**AdaBoost (Adaptive Boosting)**

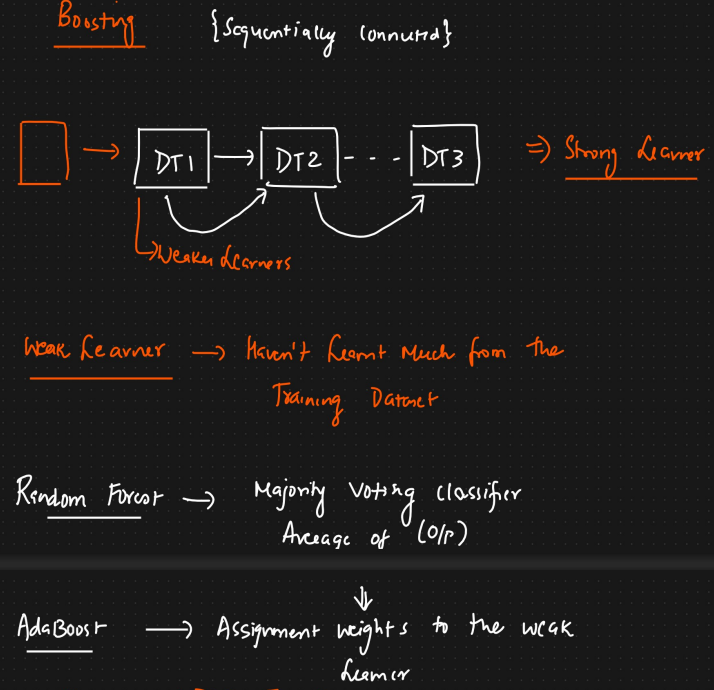
* AdaBoost is an **ensemble learning method**, specifically a **boosting** algorithm.
* **Boosting:** Builds the ensemble *sequentially*. Each new model attempts to correct the errors made by the previous models.
* **Adaptive:** It adapts by giving more weight to data points that previous models misclassified, forcing subsequent models to focus on these "harder" examples. It also weights the contribution of each weak learner based on its accuracy.

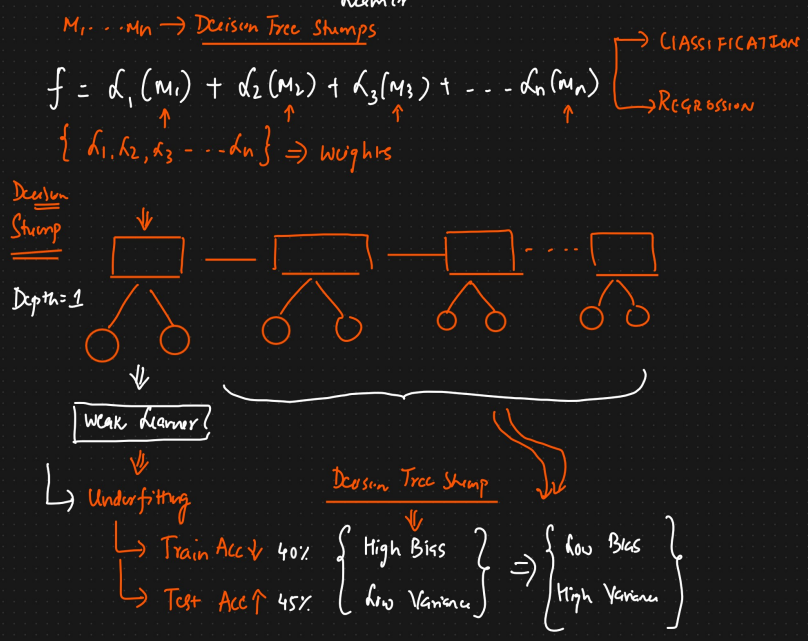
**2. Core Idea & Intuition**

Imagine you're studying for an exam with a group. Instead of everyone studying everything equally:

