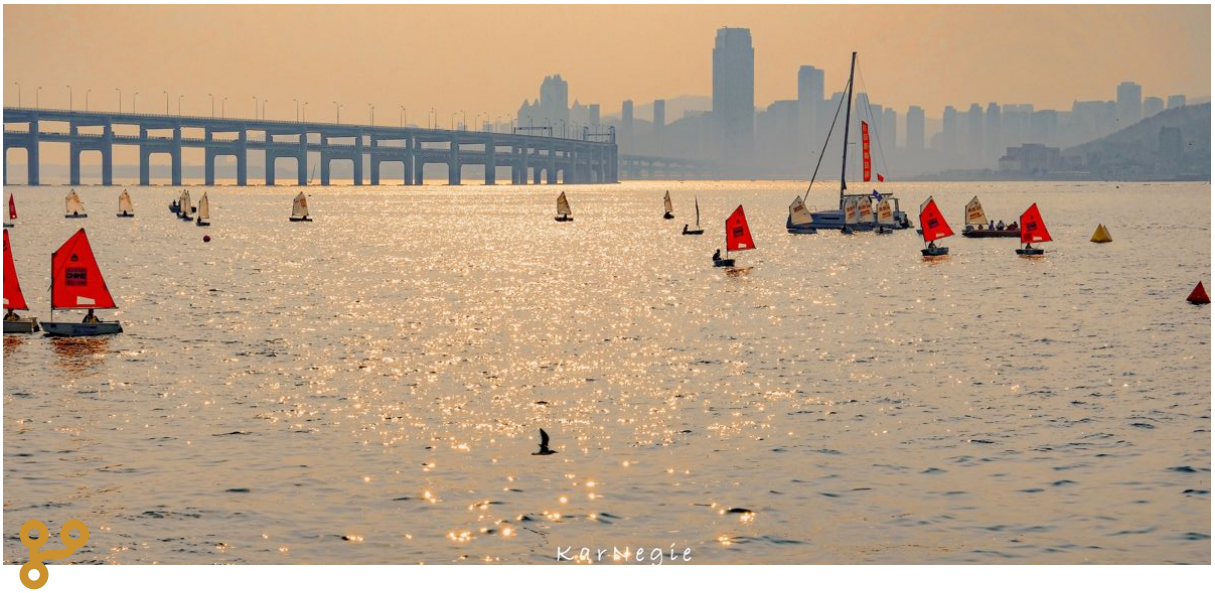


AdaBoost - Machine Learning 2

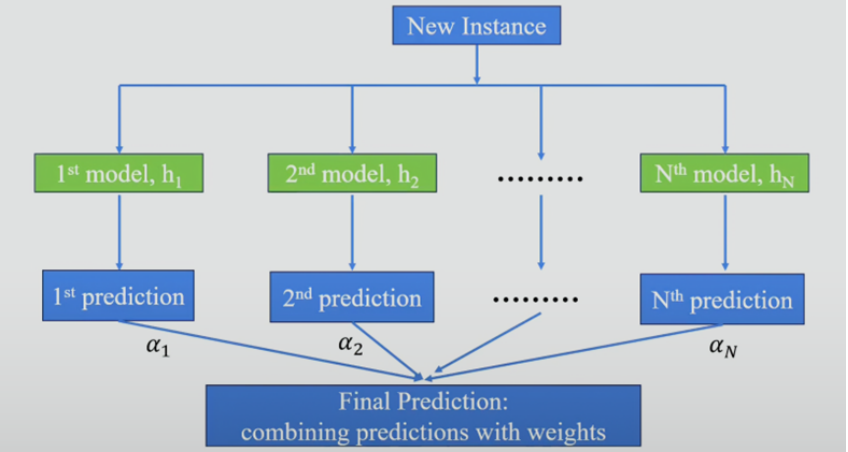
Problem 1:

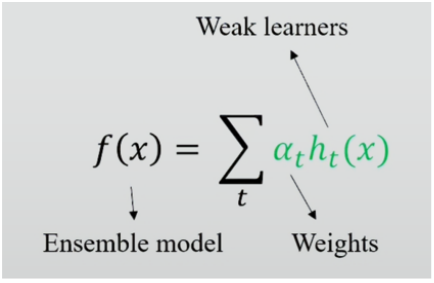
Đọc và hiểu overview về thuật toán boosting AdaBoost đã học trên lớp.

