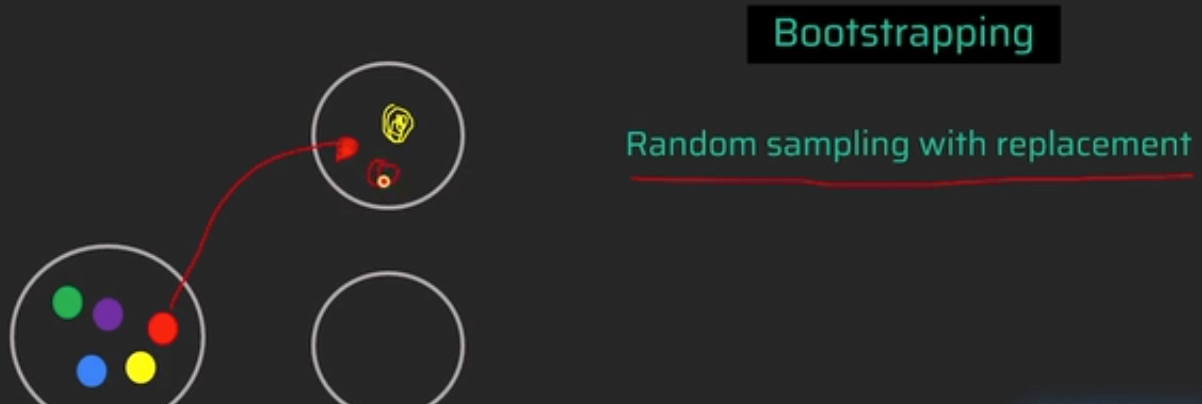
**Bagging (Bootstrap Aggregating)** is an ensemble method that:

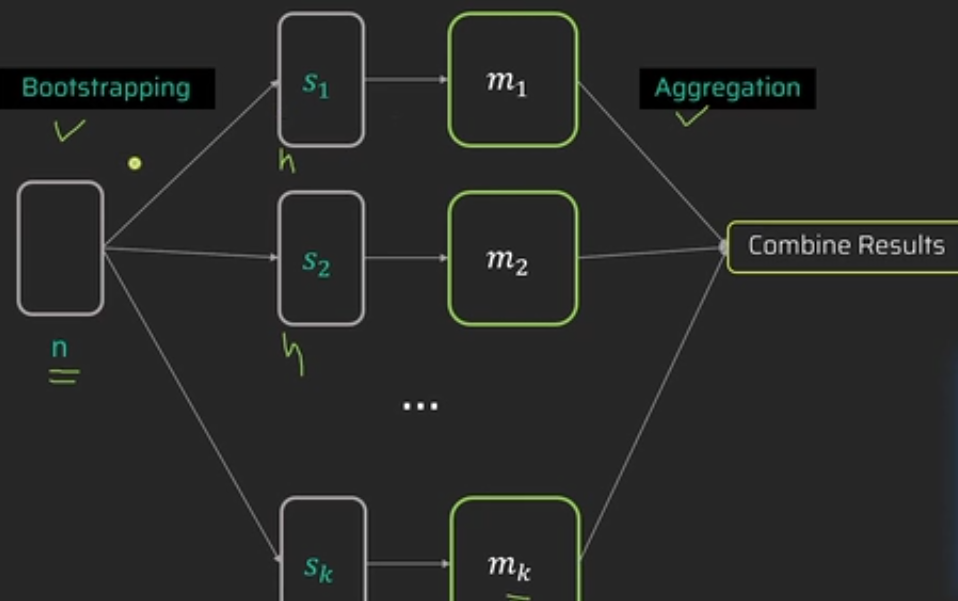
* Builds **multiple versions** of a model (usually the same type, like decision trees).
* Trains each model on a **random sample (with replacement)** of the training data.
* Combines all their predictions by:
  + **Majority vote** (for classification)
  + **Averaging** (for regression)

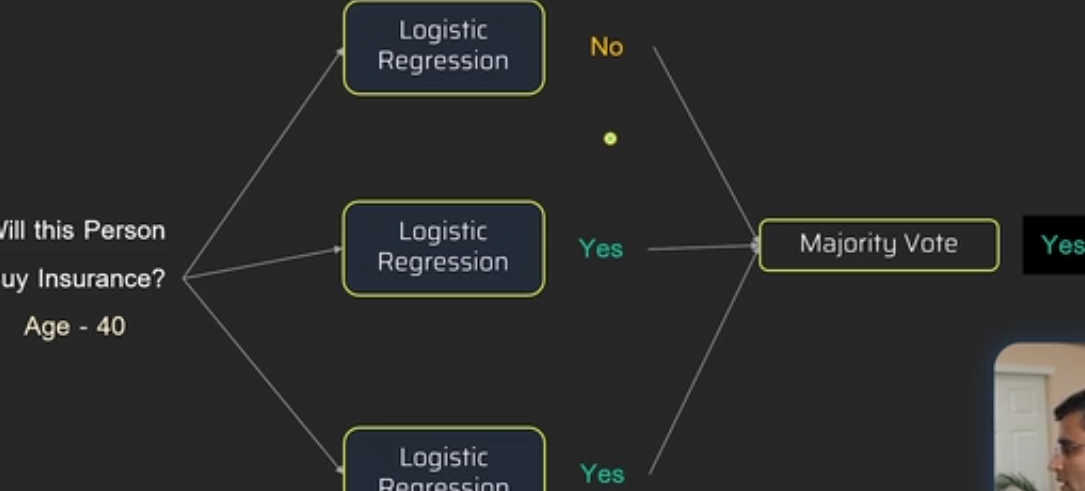
The idea is that by combining many "noisy" or weak models, you get a more accurate and stable prediction.

**🎯 Why Use Bagging?**

* Reduces **variance** (overfitting) without increasing bias.
* Helps when your model is **unstable** (e.g., decision trees that change drastically with small data changes).
* Improves **robustness** and **accuracy**.
* Robust against outliers
* A good way to handle high dimensionality







### ****Most Popular Bagging Algorithm: Random Forest****

Random Forest is essentially **bagging applied to decision trees**, with an extra twist: at each split in a tree, it considers only a **random subset of features**.

**Random Forest** is an **ensemble learning method** that builds a "forest" of decision trees, using the technique of **bagging** plus an extra layer of randomness.

It works by:

1. Training **multiple decision trees** on different **random subsets** of the data (bagging).
2. At each split in each tree, it uses a **random subset of features** instead of all features.
3. The final prediction is made by:
   * **Majority vote** (for classification)
   * **Average** (for regression)

**Why Use Random Forest?**

1. **Out-of-the-box performance**: Works well with little tuning.
2. **Handles missing values**.
3. **Less prone to overfitting** than a single decision tree.
4. **Feature importance**: Easy to interpret which features matter.
5. **Parallelizable**: Each tree can be trained independently.
6. Works well with both numerical and categorical features.
7. Handles **non-linear relationships** and interactions between variables.

