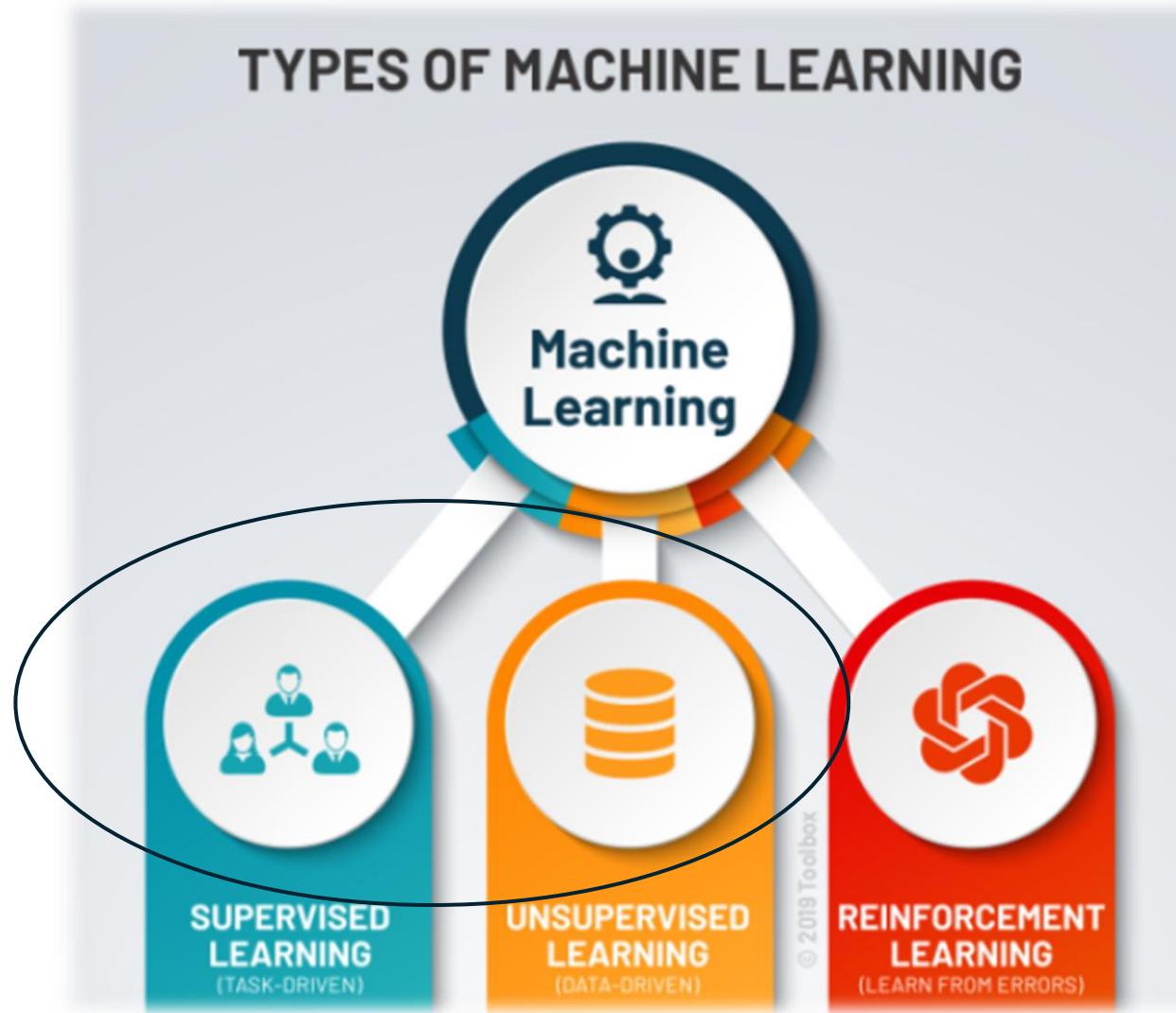
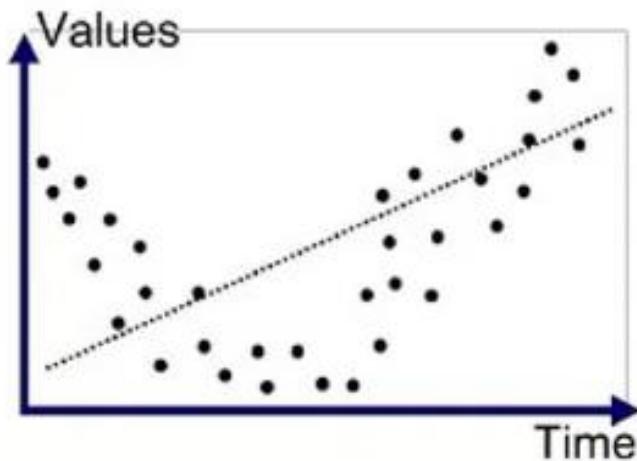


Machine Learning

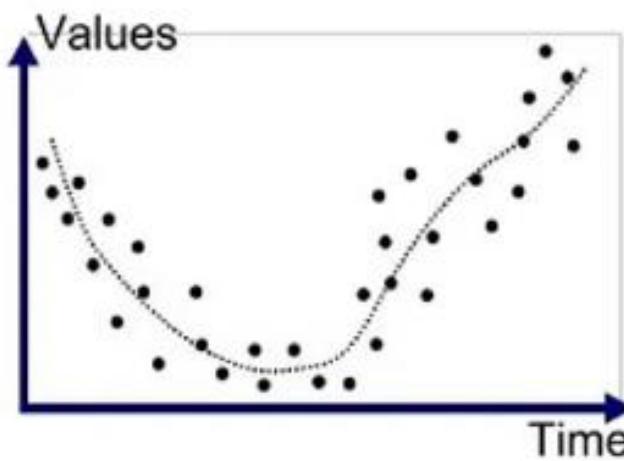
Types of ML Models



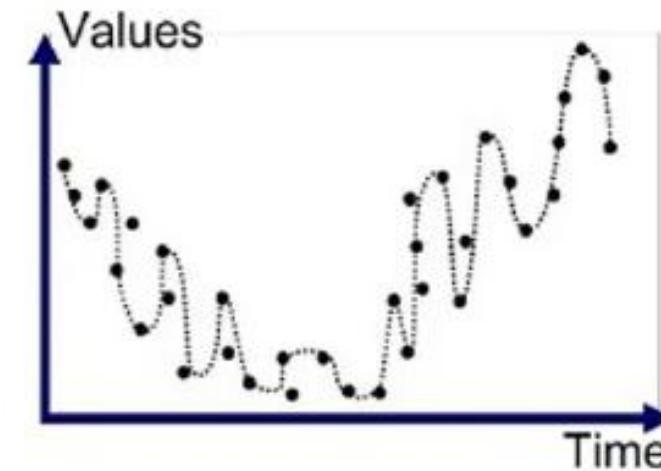
What is underfitting and overfitting in machine learning?



Underfitted

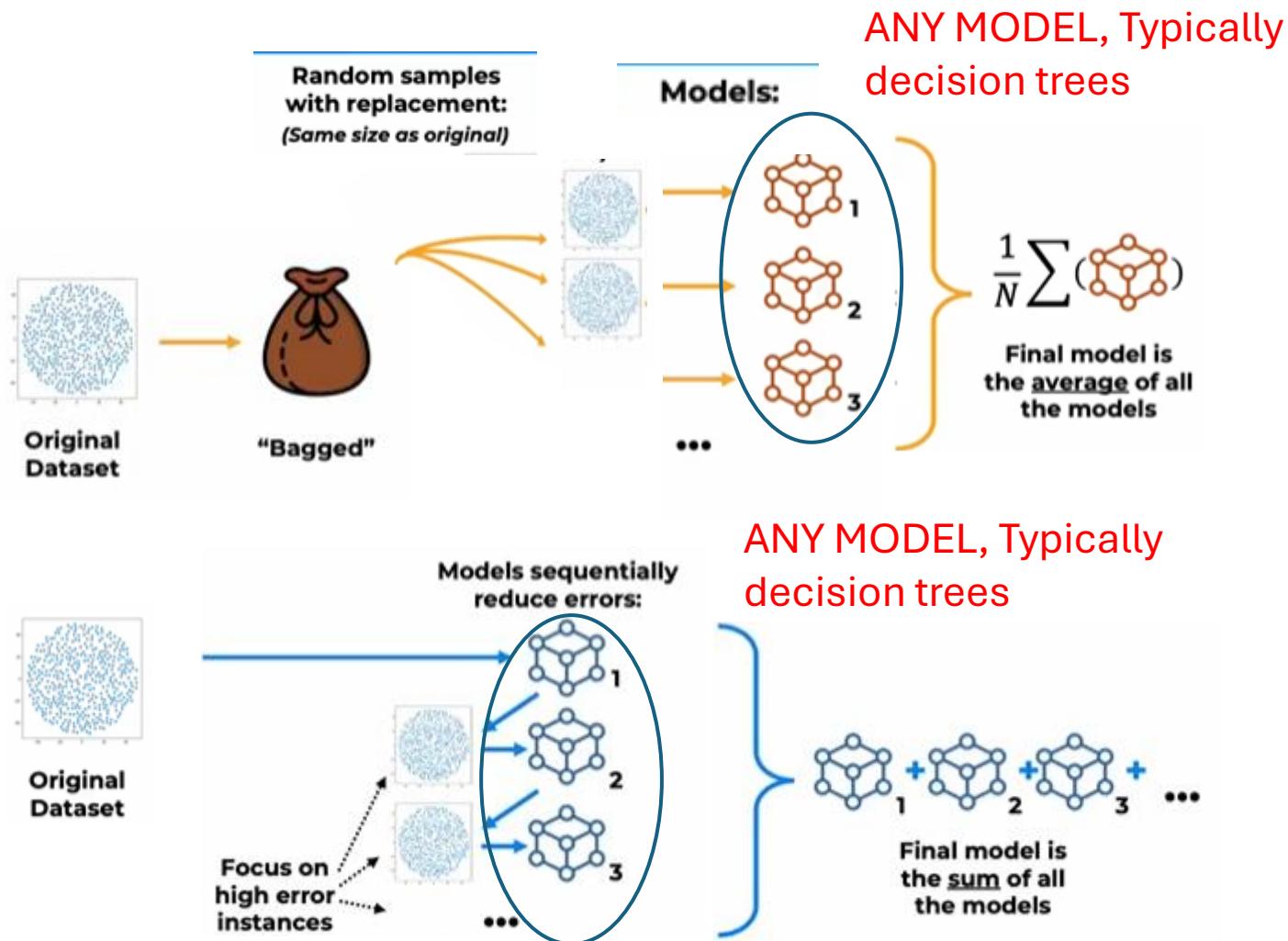
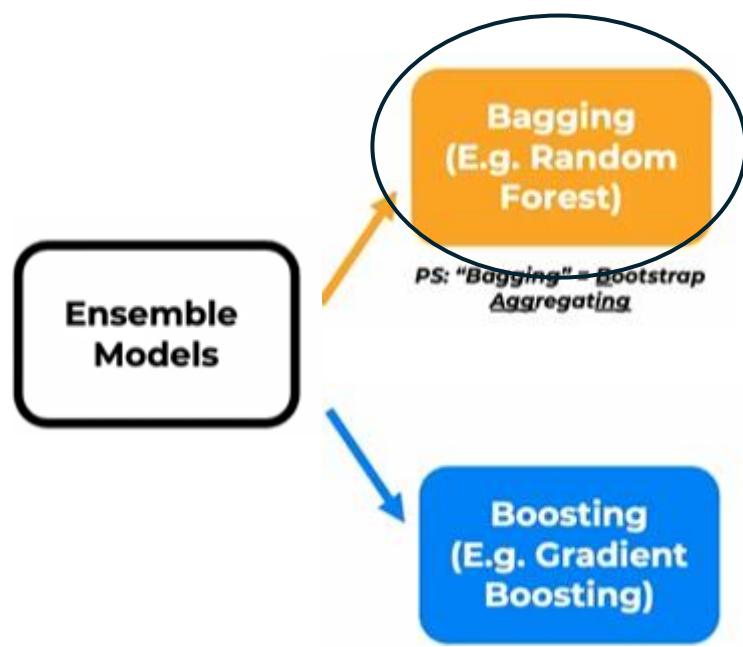