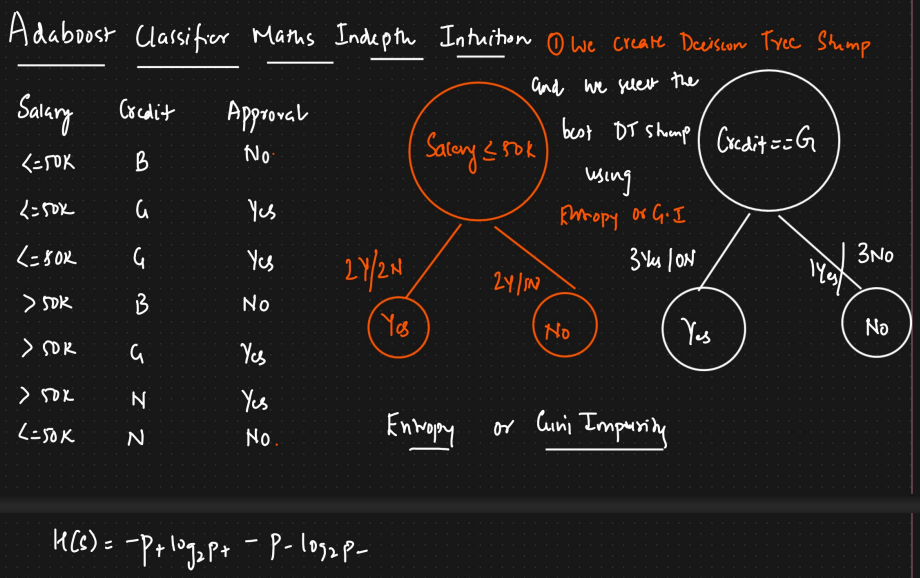
* First, everyone tries all topics (initial weak learner).
* You identify the topics where the group performed poorly (misclassified examples).
* The next study session focuses *more intensely* on these difficult topics (increasing weights for misclassified points).
* Someone who is particularly good at a certain topic gets their opinion weighted more heavily on questions related to that topic (weighting the weak learners).
* You repeat this process, adaptively focusing on weaknesses until the group (the ensemble) performs well overall.
* In this we make a decision tree of 1 level i.e. stumps.

AdaBoost does this with data points and simple models (weak learners).