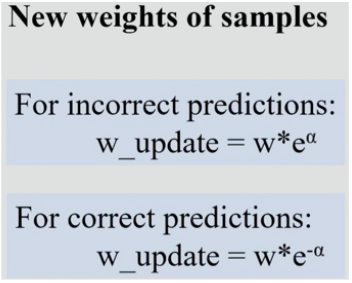
- Cho Flow Chart cách thức training, inference của AdaBoost cùng các thành phần khác như hình dưới:

Training

Inference







Thông tin bổ sung:

- Tại bước 2 của Flow chart: fit model bằng decision tree như đã học
- Tại bước 3 của Flow chart: tính alpha chính là tính "amount of say" như đã học bằng công thức:
- Công thức tính "New weights of samples" được sử dụng để tính lại weights mới cho toàn bộ các điểm trong bộ dữ liệu trước đó (1/N). Lưu ý khi tính lại weight cho các sample, tổng weights của tất cả các sample sẽ != 1 nên cần normalize lại tổng về 1.
- Hàm số "Ensemble model" tượng trưng cho việc sau khi tính toán ra đủ số lượng "Weak learners", ta tiến hành sử dụng toàn bộ những dự đoán của các "Weak learners" để đưa ra kết quả cuối cùng.

Problem 2:

Áp dụng kiến thức đã học về AdaBoost, thực hiện xây Decision Tree gốc và thực hiện Boosting 1 lần cho dataset sau:

Thích môn Tự nhiên	Thích môn Xã hội	Điểm môn toán	Thích ngành IT
Yes	Yes	8.2	**No**
No	Yes	8.3	**No**
No	Yes	8.4	**No**
Yes	No	8.7	**Yes**
Yes	No	8.9	**Yes**
No	Yes	9.2	**Yes**
Yes	No	9.2	**No**
Yes	Yes	9.3	**Yes**

- Cây được build chỉ gồm 1 root và và 2 lá (là Stump với depth = 1)
- Trường target là Thích ngành IT: Yes = 1; No = -1

Solution:

Step 1: Assign all observations with equal weights

$w_i = \frac{1}{N} = \frac{1}{8}$, for $i = 1, ..8$.

Thích môn Tự nhiên	Thích môn Xã hội	Điểm môn toán	Thích ngành IT	Sample Weight
Yes	Yes	8.2	**No**	0.125
No	Yes	8.3	**No**	0.125
No	Yes	8.4	**No**	0.125
Yes	No	8.7	**Yes**	0.125
Yes	No	8.9	**Yes**	0.125
No	Yes	9.2	**Yes**	0.125
Yes	No	9.2	**No**	0.125
Yes	Yes	9.3	**Yes**	0.125

Step 2: Build the first stump

Finding Best Splitting Attribute to build stump:

$$\text{Gini}(X) = 1 - (\tfrac{1}{2})^2 - (\tfrac{1}{2})^2 = \tfrac{1}{2}$$

Assume att1 = "Thích môn Tự nhiên":

$$\text{Gini}(X_{att1} = \text{Yes}) = 1 - (\tfrac{3}{5})^2 - (\tfrac{2}{5})^2 = 0.48$$

(3 Yes - Yes ; 2 Yes - No)

$$\text{Gini}(X_{att1} = \text{No}) = 1 - (\tfrac{1}{3})^2 - (\tfrac{2}{3})^2 = 0.44$$

(1 No - Yes; 2 No - No)

$$\implies \textbf{Gini}(X, att1) = \text{Gini}(X) - \frac{5}{8}\text{Gini}(X_{att1} = \text{Yes}) - \frac{3}{8}\text{Gini}(X_{att1} = \text{No})$$
$$= 0.03$$

Assume att2 = "Thích môn Xã hội":

$$\text{Gini}(X_{att2} = \text{Yes}) = 1 - (\tfrac{3}{5})^2 - (\tfrac{2}{5})^2 = 0.48$$

(2 Yes - Yes ; 3 Yes - No)

$\text{Gini}(X_{att2} = \text{No}) = 1 - (\frac{1}{3})^2 - (\frac{2}{3})^2 = 0.44$

(2 No - Yes; 1 No - No)

$$\implies \textbf{Gini}(X, att2) = \text{Gini}(X) - \frac{5}{8}\text{Gini}(X_{att2} = \text{Yes}) - \frac{3}{8}\text{Gini}(X_{att2} = \text{No})$$
$$= 0.03$$

For attribute Math Score:

- Finding Best Splitting Value

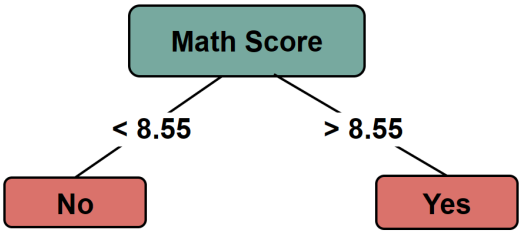
List Candidates	Delta Gini
8.25	0.07
8.35	0.17
8.55	**0.30**
8.8	0.13
9.05	0.03
9.2	0.03
9.25	0.07

Best splitting value for math score is **8.55**

$\implies \Delta\textbf{Gini}(X, score > 8.55) = 0.30$

Thus, attribute with highest Gini decrease will be choose to be stump.