Good Fit/R robust



Overfitted

Ensemble Models



Ensemble Models

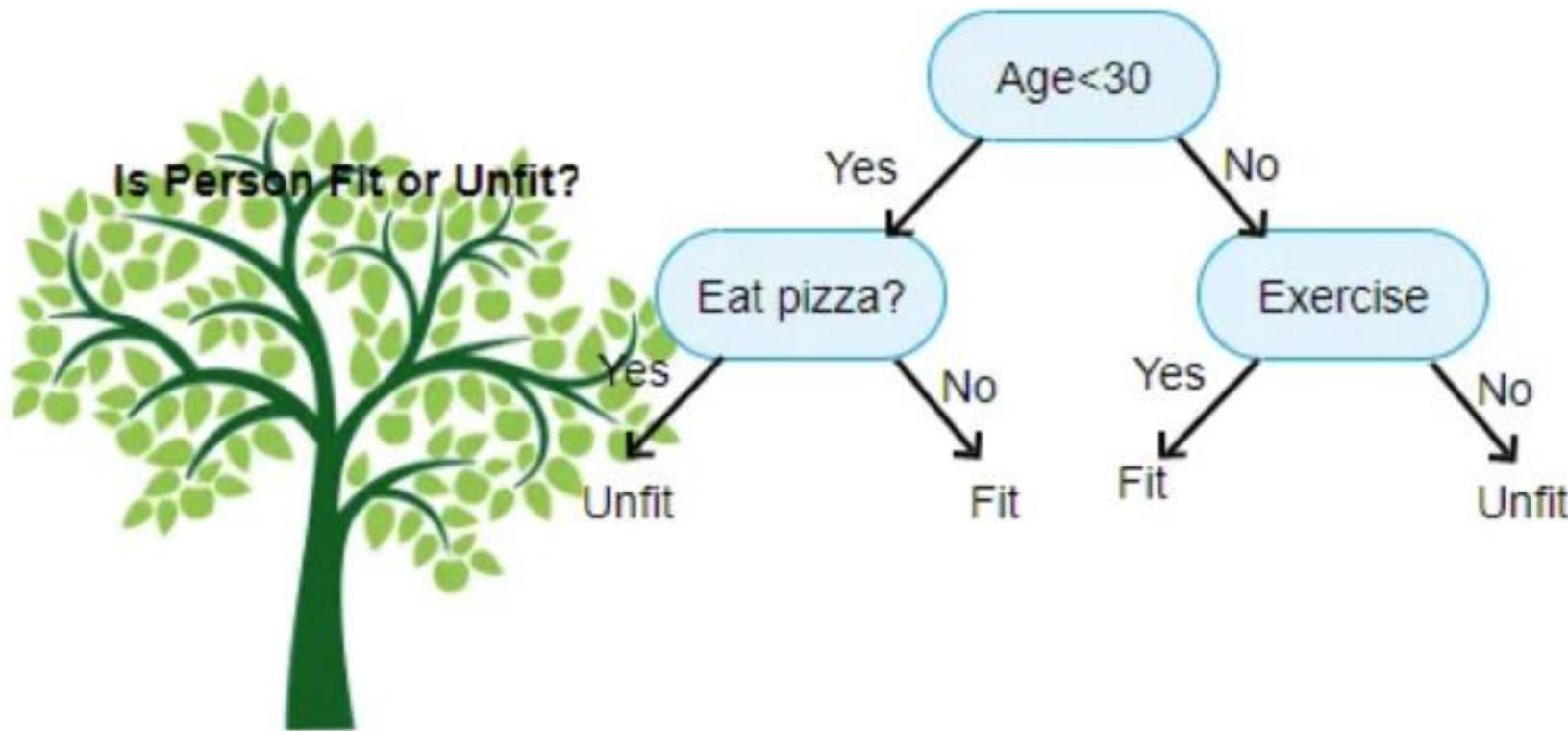
Bagging (Bootstrap Aggregating) ◆

דוגמה: Random Forest

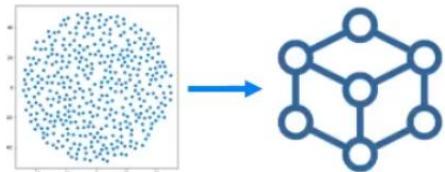
מה קורה בפועל?

- יוצרים כמה קבוצות נתונים על ידי דגימה אקראיית עם החזרה (Random samples with replacement) מתוך הדאטה המקורית.
- מאמנים כמה מודלים שונים במקביל על כל אחת מהדגימות.
- המודל הסופי הוא ממוצע (לרגסיה) או הצבעת רוב (לסיוג) של כל המודלים.
- עוזר למנוע overfitting.
- מודלים פועלים במקביל (ניתן להריץ אותם בצורה מקבילה).

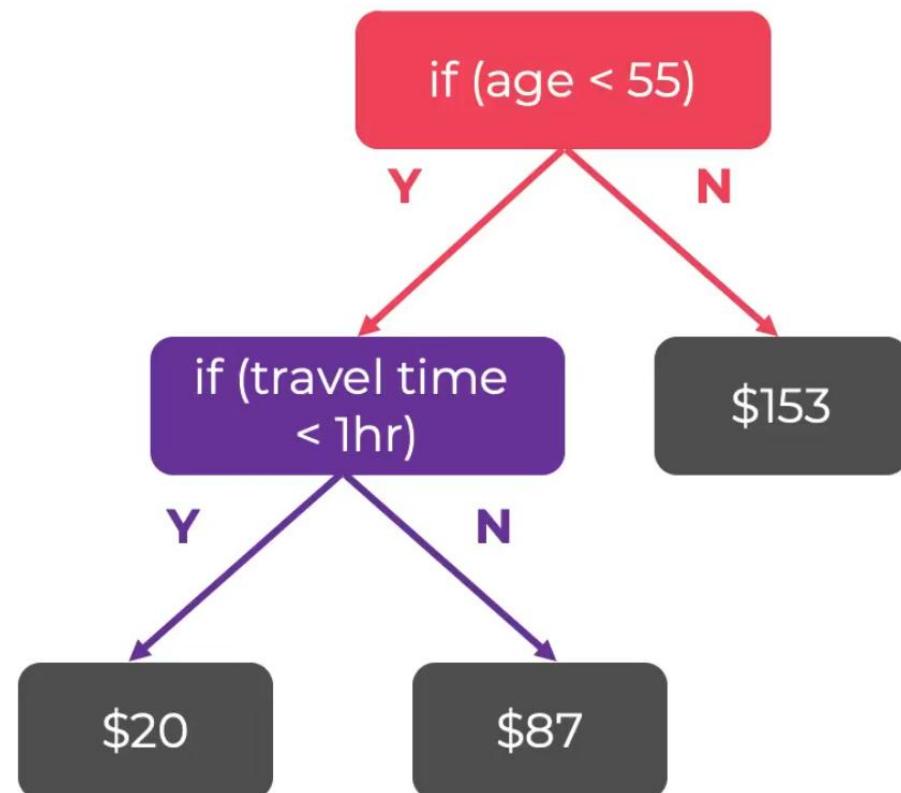
Ensemble Models (CART)



Decision Trees



Question: How much do customers spend at your store?



עץ החלטה:

1. צומת ראשון (שורש):

- שואל: האם הגיל > 55?

- אם כן (Y) → ממשיכים שמאלה.

- אם לא (N) → התחזית היא \$153.

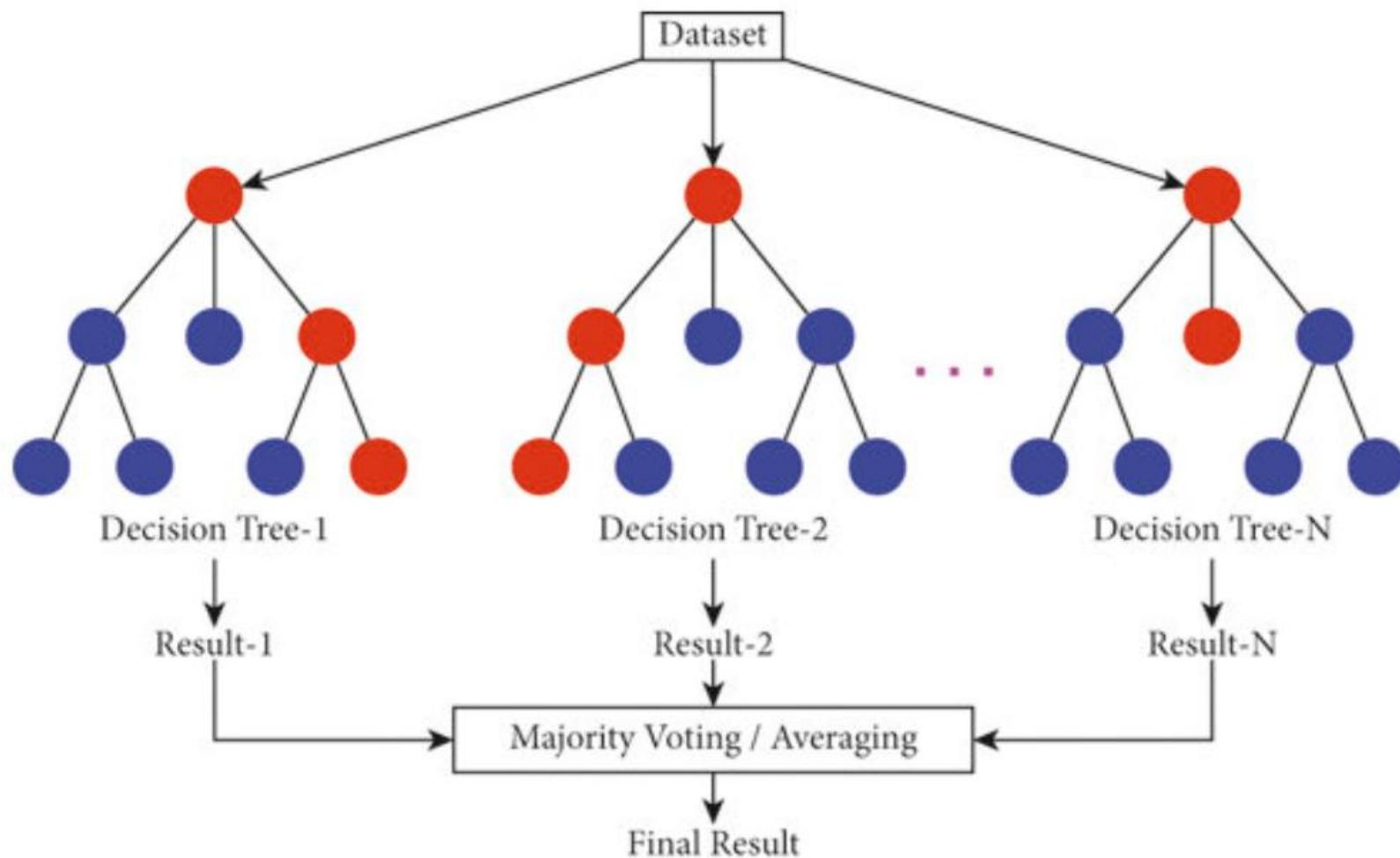
2. צומת שני (אם הגיל > 55):

- שואל: האם זמן הנסיעה > שעה?

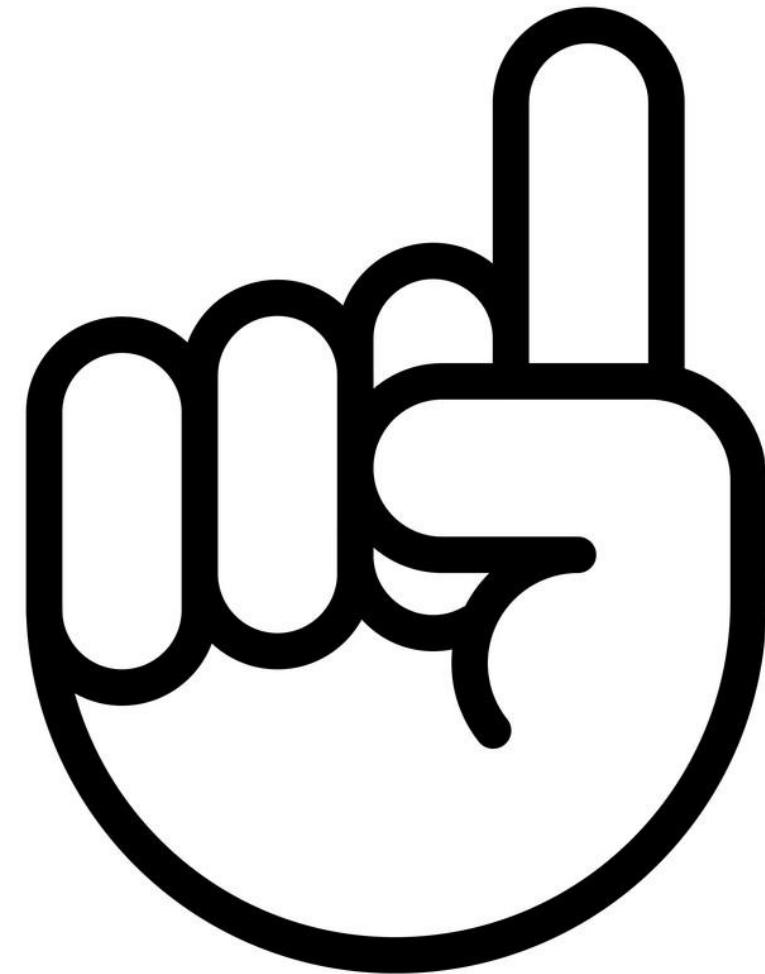
- אם כן (Y) → התחזית היא \$20.