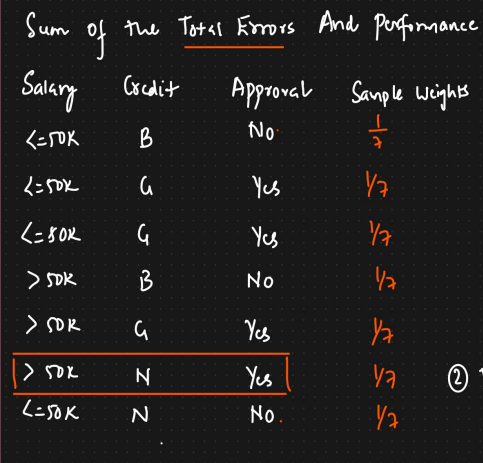
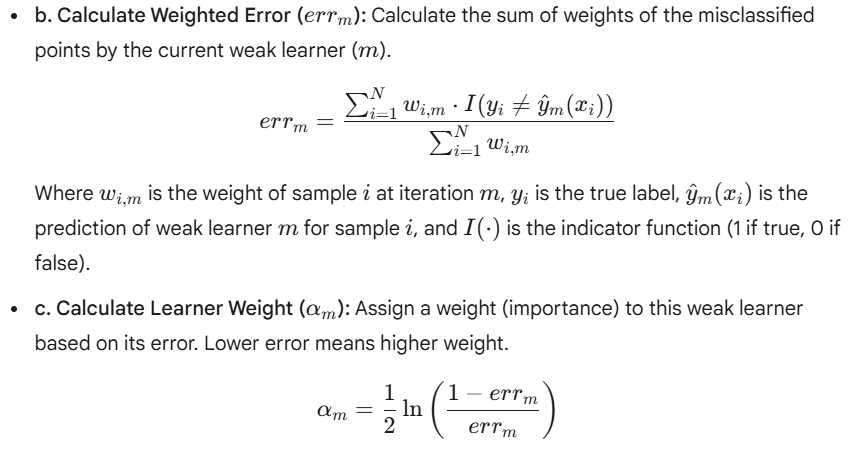
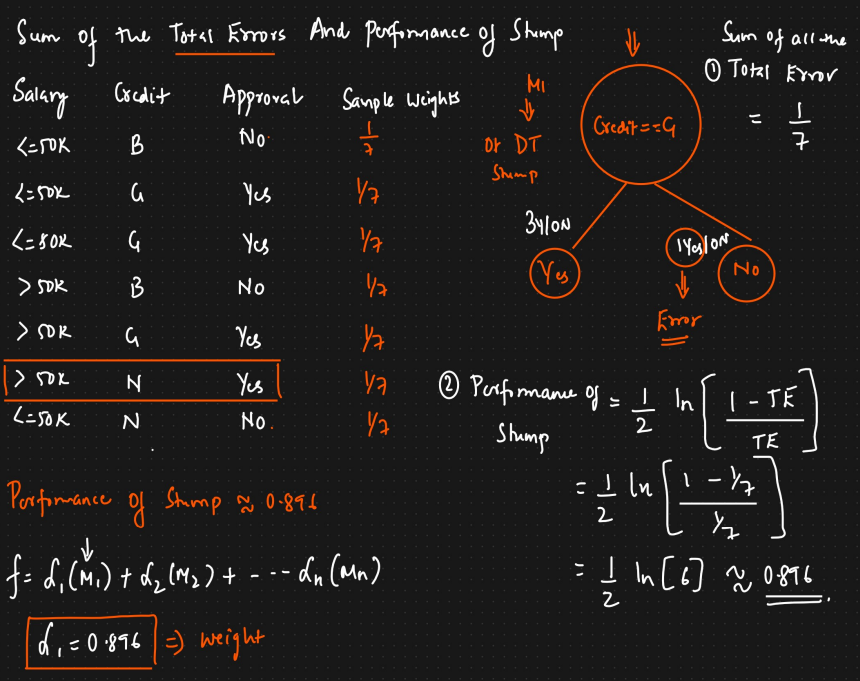
