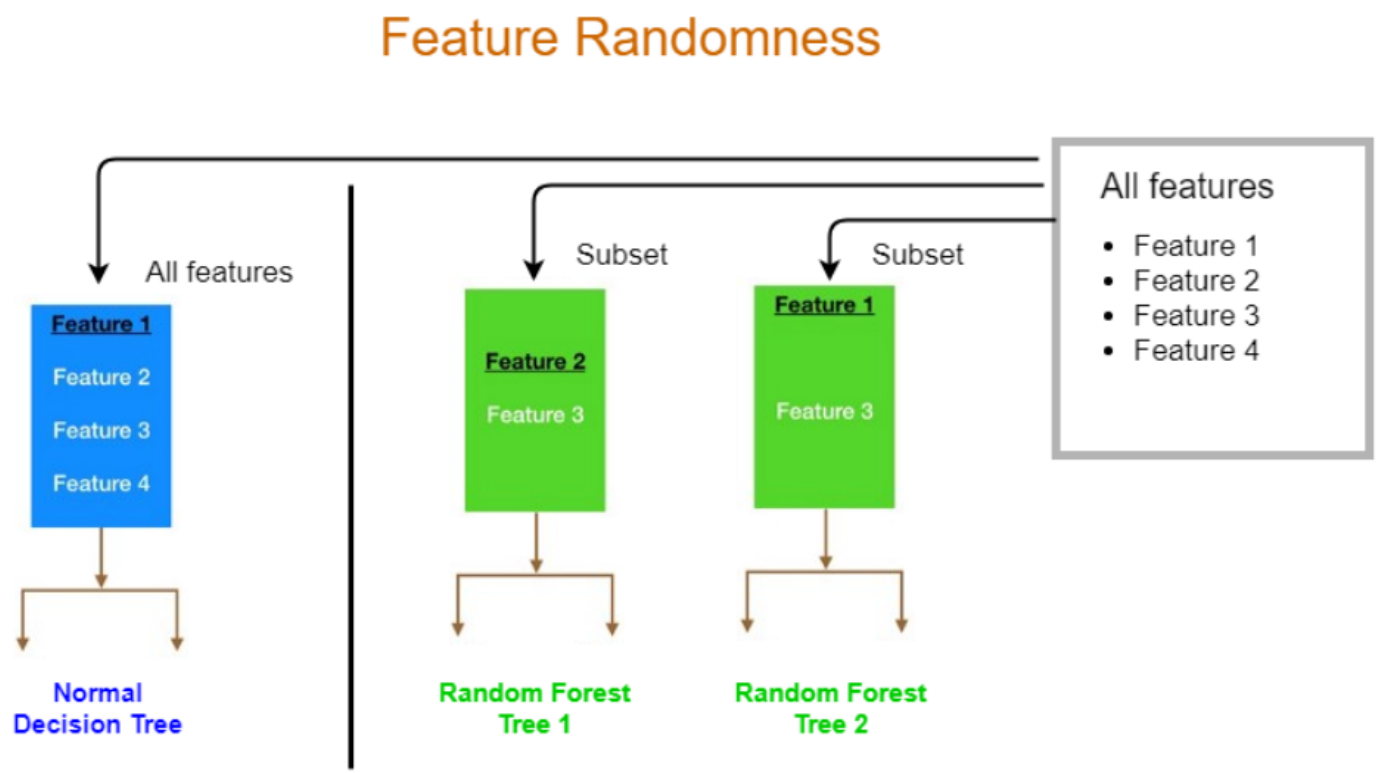
Bootstrap Sampling: Each tree gets its own unique training set, created by randomly sampling from the original data with replacement.
This means some data points may appear multiple times while others aren't used.

Random Feature Selection: When making a split, each tree only considers a random subset of features (typically square root of total features).

Growing Trees: Each tree grows using only its bootstrap sample and selected features, making splits until it reaches a stopping point (like pure groups or minimum sample size).