- אם לא (N) → התחזית היא \$87.

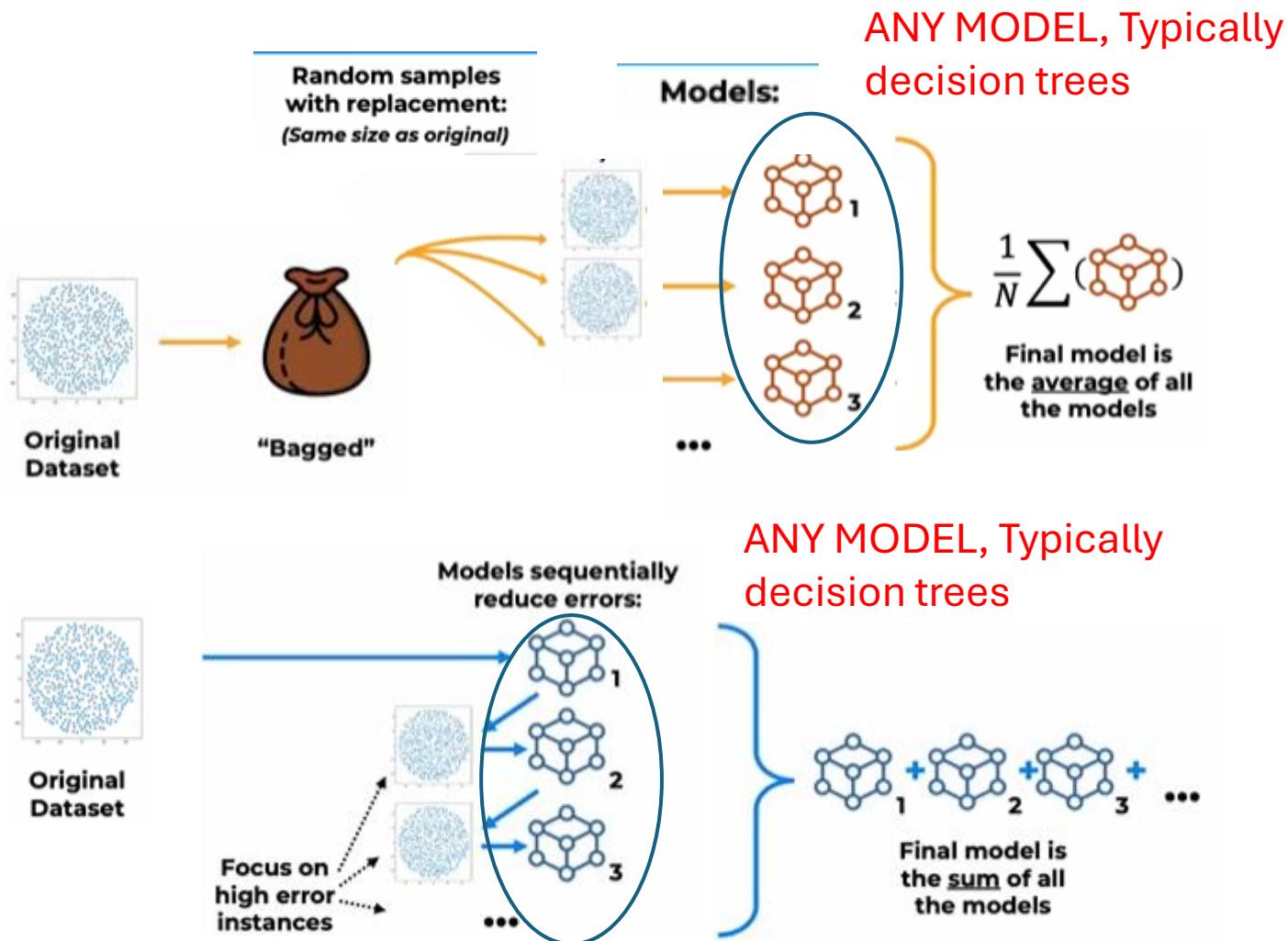
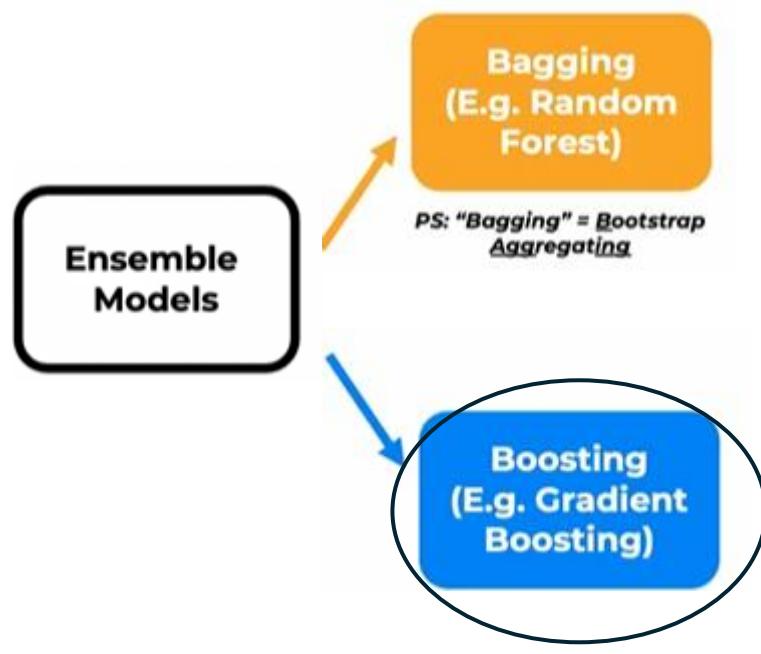
Ensemble Models (Random Forest)



Random Forest



Ensemble Models



Ensemble Models

Boosting ◆

דוגמה: Gradient Boosting

מה קורה בפועל?

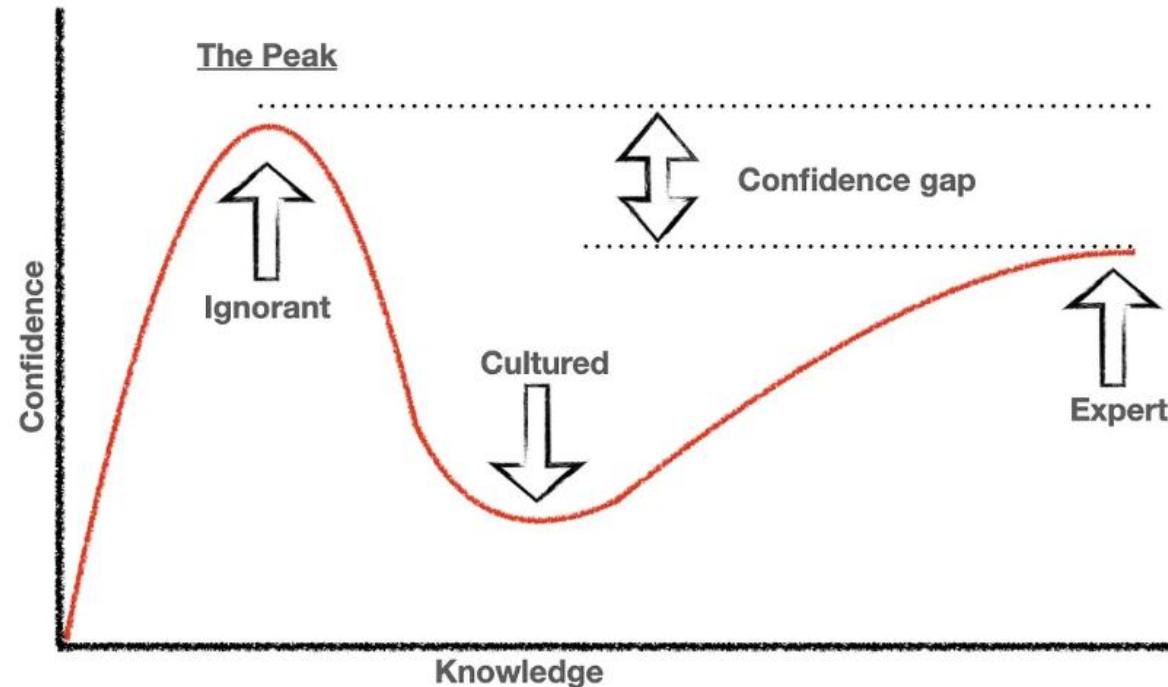
- מתחילה עם מודל פשוט ראשון.
- כל מודל חדש מתקן את השגיאות של המודל הקודם.
- הDATA שדורש יותר תשומת לב (טעויות רבות) מקבל משקל גבוה יותר בשלבים הבאים.
- המודל הסופי הוא סכום משוקל של כל המודלים.

יתרונות:

- מפחית bias.
- מושג ביצועים חזקים מאוד במקרים רבים.
- רגש יותר ל-*overfitting*, ולכן נדרש טיפול נכון (למשל, עצים רדומים או רגולריזציה).