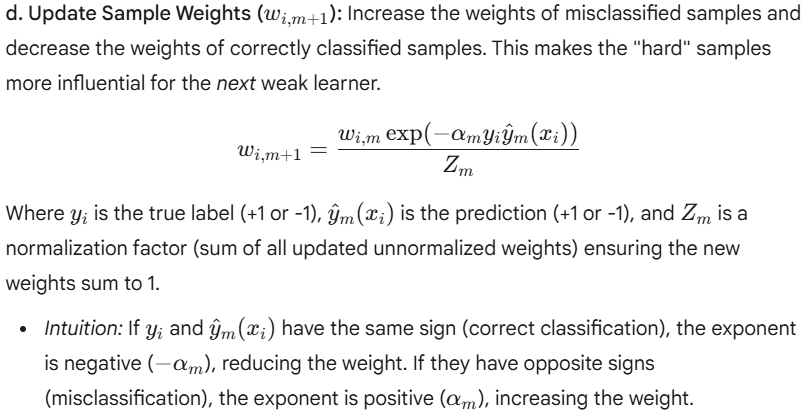
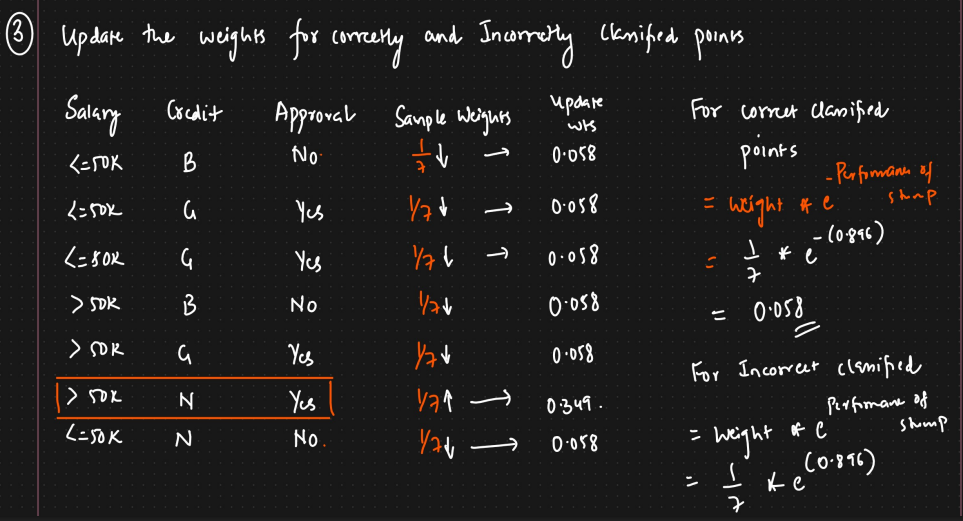
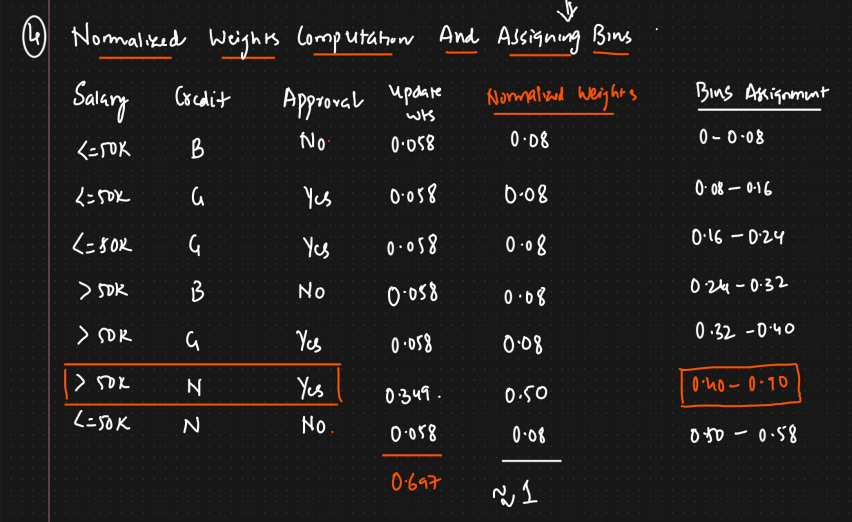
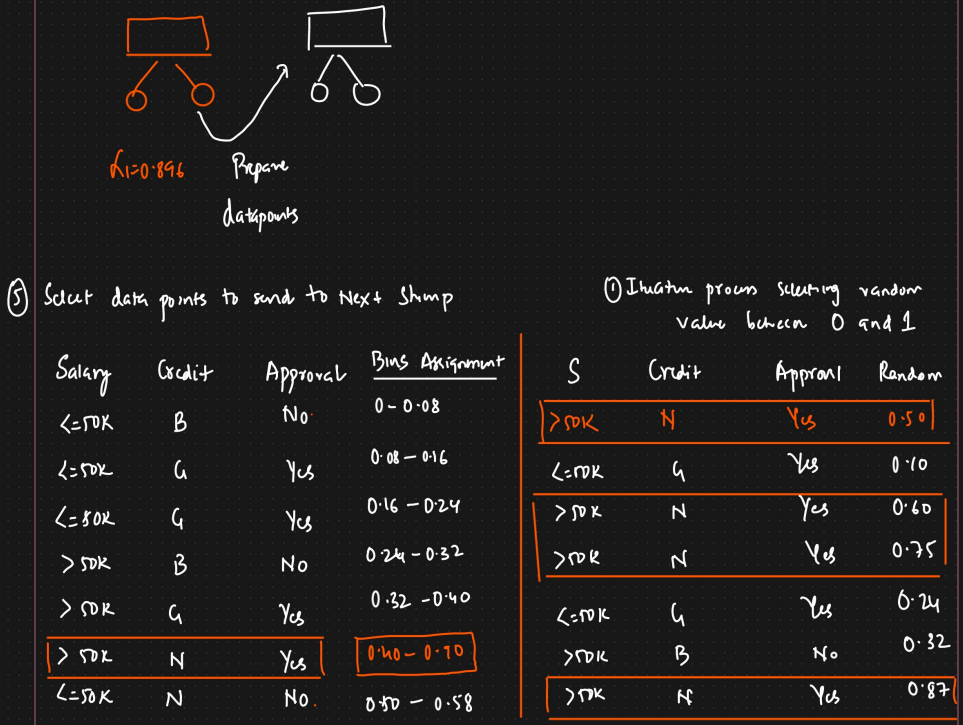