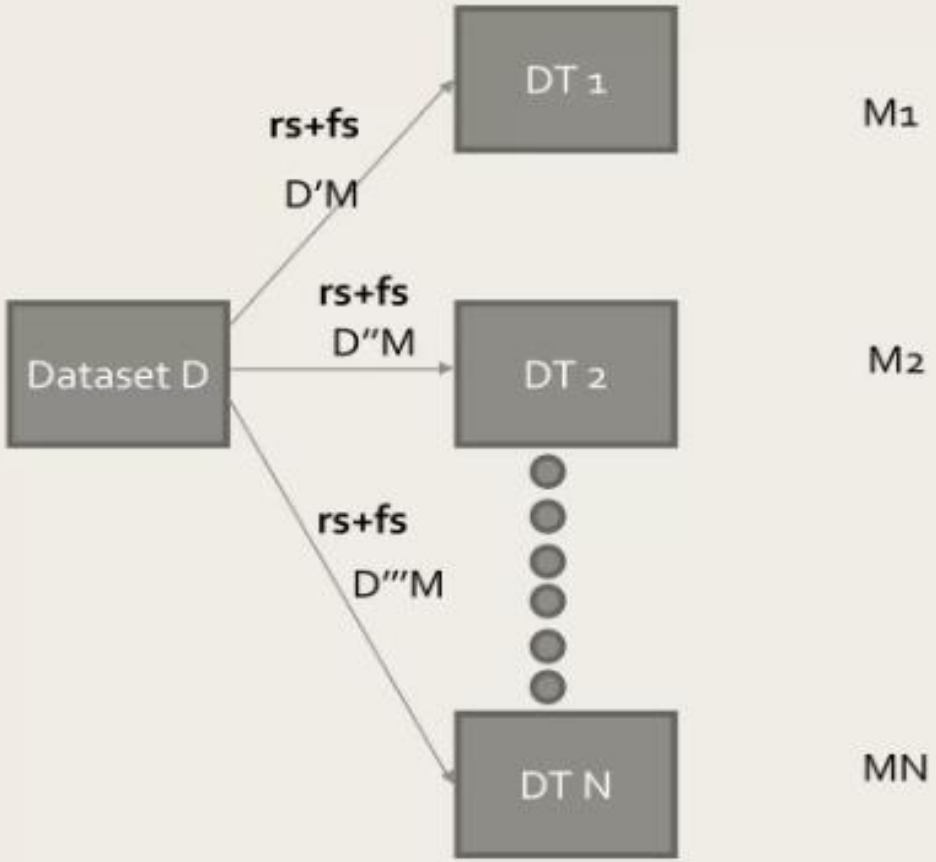
Final Prediction: All trees vote together for the final prediction. For classification, take the majority vote of class predictions; for regression, average the predicted values from all trees.