Gradient Boosting

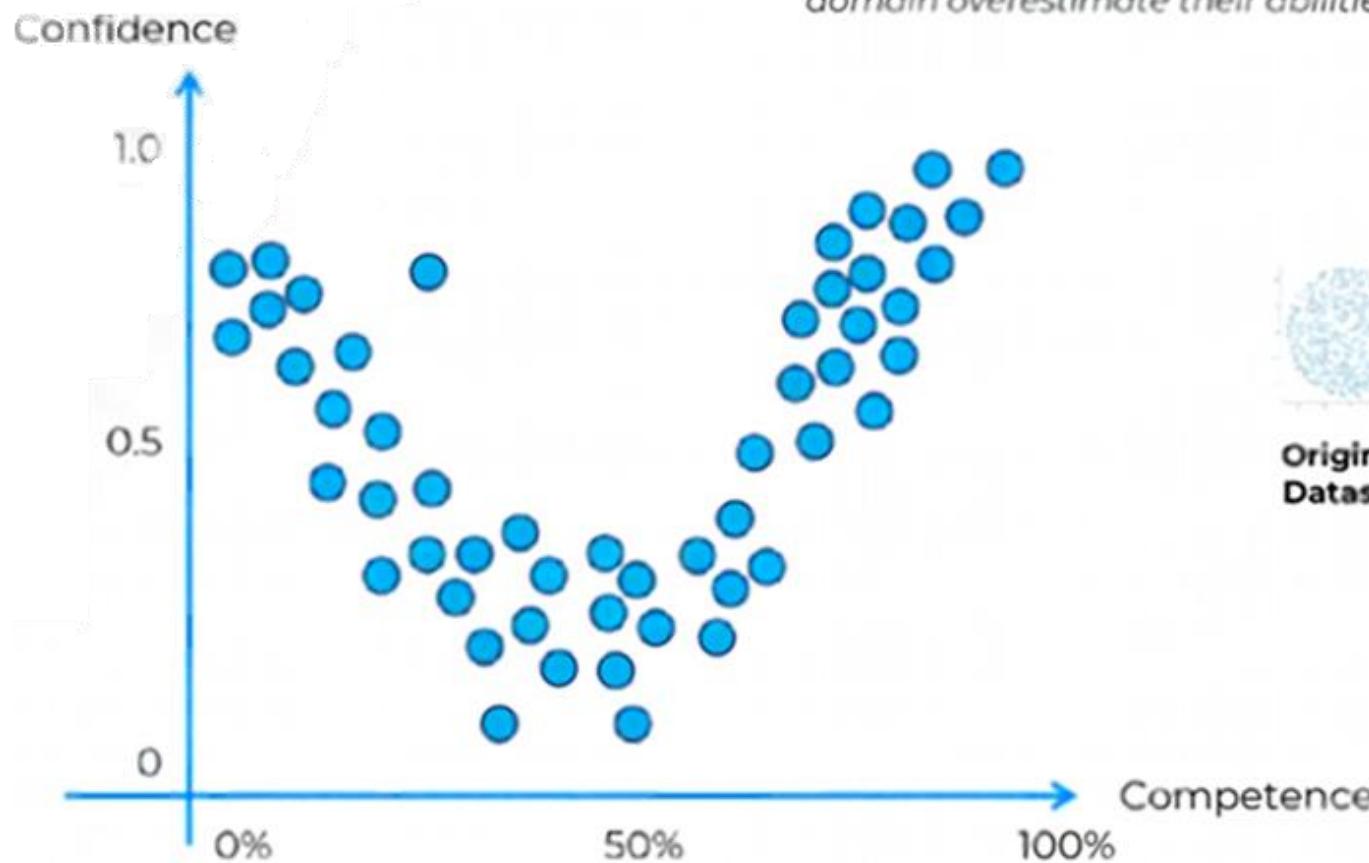
Dunning-Kruger effect



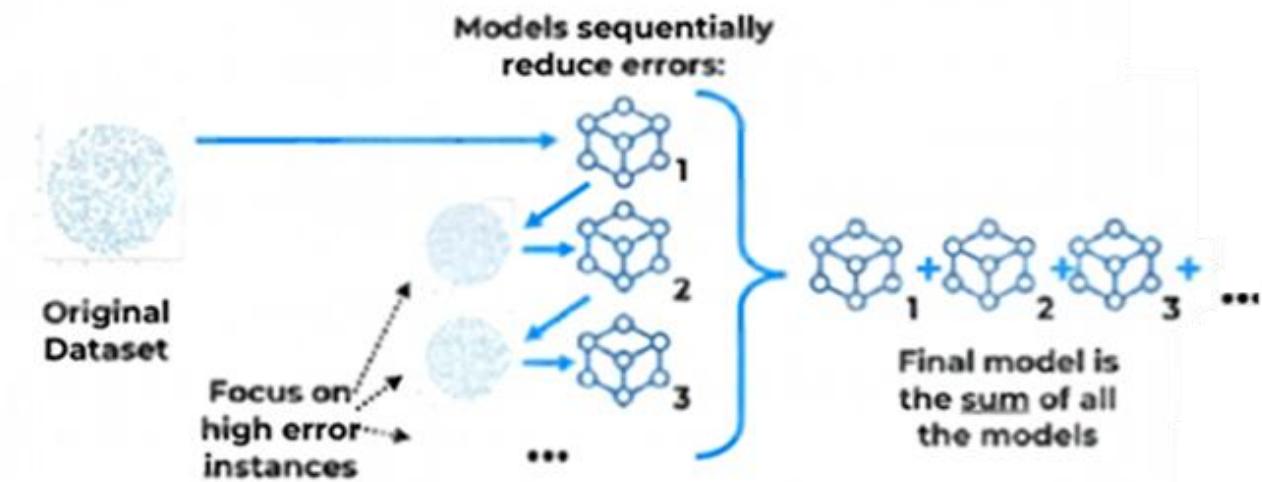
אפקט דאינינג-קרוגר: אנשים עם יכולת מוגבלת בתחום מסוים נוטים להעריך את עצם מעבר למציאות.

Gradient Boosting

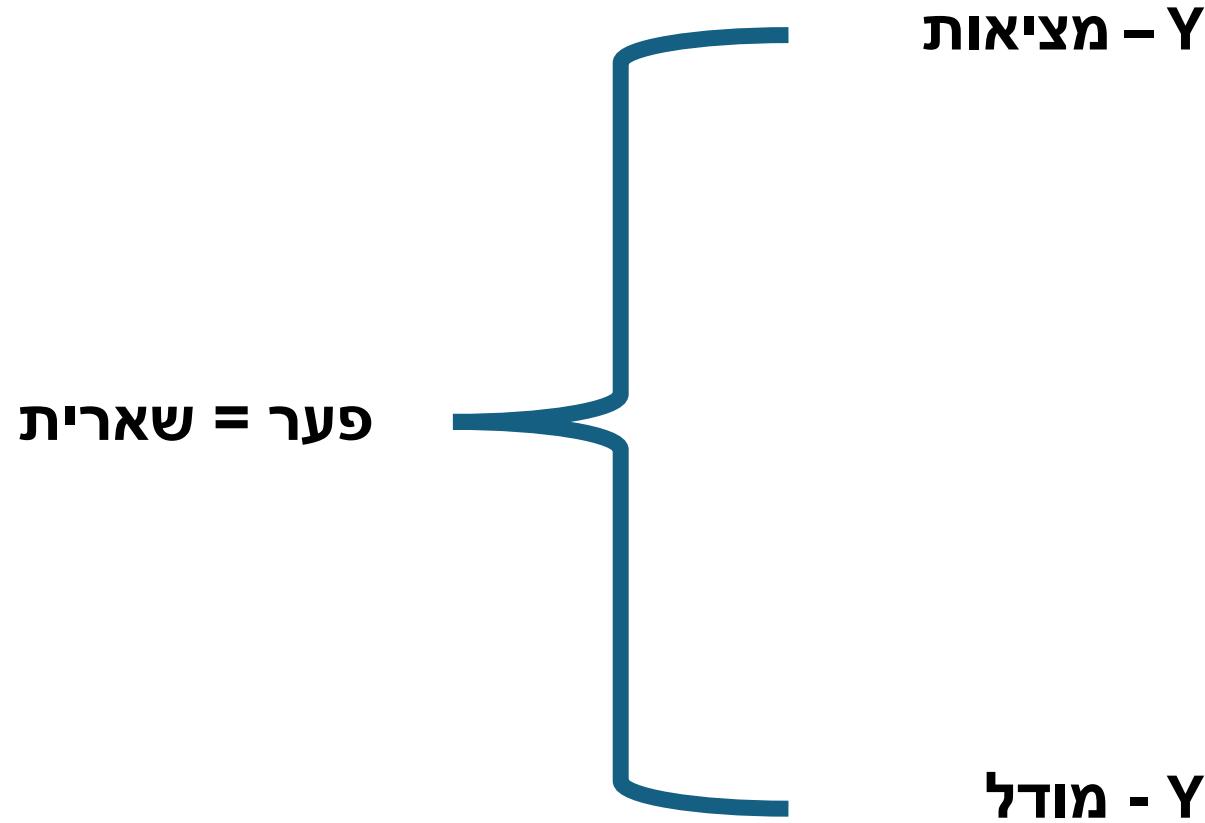
Sample mock dataset*:



*Based on the Dunning-Kruger effect: people with limited competence in a particular domain overestimate their abilities.



Gradient Boosting

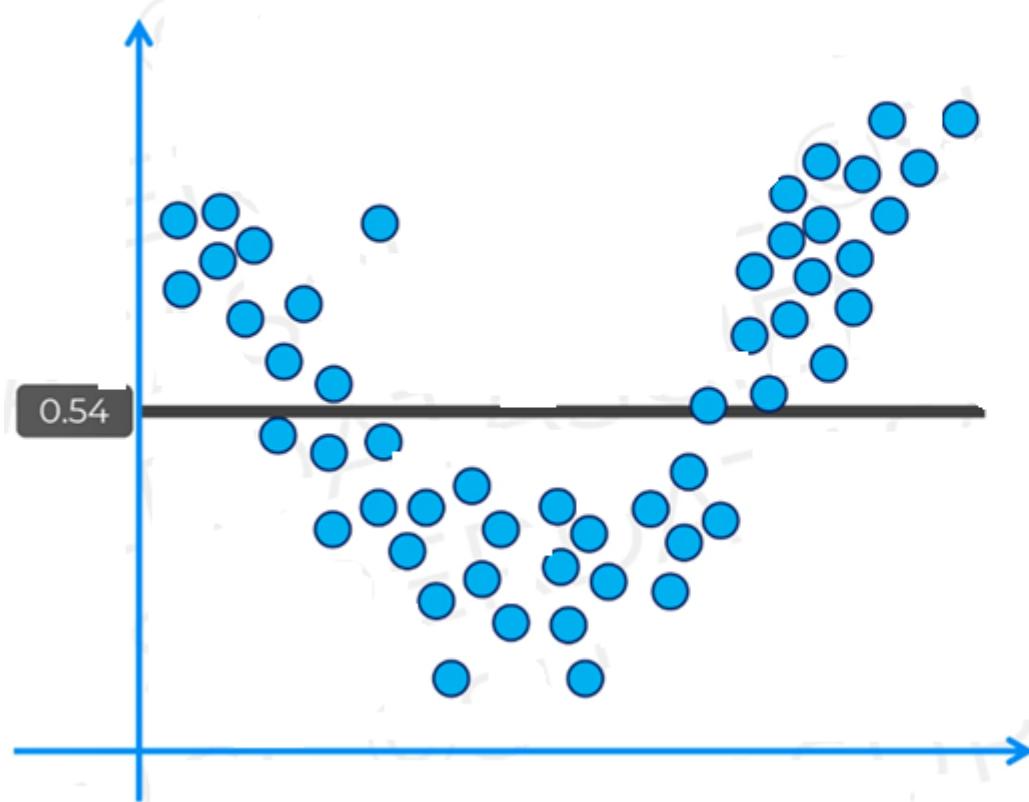


הרענון – בניית מודל על השאריות וכל פעם נשפר את המודל על ידי הוספת השאריות החזויות
(נעשה זאת בצורה איטרטיבית)

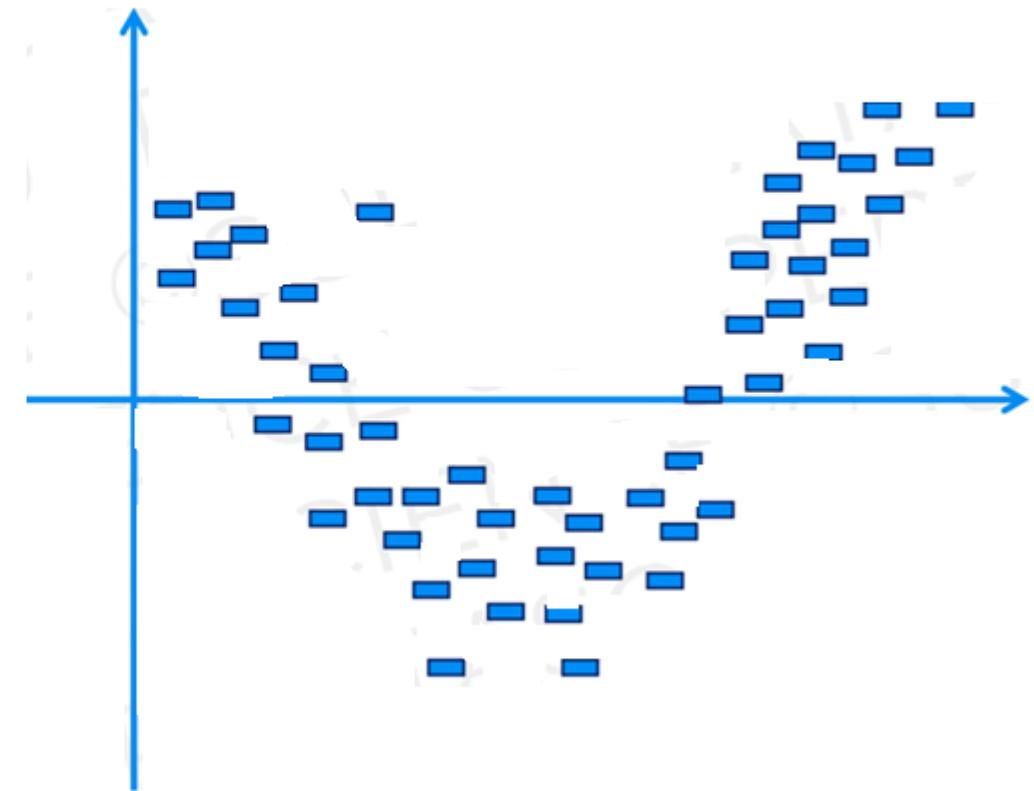


Gradient Boosting - xgboost

Original Data:



