Catboost

* Categorical Boosting
* Russian Internet company → Yandex (2017)
* Handle categorical colums >>> XGBoost
* Has automatic encoding (Better normal encoding)
* Large dataset with majority categorical features

* Symmetric trees (no of branches & nodes is equal)
* GPU support → training speed ↑
* SHAP (Shapely Additive Explanation)
(Inbuilt visualization feature
feature importance

Mathematical encoded

Index	X_1	X_2	Y	OTS X_1	OTS X_2	$\hat{Y}$
1	Red	High	1	0.5	0.5	0
2	Blue	Low	0	0.5	0.5	0
3	Green	High	1	0.5	1	0
4	Red	Medium	0	1	0.5	1
5	Green	low	1	1	0	1
6	Blue	Medium	0	0	0	0
7	Red	High	1	0.5	1	0
8	Blue	Low	0	0	0.5	0

① Initialize weights

All rows/instances ⟶ assign equal weights

② Encoding (Ordered Target Statistics)

(Mean of previous occurences)

X_1

(a) Red ⇒ Index 1 ⇒ No previous occurence
We take it as 0.5

(1, 4, 7) Index 4 ⇒ Index 1 :- 1

$$\text{Mean} = \frac{1}{1} = 1$$

Index 7 ⇒ Index 1 :- 1
Index 4 :- 0

$$\text{Mean} = \frac{1+0}{2} = \underline{0.5}$$

(b) Blue : Index 2 ⇒ No prev occurences ⇒ 0.5

(2, 6, 8) Index 6 ⇒ Index 2 : 0

$$\text{Mean} = \frac{0}{1} = \underline{0}$$

Index 8 ⇒ Index 2 :- 0
Index 6 :- 0

$$\text{Mean} = \frac{0+0}{2} = \underline{0}$$

(c) Green : Index 3 ⇒ No prev occurrence ⇒ 0.5

(3, 5) Index 5 ⇒ Index 3 :- 1

$$\text{Mean} = \frac{1}{1} = 1.$$

Q2: (a) High Index 1 ⇒ 0.5.

(1, 3, 1) Index 3 ⇒ $\text{Mean} = \frac{1}{1} = 1$

Index 7 ⇒ $\text{Mean} = \frac{1+1}{2} = 1$

(b) low (2, 5, 8) ⇒ Index 2 ⇒ 0.5

Index 5 ⇒ $\frac{0}{1} = 0$

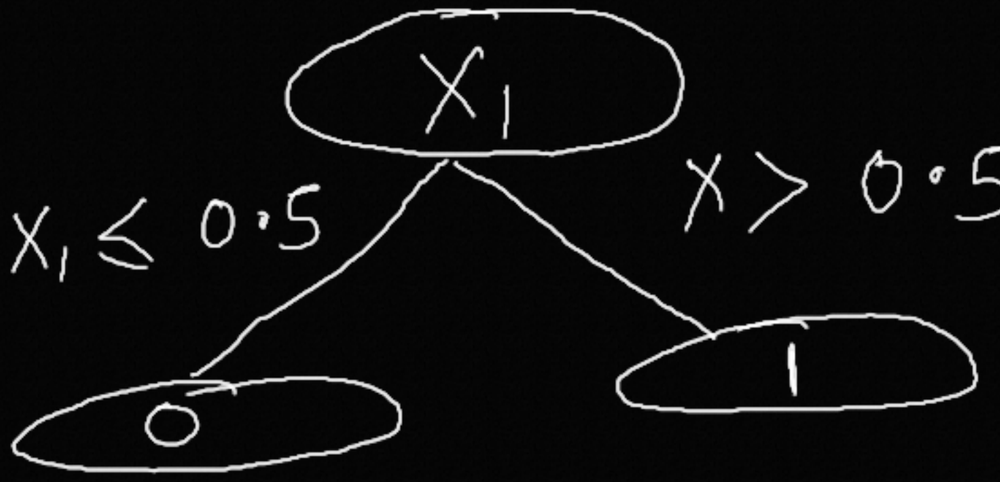
Index 8 ⇒ $\frac{0+1}{2} = 0.5$

(c) Medium (4, 6) ⇒ Index 4 ⇒ 0.5

Index 6 ⇒ $\frac{0}{1} = 0$

New X_1 & X_2 (encoded)

③ Train first weak learner



DT do predictions

we got the predictions

④ Calculate Residuals

$$\text{Residuals} = \underset{\text{actual}}{y} - \underset{\text{predicted}}{\hat{y}}$$

Calculate all residuals

⑤ Now next weak learner is trained to correct errors of first weak learner

⑥ Update Predictions

Updated Value = Initial prediction + α × weak learner prediction

Repeat until our error become small

All things explained in GB video.

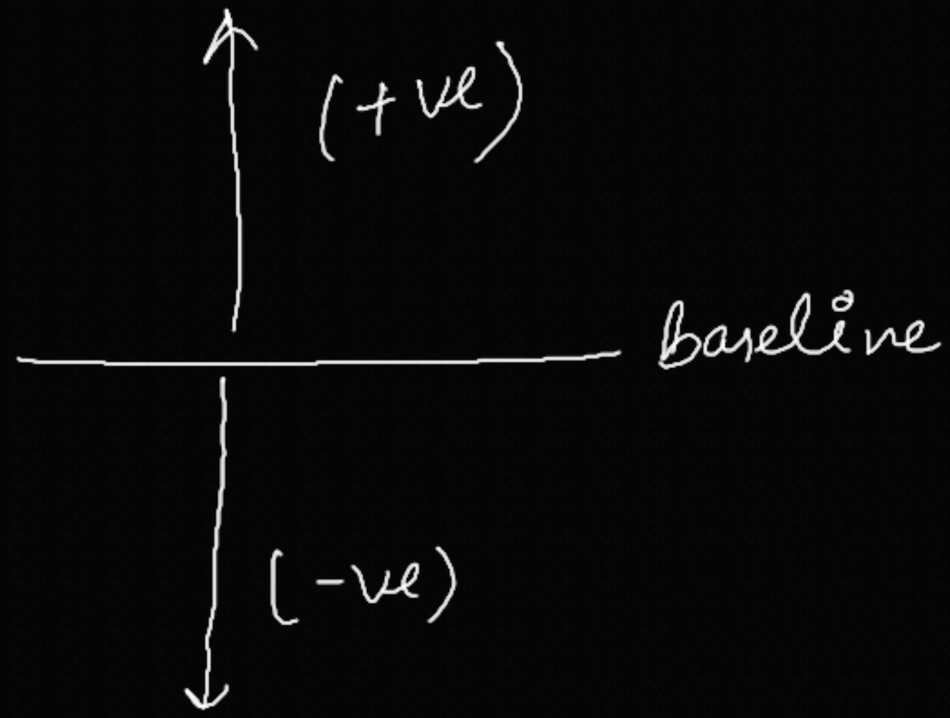
SHAP

Carat
1st → −948
4th → 2049

→ absolute mean

↓

feature importance



When to Use

1. Categorical features ✓
2. Large & complex data
3. Minimal tuning reqd
4. Regularization provided (L2 penalty)
   - → Multicollinearity
5. Encoding & Imbalanced dataset

When Not to Use

Small dataset
- → overkill

Real-time prediction X
- → computational intensive

Not good for sparse data X

like text-data (NLP data)