We build a stump with Math Score being the only node:



First Stump in AdaBoost

We can make prediction for target variable based on the stump above:

Thích môn Tự nhiên	Thích môn Xã hội	Điểm môn toán	Thích ngành IT	Sample Weight
Yes	Yes	8.2	**No**	0.125
No	Yes	8.3	**No**	0.125
No	Yes	8.4	**No**	0.125
Yes	No	8.7	**Yes**	0.125
Yes	No	8.9	**Yes**	0.125
No	Yes	9.2	**Yes**	0.125
Yes	No	9.2	**No**	0.125
Yes	Yes	9.3	**Yes**	0.125

Green marked for observations that our Stump predict correctly, while the Red marked observations that Stump predict incorrectly.

For Adaboost, we will emphasize points that Stump incorrectly predict and less on the points that Stump correctly predict.

$$\text{Total Error} = \frac{1}{8}$$

$$\text{Amount of say} = \frac{1}{2} log(\frac{1 - \text{Total Error}}{\text{Total Error}}) = 0.97$$

We will calculate new weight for our observations:

$$\text{new weight}_i = \begin{cases} \text{old weight}_i \times e^{\text{-amount of say}} \ (\text{ for correct point }) \\ \text{old weight}_i \times e^{\text{amount of say}} \ (\text{ for incorrect point}) \end{cases}$$

Following this formular, we have:

Thích môn Tự nhiên	Thích môn Xã hội	Điểm môn toán	Thích ngành IT	Sample Weight	New Weight
Yes	Yes	8.2	**No**	0.125	0.047
No	Yes	8.3	**No**	0.125	0.047
No	Yes	8.4	**No**	0.125	0.047
Yes	No	8.7	**Yes**	0.125	0.047
Yes	No	8.9	**Yes**	0.125	0.047
No	Yes	9.2	**Yes**	0.125	0.047
Yes	No	9.2	**No**	0.125	0.33
Yes	Yes	9.3	**Yes**	0.125	0.047

However, we see that the sum of new weight is not equal to 1. Therefore, we have to normalize it by divide all weights to its sum.

Thích môn Tự nhiên	Thích môn Xã hội	Điểm môn toán	Thích ngành IT	Sample Weight	New Norm Weight
Yes	Yes	8.2	**No**	0.125	0.072
No	Yes	8.3	**No**	0.125	0.072
No	Yes	8.4	**No**	0.125	0.072
Yes	No	8.7	**Yes**	0.125	0.072
Yes	No	8.9	**Yes**	0.125	0.072
No	Yes	9.2	**Yes**	0.125	0.072
Yes	No	9.2	**No**	0.125	**0.499**
Yes	Yes	9.3	**Yes**	0.125	0.072

Problem 3:

LightGBM nhanh và vẫn đạt độ chính xác cao do model được sử dụng các kỹ thuật sau:

- **Histogram-Based Learning** (binning dataset)
- **Exclusive Feature Bundling** (gộp các categorical feature lại khi chúng “mutually exclusive”)
- **Gradient-Based One-side Sampling** (được dùng khi sample data mới sau mỗi lần build cây)
- Hiểu và giải thích tại sao các kỹ thuật này lại tốt.

Solution:

1. **Histogram-Based Learning (Binning Dataset)**:
   - Nguyên lý: Khi xử lý dữ liệu, LightGBM sử dụng histogram-based learning để tạo ra các histogram từ các feature numeric. Quá trình này bao gồm việc chia dữ liệu thành các bin (các khoảng giá trị liên tục). Thay vì sử dụng tất cả các giá trị riêng lẻ, việc chia thành các bin giúp giảm độ phức tạp tính toán.