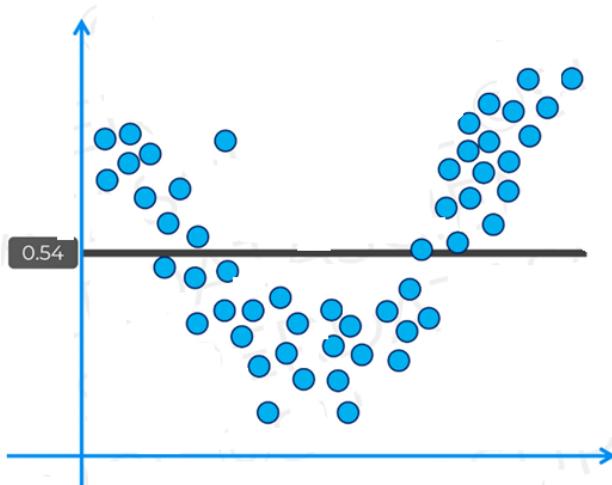
Residuals (Level 1):



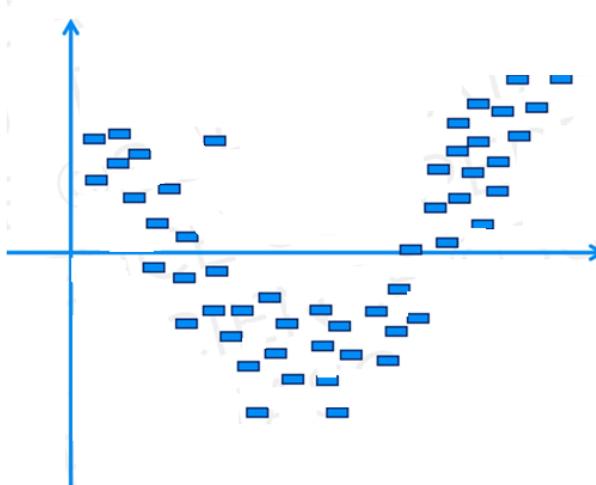


Gradient Boosting - xgboost

Original Data:



Residuals (Level 1):



■ Step 1: Start with the Original Data

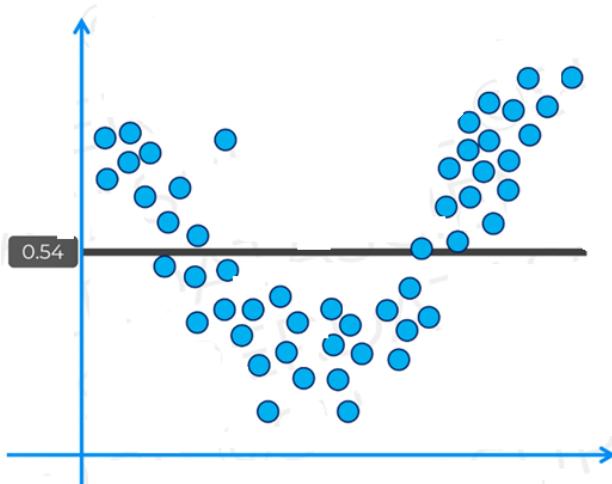
You have a dataset with features (x-axis) and target values (y-axis).

In the image, the data forms a **non-linear U-shape** pattern.

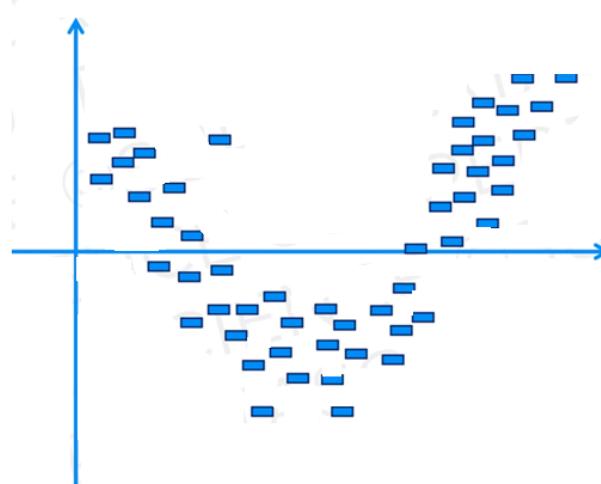


Gradient Boosting - xgboost

Original Data:



Residuals (Level 1):



■ Step 2: Fit the First Model (Model 1)

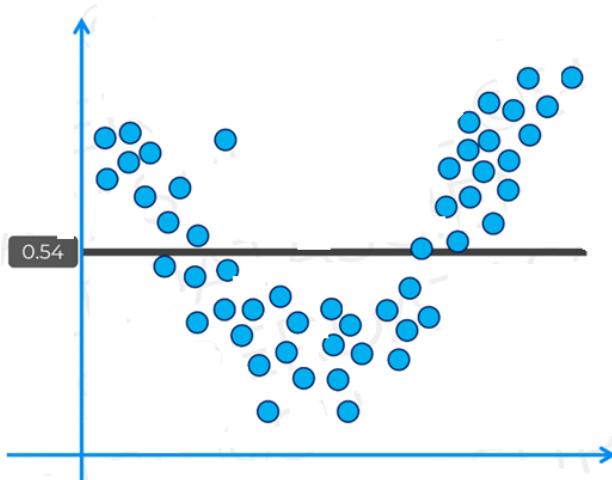
You train a **very simple model**, like a small decision tree or even a constant predictor.

- In the image, Model 1 predicts a **constant value** for all inputs:
→ Prediction = 0.54 (the average of the target values).
- This model clearly doesn't capture the U-shape at all.

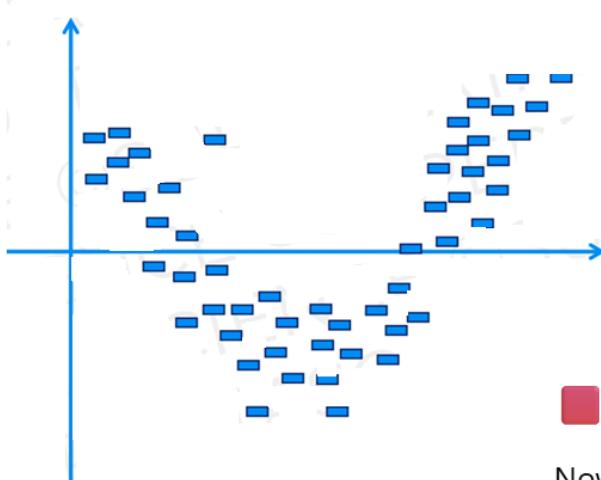


Gradient Boosting - xgboost

Original Data:



Residuals (Level 1):



■ Step 3: Compute the Residuals (Level 1)

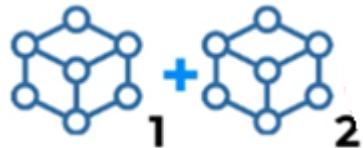
Now calculate the residuals (i.e., the errors):

$$\text{Residual} = y_{\text{true}} - \hat{y}_{\text{Model 1}} = y - 0.54$$

In the plot:

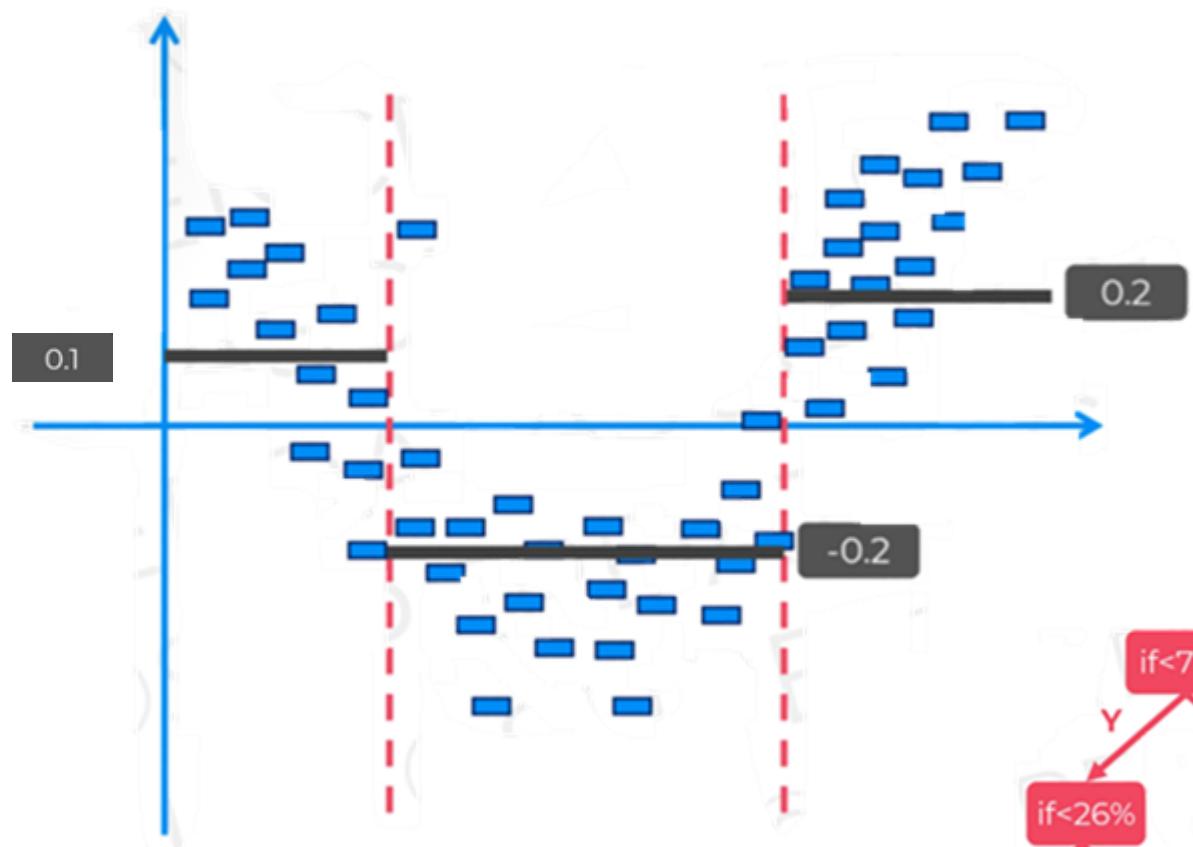
- On the **left side**, residuals are **positive** → model under-predicted.
- In the **middle**, residuals are **negative** → model over-predicted.
- On the **right**, residuals are **positive** again → under-predicted again.

These residuals are shown as **small blue rectangles** in the right graph.

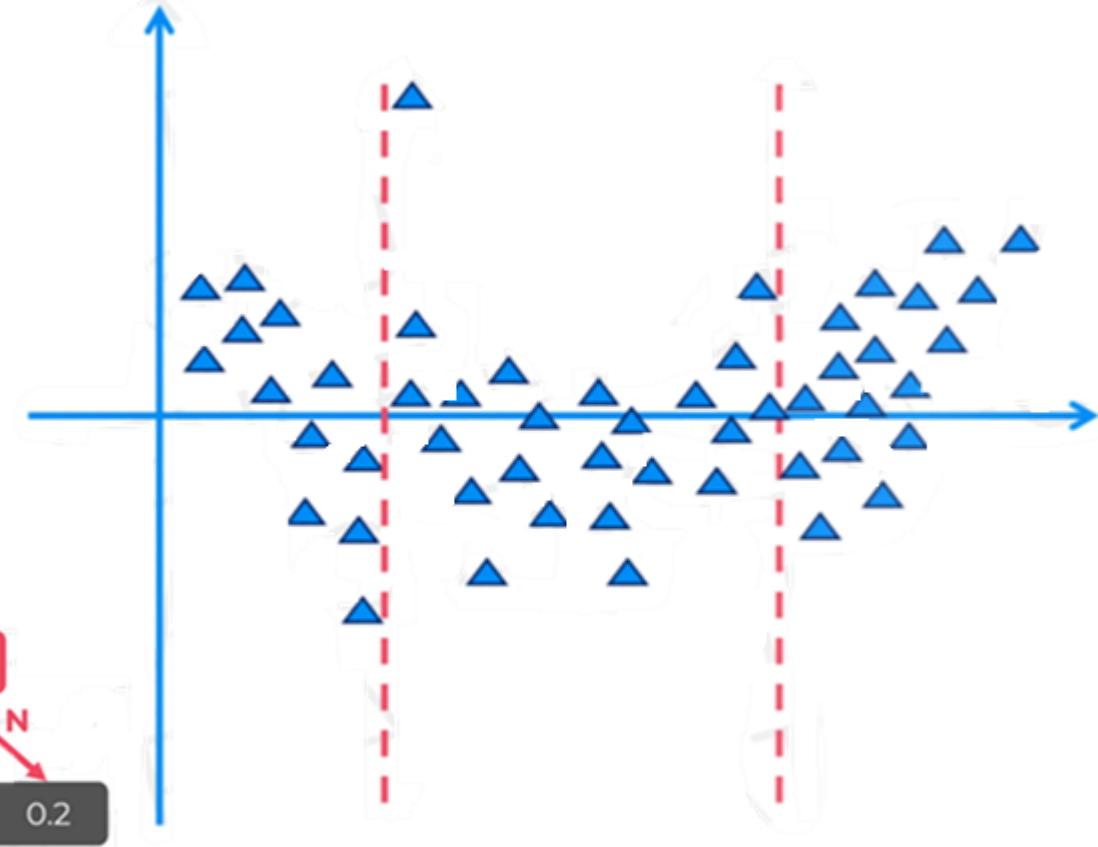


Gradient Boosting - xgboost

Residuals (Level 1):



