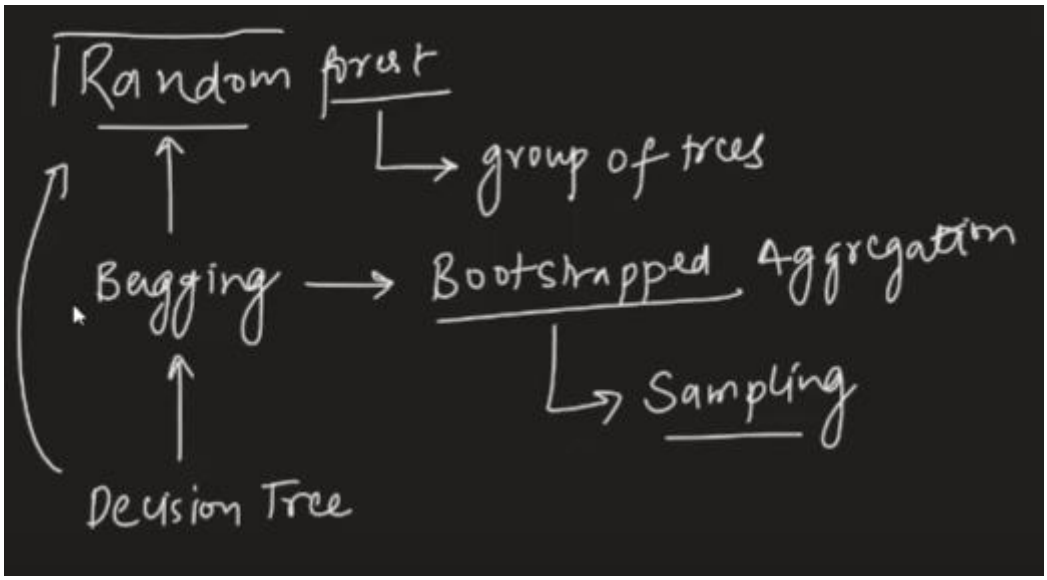
Machine Learning

Video 91:

Introduction to Random Forest | Intuition behind the Algorithm

Random Forest is named for its structure: "Random" refers to the randomness in feature selection and data sampling, while "Forest" represents the collection of decision trees. It builds multiple trees using random subsets of data and features, then aggregates their predictions for better accuracy and reduced overfitting.



Example:

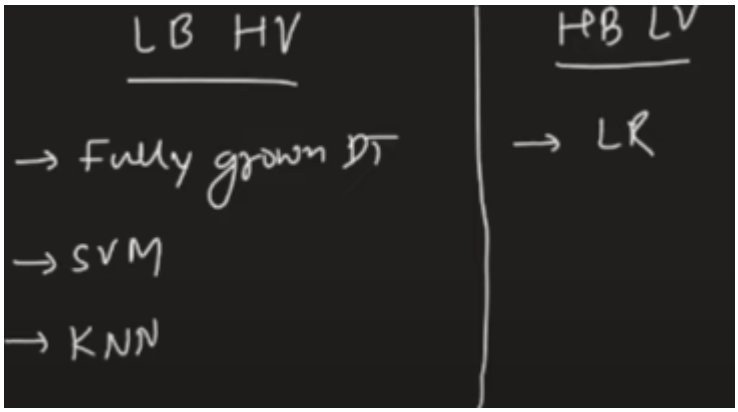
Code link:

https://colab.research.google.com/drive/1fZ_A0lQPuYXXR607byOaPqb-_BS5eRdW?usp=sharing

Video 92:

How Random Forest Performs So Well? Bias Variance Trade-Off in Random Forest

Low variance and low variance will be ideal, but bias is inversely proportional to variance.



Random forest converts LBHV to LBLV.

Random Forest performs well because it combines multiple decision trees, reducing overfitting and improving generalization. It uses bagging (bootstrap aggregation) to train each tree on random subsets of data and selects random features at each split, ensuring diversity. By averaging predictions (or majority voting in classification), it achieves high accuracy and robustness.

Random Forest balances **bias** and **variance** effectively:

- It **reduces variance** by averaging multiple decision trees, preventing overfitting that a single deep tree might have.
- It **maintains low bias** since individual decision trees are low-bias models.

This trade-off makes it more robust and accurate than a single tree while avoiding high variance.

Example:

Code link:

https://colab.research.google.com/drive/1fZ_A0lQPuYXXR607byOaPqb-_BS5eRdW?usp=sharing

Video 93:

Bagging Vs Random Forest | What is the difference between Bagging and Random Forest | Very Important

Feature	Bagging	Random Forest
Definition	An ensemble method that trains multiple models (often decision trees) on different bootstrapped subsets of data and averages their predictions.	A specialized form of bagging that builds multiple decision trees with additional randomness in feature selection.
Feature Selection	Uses all features for tree splits.	Selects a random subset of features at each split, adding extra randomness.
Overfitting Prevention	Reduces variance but may still overfit if trees are too deep.	More robust against overfitting due to feature randomness.
Performance	Works well but may not always generalize as effectively.	Usually outperforms bagging by reducing correlation between trees.
Common Algorithm	Bagging applied to Decision Trees (a.k.a. Bagged Trees).	Random Forest algorithm.

🔹 **Key Difference:** Random Forest is an improvement over Bagging by introducing random feature selection, making it more powerful and less prone to overfitting.

Example:

Code link:

https://github.com/campusx-official/100-days-of-machine-learning/blob/main/day65-random-forest/bagging_vs_random_forest.ipynb

Video 94:

Random Forest Hyper-parameters

Random Forest has several important hyperparameters that control its performance. They can be categorized into:

1. Main Hyperparameters

- **n_estimators** → Number of trees in the forest (higher = better but slower).
- **max_depth** → Maximum depth of each tree (prevents overfitting).
- **min_samples_split** → Minimum samples required to split a node (higher = less overfitting).
- **min_samples_leaf** → Minimum samples required in a leaf node (higher = smoother predictions).
- **max_features** → Number of features considered for splitting at each node (controls randomness).

2. Regularization Parameters (Prevent Overfitting)

- **max_samples** → Maximum number of samples for each tree (used in bootstrapping).
- **max_leaf_nodes** → Maximum number of leaf nodes per tree (reduces complexity).
- **bootstrap** → Whether to use bootstrapping (True by default).

3. Performance Optimization Parameters

- **n_jobs** → Number of parallel jobs for training (faster computation).
- **random_state** → Controls randomness for reproducibility.
- **oob_score** → Use out-of-bag samples for validation (True/False).

Video 95:

Hyperparameter Tuning Random Forest using GridSearchCV and RandomizedSearchCV | Code Example

Example;
Code link:

https://github.com/campusx-official/100-days-of-machine-learning/blob/main/day65-random-forest/code-example-random-forest.ipynb