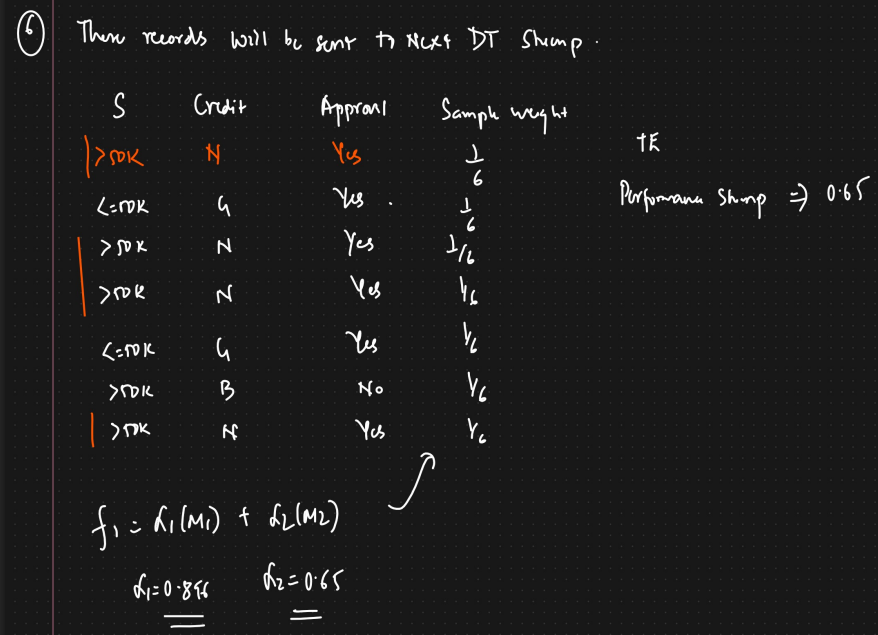
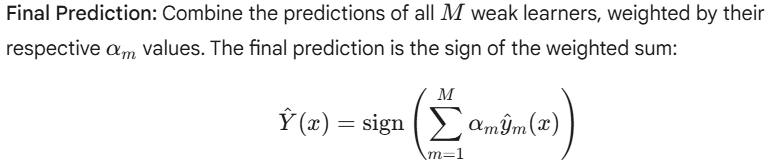
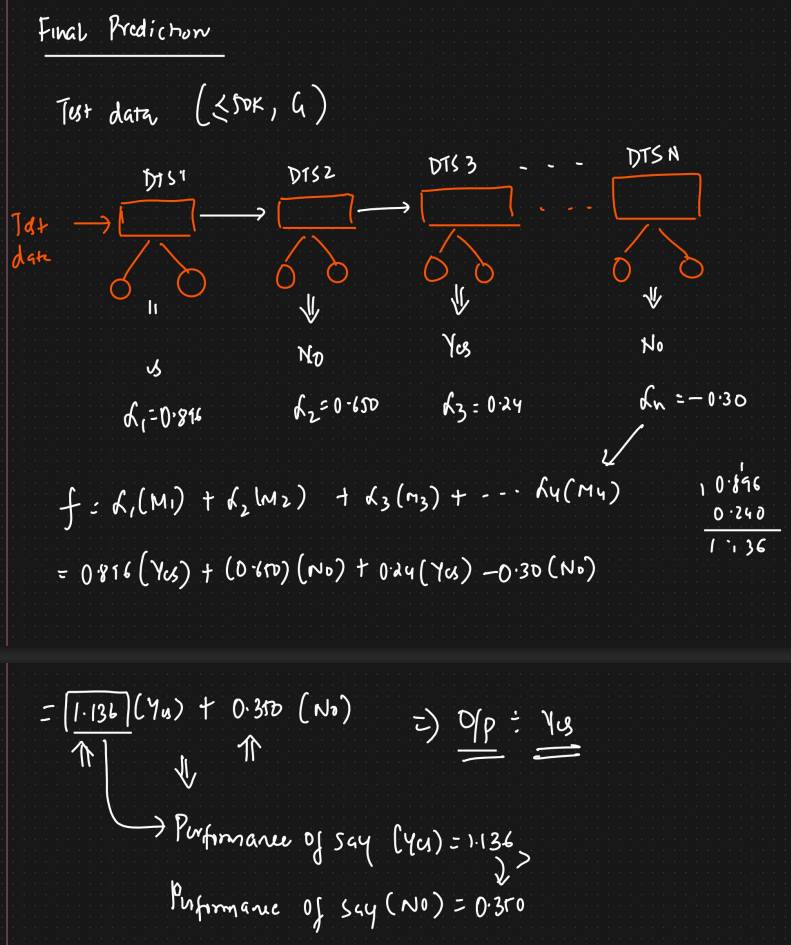