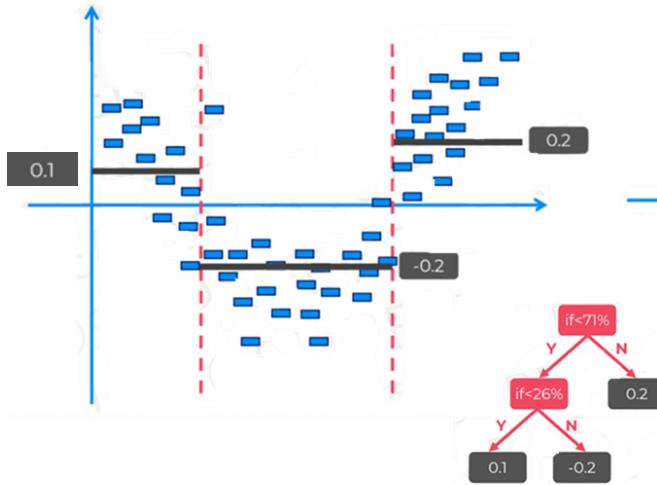
Residuals (Level 2):



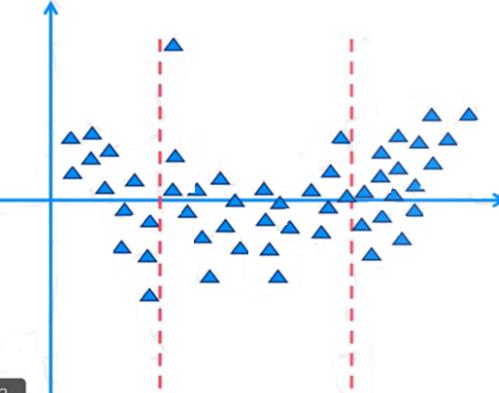


Gradient Boosting - xgboost

Residuals (Level 1):



Residuals (Level 2):



Step 4: Train the Second Model on the Residuals

You now train a **new model** (Model 2) to predict the residuals.

This model's job is:

👉 “How much did the previous model miss, and where?”

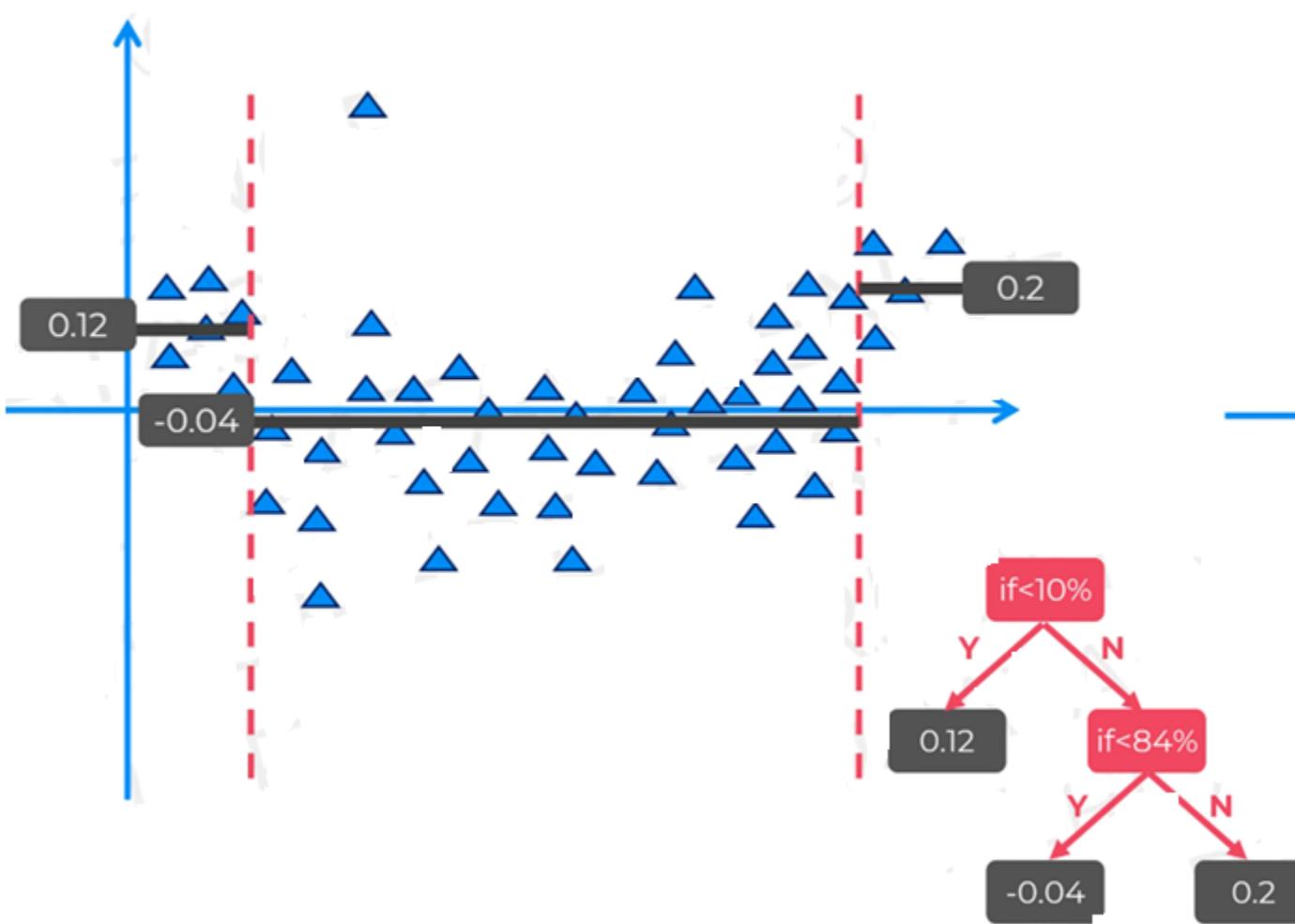
Because the residuals follow a pattern (like an upside-down U),

→ Model 2 can learn to fit that shape and make corrections.

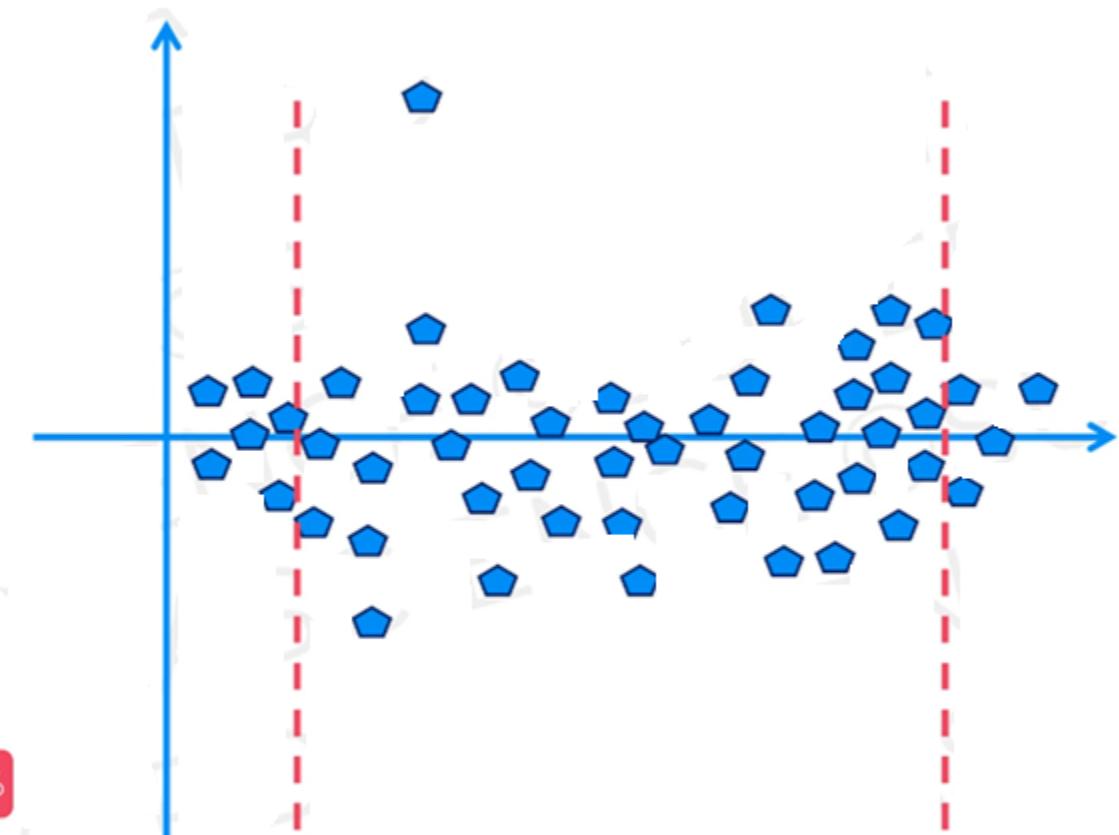


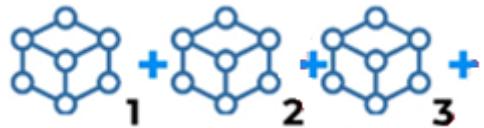
Gradient Boosting - xgboost

Residuals (Level 2):



