

XGBoost



Ensemble Methods Overview

- 1. Bagging:** Combines predictions from multiple models trained on different random subsets of the data to reduce variance.
- 2. Boosting:** Trains models sequentially, where each model focuses on correcting the errors of the previous one, reducing bias.
- 3. Stacking:** Combines the outputs of multiple models (even different types) into a "meta-model" for better predictions.

XGBoost קיצור של *Extreme Gradient Boosting* הוא אחד האלגוריתמים החזקים למידת מבוניה.

הוא מבוסס על עצי החלטה ועובד בטכניקה של – **Boosting** בלם, בניית סדרת מודלים פשוטים (לרוב עצים קטנים) שמתקנים יחד את הטעויות של השני.

- הוא מדויק מאוד ומנצח בתחרויות Kaggle
- מתאים גם לביעות סיווג וגם לביעות רgression
- ממודד היטב עם נתוניים חסריים, פיצ'רים קטגוריים ו-**Overfitting**

מטרה : לקבוע איך XGBoost ימדד ביצועים במהלך האימון.

אפשרויות נפוצות:

- 'logbinary' / multi-class ל-
הסתברויות
- 'error' – אход טוויות(1-0)
- 'mlogloss' – log loss
למספר קלאסים
- 'rmse', 'mae'

objective

מטרה: להגדיר את סוג הבעיה שהמודל פותר.

אפשרויות נפוצות:

'binary:logistic' סיווג בינארי עם הסתברויות

'multi:softprob' סיווג מרובה מחלקות עם הסתברויות

'multi:softmax' סיווג מרובה מחלקות עם תחזית סופית

'reg:squarederror' רגרסיה רגילה

tree_method

tree_method

מטרה: לקבוע איך לבנות את עצי החלטה (איזה אלגוריתם).

אפשרויות נפוצות:

'auto' בירית מחדל (יבחר לפי הדאטה)

'exact' חישוב מדויק, איטי (לדאטה קטן)

'approx' חישוב מקורב, מהיר יותר

'hist' מהיר במיוחד לדאטה גדול, תומך

ב-'categorical'

'gpu_hist' מהיר מאוד על GPU

פרמטר	תפקיד	ערך נפוץ	הערות שימושיות
objective	מגדיר את סוג הבעיה	'binary:logistic', 'multi:softprob', 'reg:squarederror'	חובה להתאים לבעה (סיווג/רגרסיה)
eval_metric	מדד להערכת ביצועי המודל	'logloss', 'mlogloss', 'rmse', 'mae'	משפיע רק על מדידות בזמן אימון
tree_method	שיטת לבנית עצים	'auto', 'hist', 'approx', 'gpu_hist'	'hist' תומך בקטגוריים ומאיצ' ביצועים
n_estimators	מספר העצים	100–1000	יוטר עצים = מודל חזק יותר (ולעתים איטי יותר)
max_depth	עומק מקסימלי לכל עץ	3–6	עומק גדול מדי = סכנת overfitting
learning_rate	קצב הלמידה של כל עץ	0.01–0.3	קטן = למידה איטית ובטוחה
subsample	אחוז דגימת השורות לכל עץ	0.5–1.0	עזר להכללה ומניעת overfitting
colsample_bytree	אחוז דגימת העמודות לכל עץ	0.5–1.0	טוב אם יש הרבה פיצ'רים
gamma	מינימום רוח נדרש לפיזול	0, 1, 5	מעלה = פחות פיצולים = מודל פשוט יותר
reg_alpha	רגוליזציה (1) עונש על גודל משקלים)	0, 0.1, 1	מעודד משקלים = 0 (sparsity))
reg_lambda	רגוליזציה 2)	1, 5, 10	עזר ליציבות המודל
early_stopping_rounds	עצירה אוטומטית אם אין שיפור	10–50	עובד עם eval_set, overfitting מנע