Decision Stump: Depth = 1.



**3. How AdaBoost Works (Algorithm Steps)**

Let's consider a binary classification problem (labels +1 and -1).

* **Initialization:** Assign equal weights to all data points in the training set. Let N be the number of data points, so each point starts with weight .
* **Iterative Training (for M rounds/estimators):**
  + **a. Train a Weak Learner:** Train a simple model (e.g., a decision stump - a one-level decision tree) on the *weighted* training data. The goal is to minimize the weighted classification error.
  + 
  + ****
  + ****
  + **Error is 1/7 because 1 row is misclassified.**
  + ****
  + ****
  + ****
  + **Iteratively select random values between 0 and 1 and chances of picking the point that is misclassified by the last learner is more.**
  + ****
  + ****
* ****
* ****

**4. Key Components**

* **Weak Learners:** Models that perform slightly better than random guessing (e.g., accuracy > 50% for binary classification). Decision stumps are very common, but other simple models can be used.
* **Sample Weights:** Indicate the importance of each data point during training. Adapts dynamically.
* **Learner Weights (α):** Indicate the contribution or "say" of each weak learner in the final prediction. More accurate learners get a bigger say.

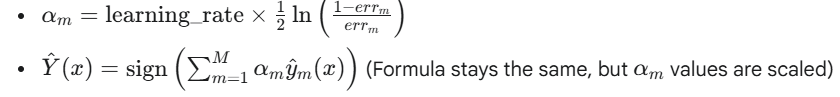
**5. Advantages of AdaBoost**

* **High Accuracy:** Often achieves very good performance on classification tasks.
* **Less Prone to Overfitting (Theoretically):** While it can overfit, especially with noisy data, it's generally considered more resistant than some other algorithms, as the weak learners are simple. The number of estimators is a key parameter to tune.
* **Versatility:** Can be used with various types of weak learners.
* **Feature Importance:** Can provide insights into feature importance based on how often features are selected by the weak learners (especially when using decision stumps).

**6. Disadvantages of AdaBoost**

* **Sensitive to Noisy Data and Outliers:** Since AdaBoost focuses on misclassified points, outliers or noisy data can receive very high weights, potentially skewing the model. Preprocessing or using variants less sensitive to outliers might be needed.
* **Computationally Intensive:** Training is sequential, meaning models cannot be trained in parallel like in Bagging (e.g., Random Forests). Each step depends on the previous one.
* **Can be Complex to Tune:** Requires tuning parameters like the number of estimators and potentially the complexity of the weak learner.

**7. Common Parameters (e.g., in Scikit-learn)**

* base\_estimator: The type of weak learner to use (default is usually DecisionTreeClassifier(max\_depth=1) - a **decision stump**).
* n\_estimators: The number of weak learners to train sequentially (M in the algorithm steps).
* learning\_rate: Shrinks the contribution of each weak learner by this factor (values between 0 and 1). It helps prevent overfitting by making the model learn more slowly. This is technically a modification to the original AdaBoost algorithm but is standard in implementations. The weight update and final prediction formulas are adjusted slightly:
* 

**8. Applications**

* Primarily used for **binary classification** tasks (e.g., face detection - Viola-Jones algorithm, spam filtering, medical diagnosis).
* Can be extended to **multi-class classification** (variants like AdaBoost.M1, SAMME, SAMME.R).
* Can be adapted for **regression** tasks (variants like AdaBoost.R, AdaBoost.RT).