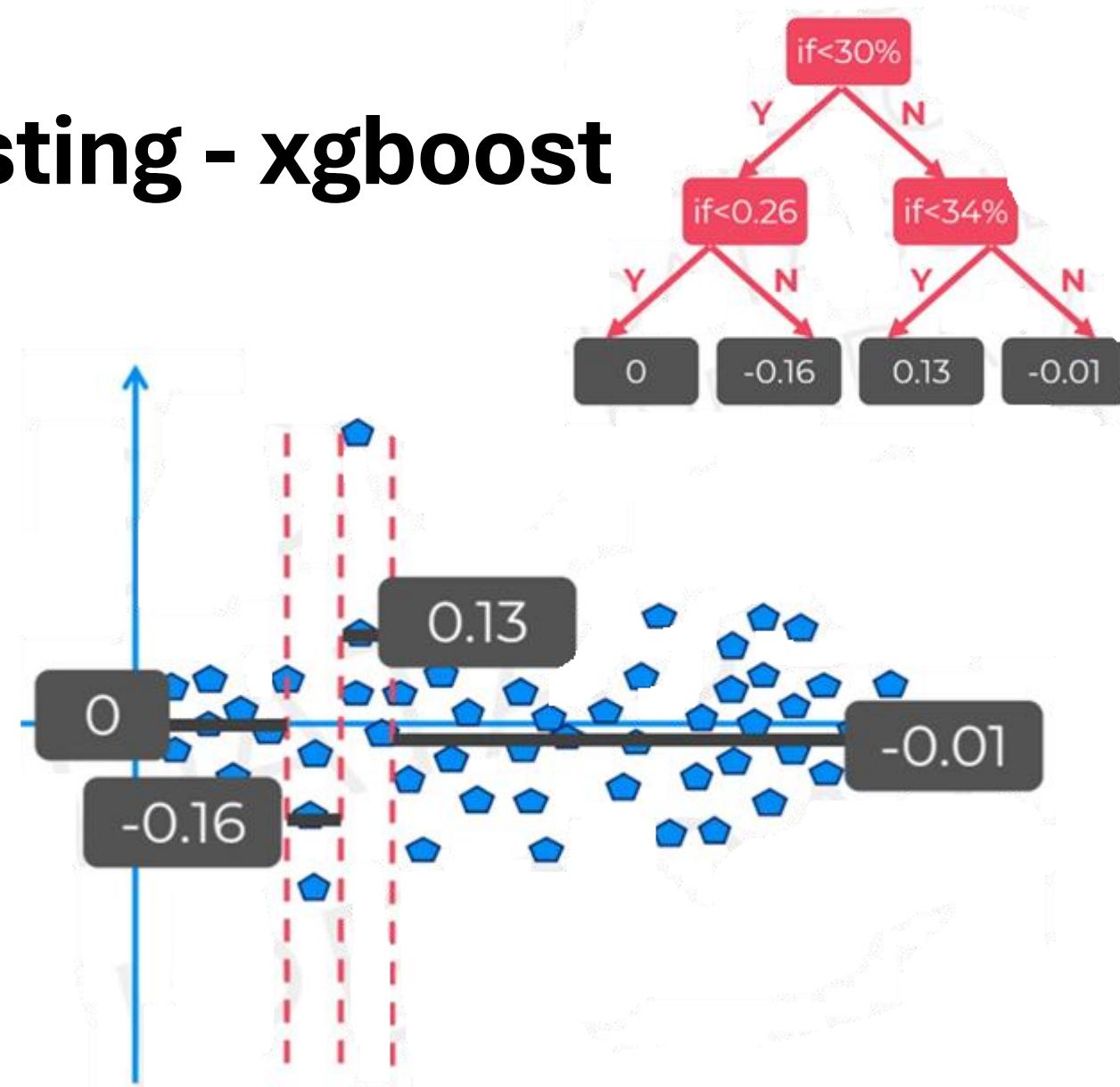
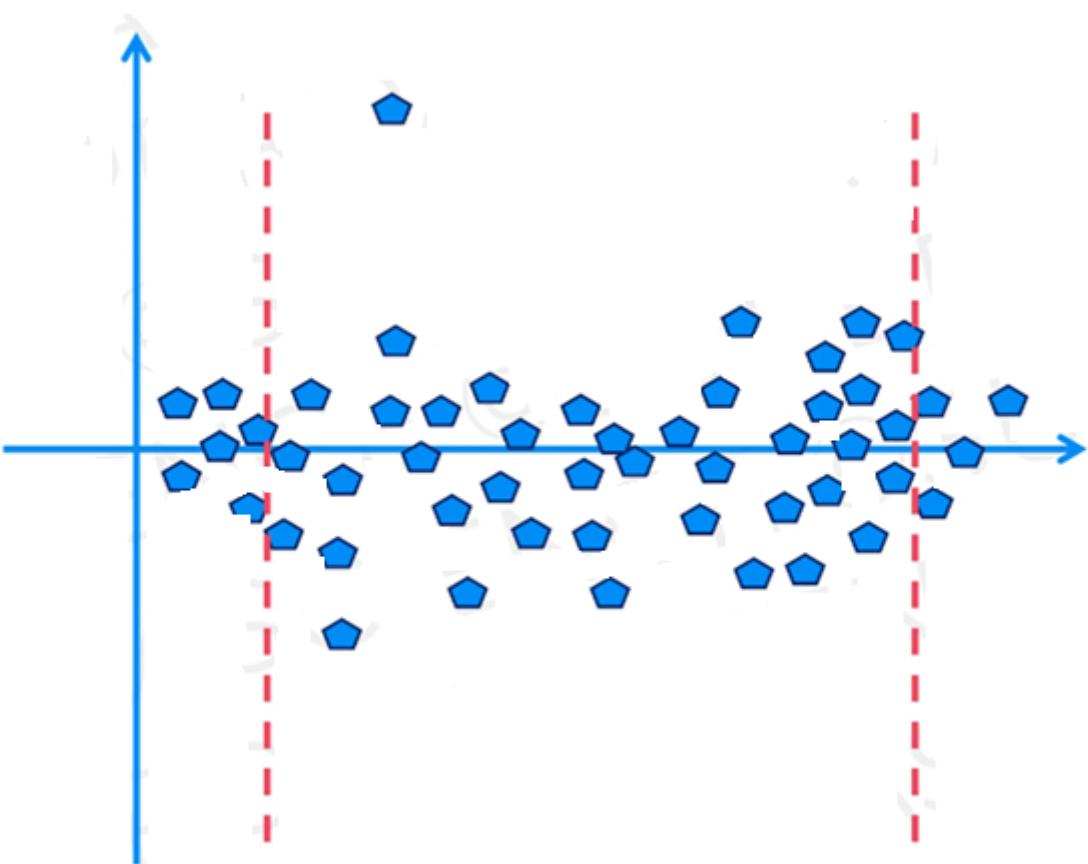
Residuals (Level 3):





Gradient Boosting - xgboost

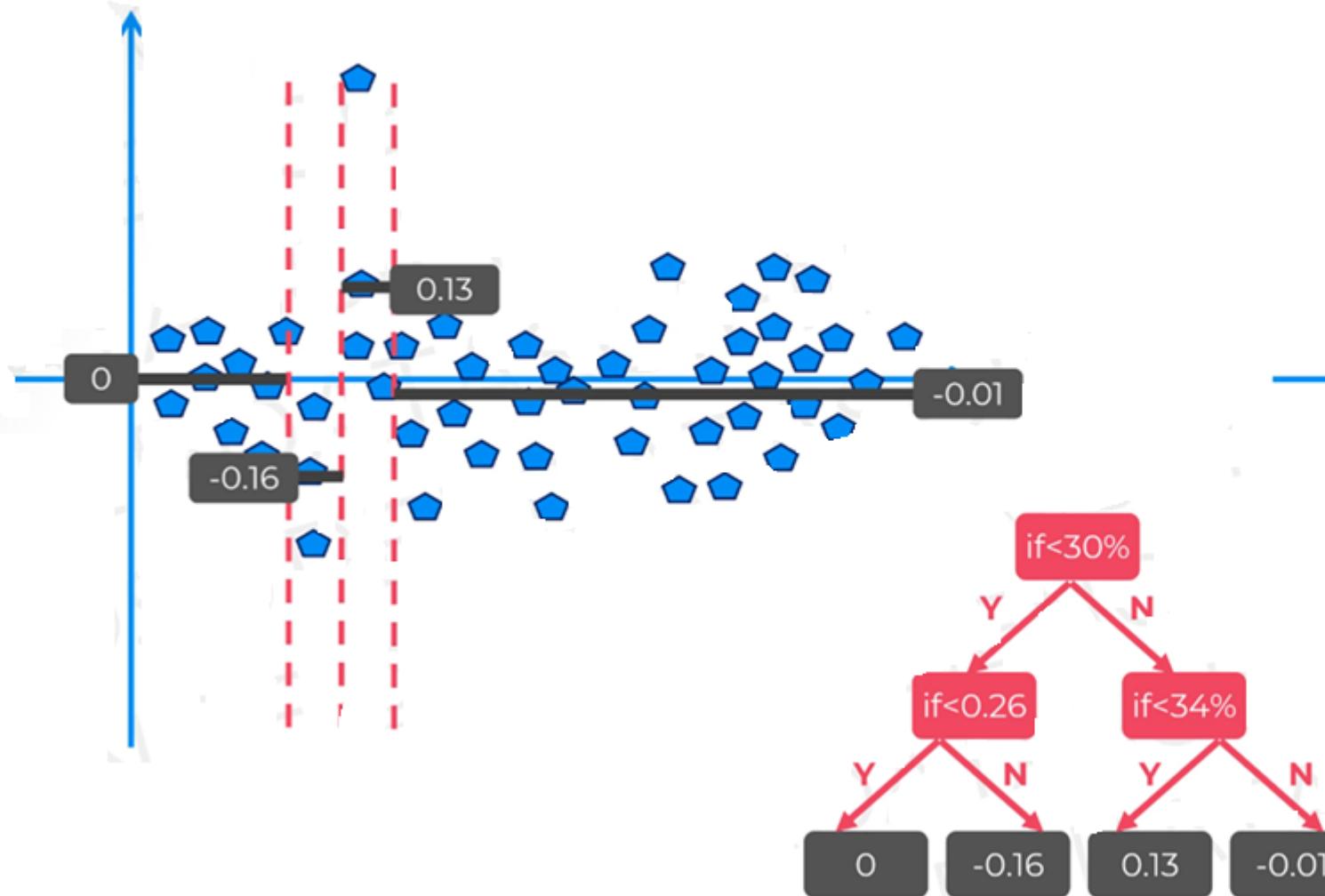
Residuals (Level 3):