   → Bằng cách này, LightGBM giảm bớt lượng dữ liệu cần xử lý và tính toán, làm tăng tốc độ huấn luyện mô hình. Ngoài ra, việc chia thành các bin cũng giúp mô hình hiểu được sự phân bố của dữ liệu một cách hiệu quả hơn, đồng thời giảm thiểu ảnh hưởng của nhiễu.
2. **Exclusive Feature Bundling**:
   - Nguyên lý: Khi xử lý các categorical features, LightGBM sử dụng exclusive feature bundling để gộp các feature lại với nhau nếu chúng "mutually exclusive", tức là không thể xảy ra cùng một thời điểm. Ví dụ, nếu một người chỉ có thể thuộc vào một trong các nhóm: "trẻ em", "thanh thiếu niên", "người trưởng thành", thì các feature liên quan đến nhóm tuổi có thể được gộp lại thành một feature duy nhất.

   → Gộp các feature giúp giảm số lượng feature, giúp mô hình huấn luyện nhanh hơn và giảm thiểu nguy cơ overfitting. Nó cũng giảm độ phức tạp của mô hình bằng cách loại bỏ sự trùng lặp thông tin giữa các feature.
3. **Gradient-Based One-side Sampling**:
   - Nguyên lý: Kỹ thuật này được áp dụng khi sample data mới sau mỗi lần xây dựng cây. Thay vì chọn ngẫu nhiên, nó tập trung vào việc chọn ra các điểm dữ liệu quan trọng. Cụ thể, nó tập trung vào việc lấy mẫu các điểm dữ liệu mà gradient lớn, tức là những điểm dữ liệu có ảnh hưởng lớn đến quá trình xây dựng cây.

   → Bằng cách tập trung vào các điểm dữ liệu quan trọng, Gradient-Based One-side Sampling giúp giảm thời gian huấn luyện mà vẫn đảm bảo độ chính xác của mô hình. Nó làm cho quá trình xây dựng cây hiệu quả hơn bằng cách tối ưu hóa quá trình lấy mẫu, đồng thời giảm thiểu số lượng dữ liệu không cần thiết.

Problem 4:

[Optional] So sánh CatBoost, XGBoost, LightGBM trong 3 aspects sau: Cách decision tree được build; categorical feature được handle như thế nào; Cách sampling

- Với mỗi đặc điểm của mỗi thuật toán phải hiểu sơ lược và giải thích được

Solution:

CatBoost, XGBoost, LightGBM Comparison:

Aspect	CatBoost	XGBoost	LightGBM
Tree Building Approach	Utilizes Symmetric Decision Trees. During tree construction, all nodes are considered to choose the tree with the greatest total error reduction.	Uses Greedy Algorithm. At each step, the tree is built by selecting the optimal feature and threshold to minimize the error.	Employs Leaf-Wise (Best-First) tree building. LightGBM constructs trees by finding the node with the largest error reduction.
Handling Categorical Features	Automatically handles categorical features by encoding them into integers based on their order in the dataset.	Requires one-hot encoding or ordinal encoding before usage.	Supports categorical features directly without the need for pre-encoding.
Sampling Approach	Utilizes Ordered boosting. CatBoost uses previous trees to determine the sampling order for the next tree, minimizing overfitting.	Uses random sampling. Each tree is trained on a randomly sampled subset of the training data.	Uses Gradient-Based One-side Sampling to focus on important data points, reducing training time and ensuring accuracy.

vietsub:

Aspects	CatBoost	XGBoost	LightGBM
Tree Buiding	Sử dụng Symmetric Decision Trees. Khi xây dựng cây, tất cả các node đều được xem xét để chọn ra cây có tổng giảm lỗi lớn nhất.	Sử dụng Greedy Algorithm. Tại mỗi bước, cây được xây dựng bằng cách chọn ra feature và threshold tối ưu nhất để giảm lỗi.	Sử dụng Leaf-Wise (Best-First) tree building. LightGBM xây dựng cây bằng cách tìm node có giảm lỗi lớn nhất.
Categorical features handling	Tự động xử lý categorical feature bằng cách mã hóa chúng thành các số nguyên dựa trên thứ tự trong tập dữ liệu.	Yêu cầu one-hot encoding hoặc ordinal encoding trước khi sử dụng.	Hỗ trợ trực tiếp categorical feature mà không cần mã hóa trước.
Tree Sampling	Sử dụng Ordered boosting. Tức là CatBoost sử dụng các cây trước đó để xác định thứ tự lấy mẫu cho cây tiếp theo, giảm thiểu việc overfitting.	Sử dụng random sampling. Mỗi cây được huấn luyện trên một tập dữ liệu con được lấy mẫu ngẫu nhiên từ tập huấn luyện.	Sử dụng Gradient-Based One-side Sampling để tập trung vào các điểm dữ liệu quan trọng, giảm thời gian huấn luyện và đảm bảo độ chính xác.