Feature randomness



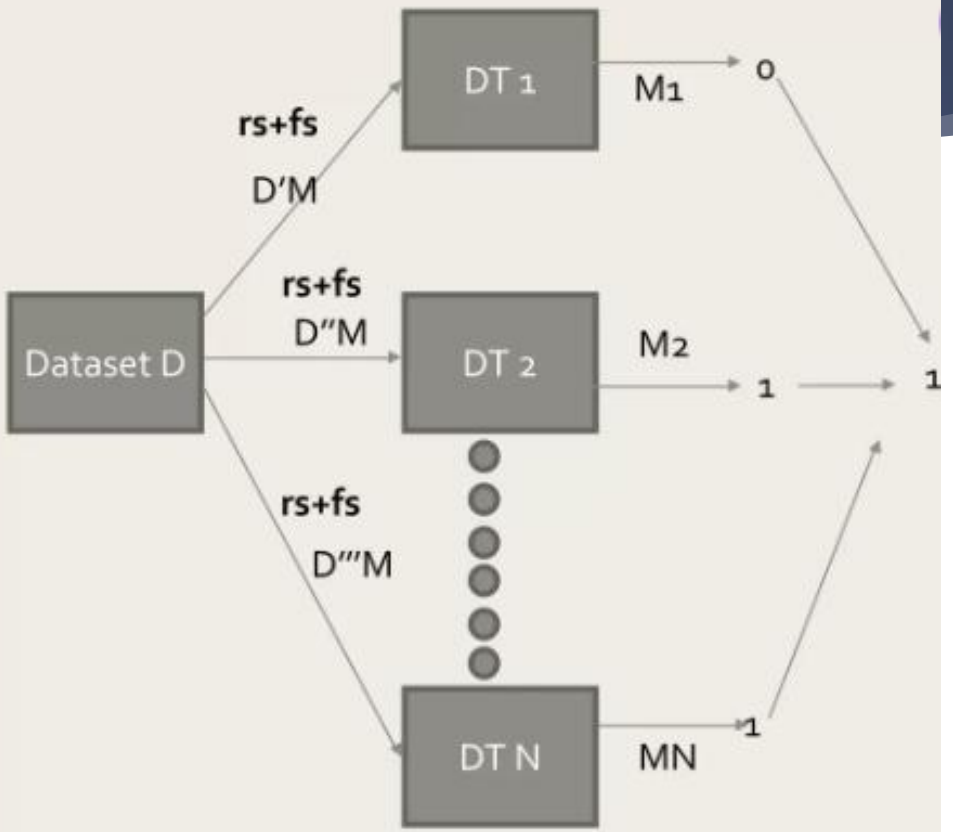
.....Working with Random forest

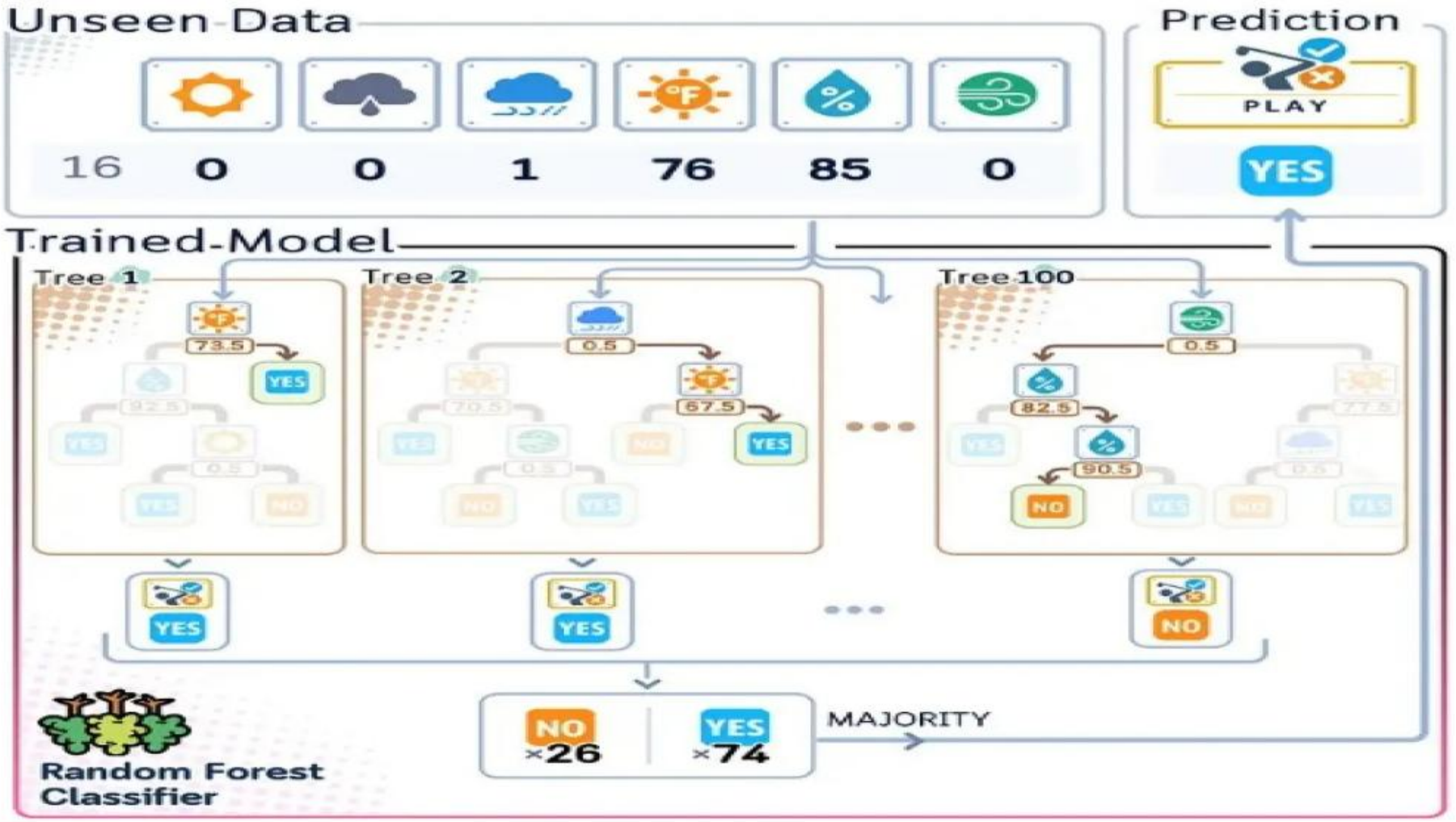
- Consider dataset D.
- It has many rows and columns
- Consider models or base learners and decision tree M(M1,M2,...,Mn) & Decision tree (DT 1,DT 2,DTN) for dataset D
- For each model we provide dataset D'M,D''M,Etc..
- Suppose we n records we select sample of n records and provide a particular record to model 1
- Similarly for next model we use **row sampling with replacement (rs) and sample of feature (FS)**
- Take some no of rows and columns and give it to the DT.It will be trained on particular dataset. Similarly for DT2,,,
- After training is done we give new test data and training to predict.
- Now we consider this method in binary classifier model



.....Working with Random forest

- Suppose we give new test data and made them to pass
- The models gives their values as 1 or 0 as we consider binary classifier
- In the given dataset by voting classifier the majority (1) is taken as O/P




Unseen Data
16 0 0 1 76 85 0
Prediction
PLAY
YES
Trained-Model
Tree 1
73.5
YES
92.5
YES
0.5
YES
NO
Tree 2
0.5
67.5
YES
70.5
YES
NO
0.5
NO
YES
Tree 100
0.5
82.5
90.5
NO
77.5
YES
NO
0.5
YES
YES
NO
Random Forest Classifier
NO ×26
YES ×74
MAJORITY

Key benefits and challenges

- **Benefits:**
  - **Reduced risk of overfitting:** Decision trees are prone to overfitting because they try to fit all the samples in the training data too closely. Random Forests with a large number of decision trees reduce the risk of overfitting because the averaging of independent trees lowers the overall variance and prediction error .
  - **High accuracy:** Random Forest can handle both regression and classification problems with high accuracy, making it a popular method among data scientists.
  - **Robustness:** Random Forest is robust to noise and missing values, which makes it a good choice for data with missing or incomplete values.
- **Challenges:**
  - **Time and Resource Demanding:** As Random Forest computes a separate decision tree with different feature space for each subset of data, it takes a lot more time and computation power compared to a simple decision tree.
  - **Interpretability:** Even though Random Forest provides a measure of feature importance, prediction of a single decision tree is easier to interpret since it shows the path of the decision-making process for that specific sample.