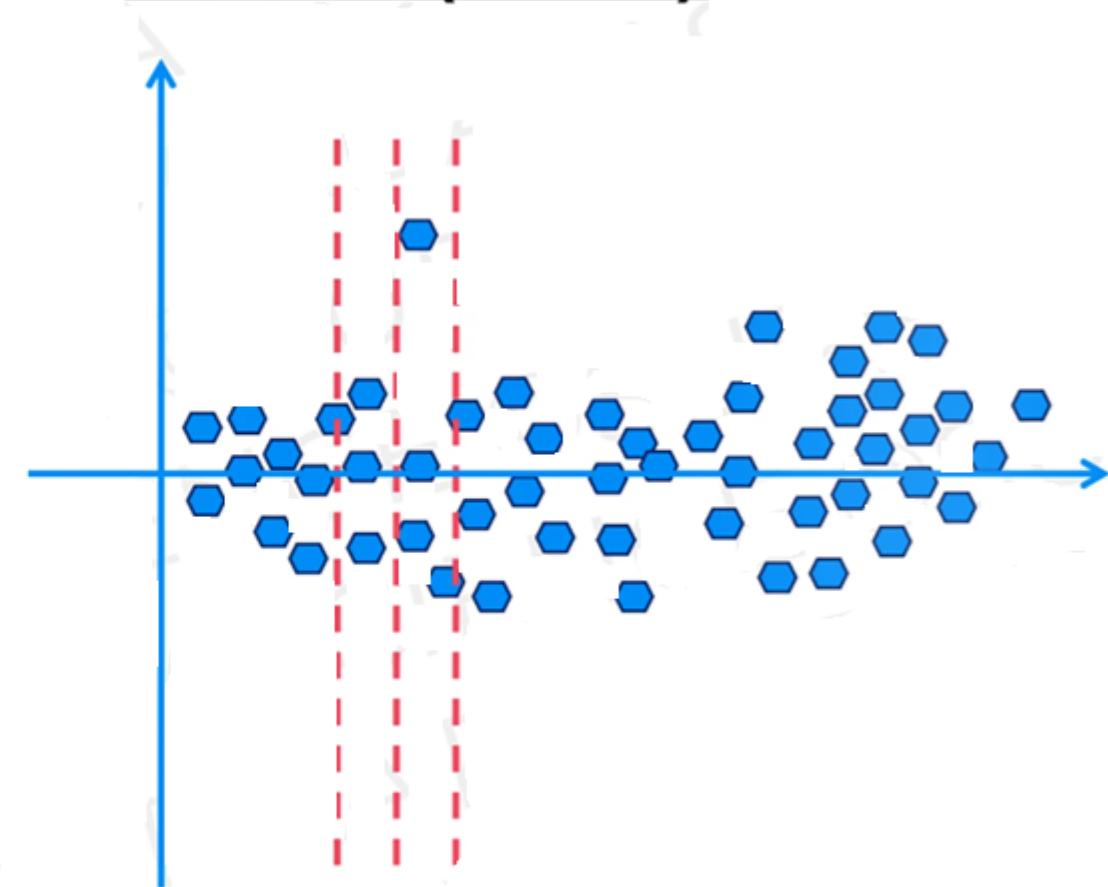


Gradient Boosting - xgboost

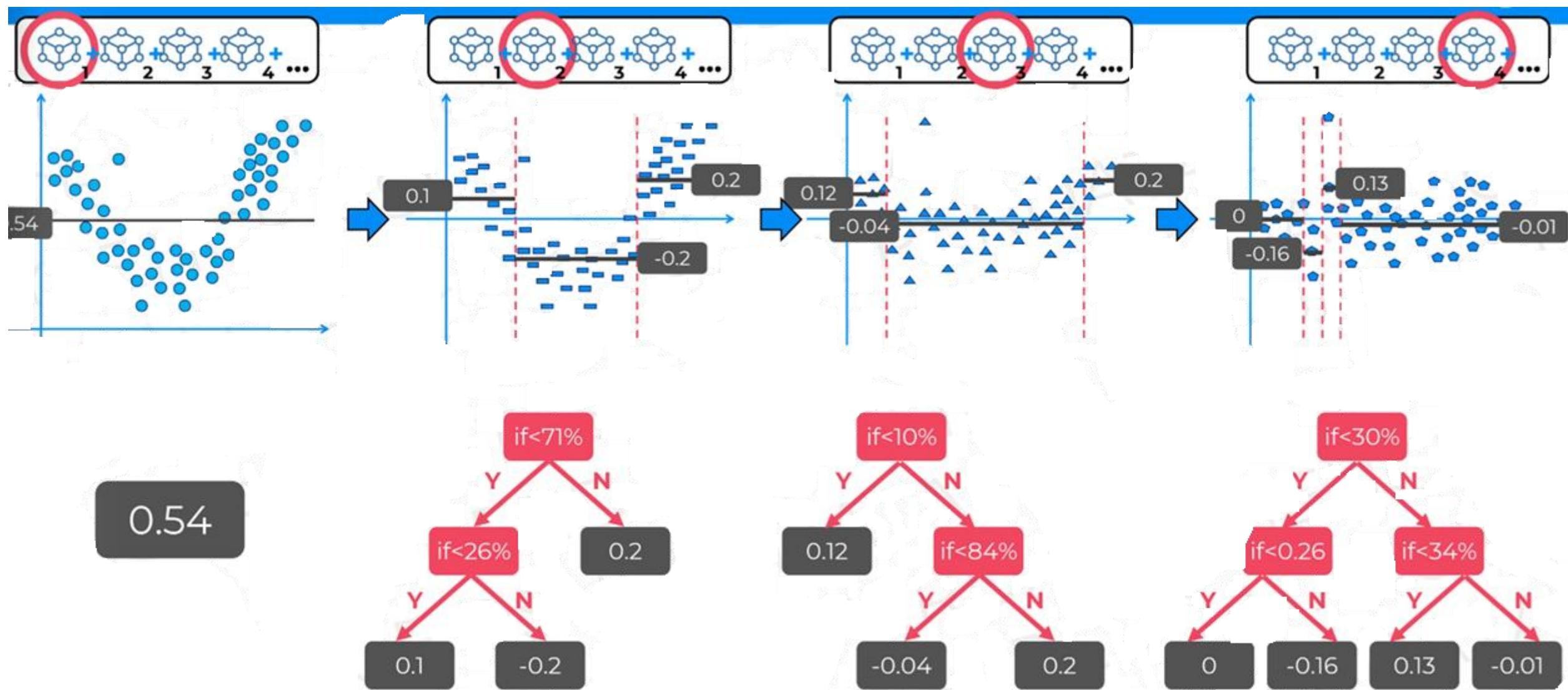
Residuals (Level 3):



Residuals (Level 4):

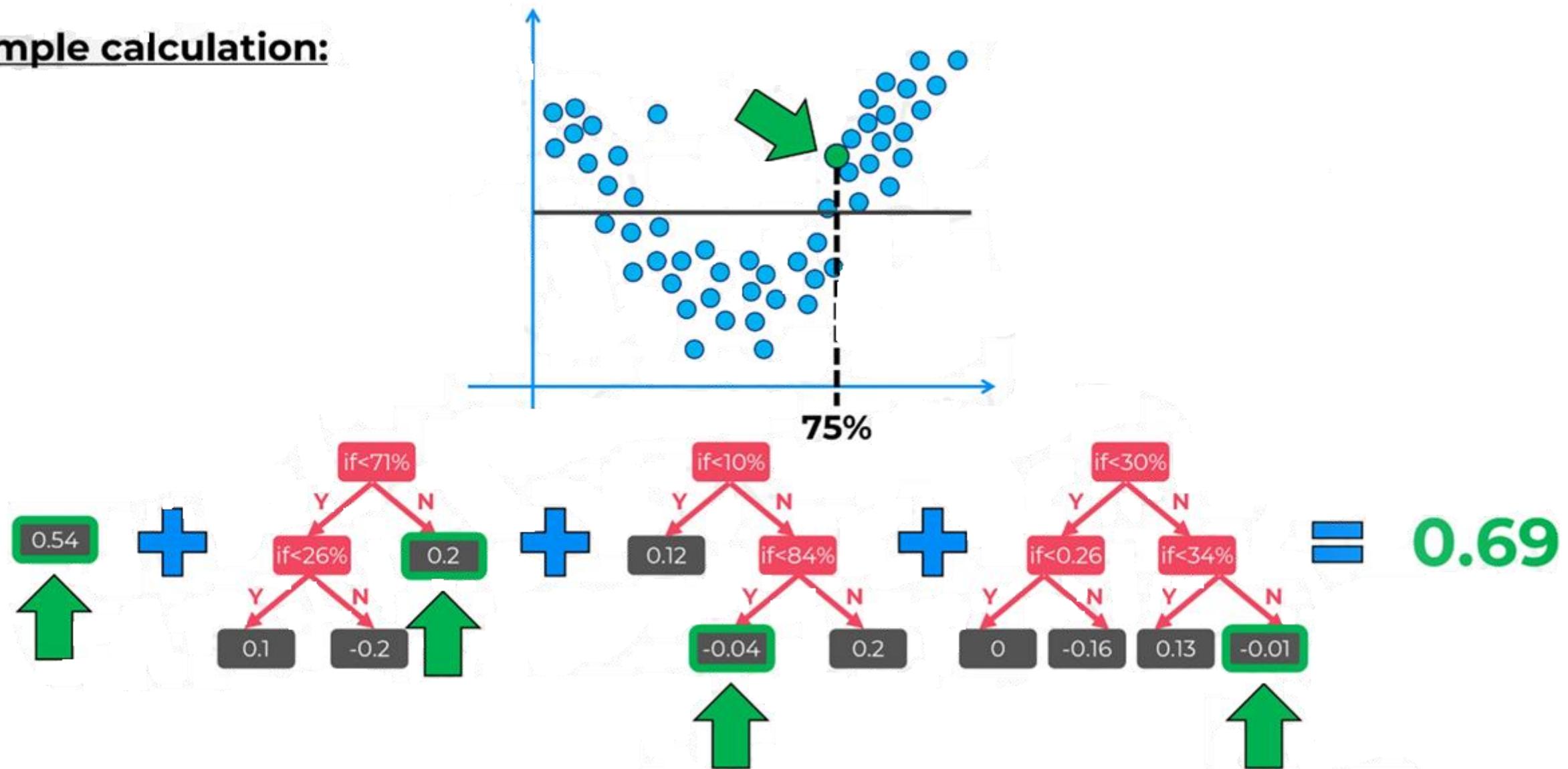


Gradient Boosting - xgboost



Gradient Boosting - xgboost

Sample calculation:



Gradient Boosting

During Training:

Row Sampling
(Subsampling)

ID	X ₁	X ₂	X ₃	X ₄	...	X _n	y
1
2
3
4
5
6
7
8
...
998
999
1,000

Gradient Boosting

During Training:

ID	x_1	x_2	x_3	x_4	...	x_n	y
1
2
3
4
5
6
7
8
...
998
999
1,000

Row Sampling
(Subsampling)

- בכל שלב של Gradient Boosting (למשל, כל פעם שאנחנו בונים עץ חדש), האלגוריתם לא משתמש בכל הדadata, אלא רק **בחלק** ממנו — לדוגמה, רק 70% מהשורות נבחרות באקראי.
- זה דומה לבאגינג (bagging), אבל שונה:
- ב-Bagging עושים דוגמה עם החזרה.
- ב-Gradient Boosting זו דוגמה **בלי** החזרה.

למה עושים את זה? 

הסביר

מטרה

עוזר למנוע ממודול "לזכור" את כל הדadata בצורה מדויקת מדי.

Overfitting 

עד קטן שמתאים רק על חלק מהנתונים רצ' מהר יותר.

לשפר ביצועים 

כל עד לנוד קצת זווית אחרת של השאריות.

לשפר גיאון בין המודלים 



Gradient Boosting - xgboost

🎯 Why Use Row & Column Sampling in XGBoost?

✓ 1. Prevents Overfitting

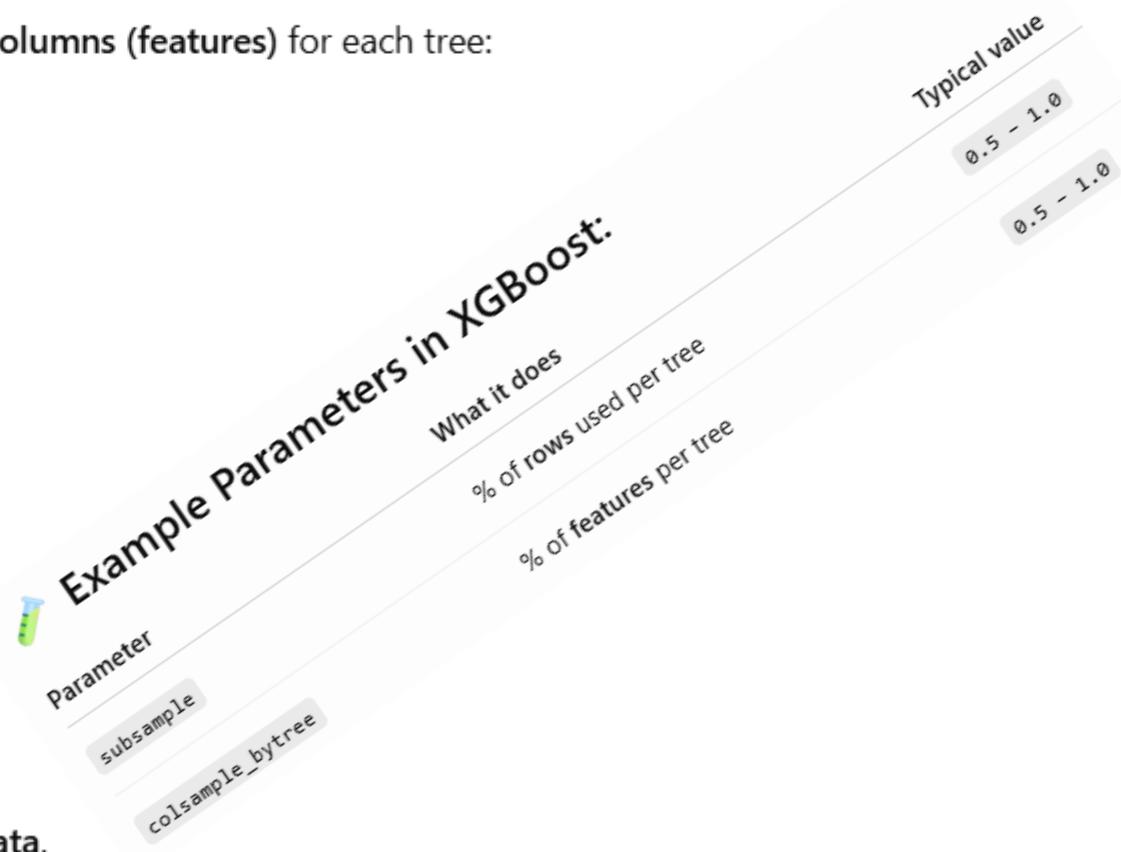
- By randomly selecting **only some rows (observations)** and **columns (features)** for each tree:
 - The model doesn't memorize the data.
 - Forces diversity across trees → better generalization.

✓ 2. Speeds Up Training

- Smaller data subset → each tree trains faster.
- Especially useful with **large datasets** or many features.

✓ 3. Creates Tree Diversity

- Similar to Random Forest's idea:
 - Less correlation between trees → stronger ensemble.
- Trees make decisions based on **different combinations of data**.



Gradient Boosting - xgboost

האם צריך לנורמל או לקודד?

1. נורמל / Scaling — לא צריך

- xGBoost לא רגש לסקאלה של המאפיינים, כי הוא מוכנס על עצים (Trees), ולא על חישובי מרחק.

- ככלומר: לא משנה אם ערך של מאפיינה הוא בין 0–1 או בין 0–10,000 — העץ רק בודק תנאים כמו `feature < .threshold`.

לכן: אין צורך לעשות MinMax או StandardScaler לפני שימוש ב-xGBoost. 

Gradient Boosting - xgboost

האם צריך לנרמל או לקודד?

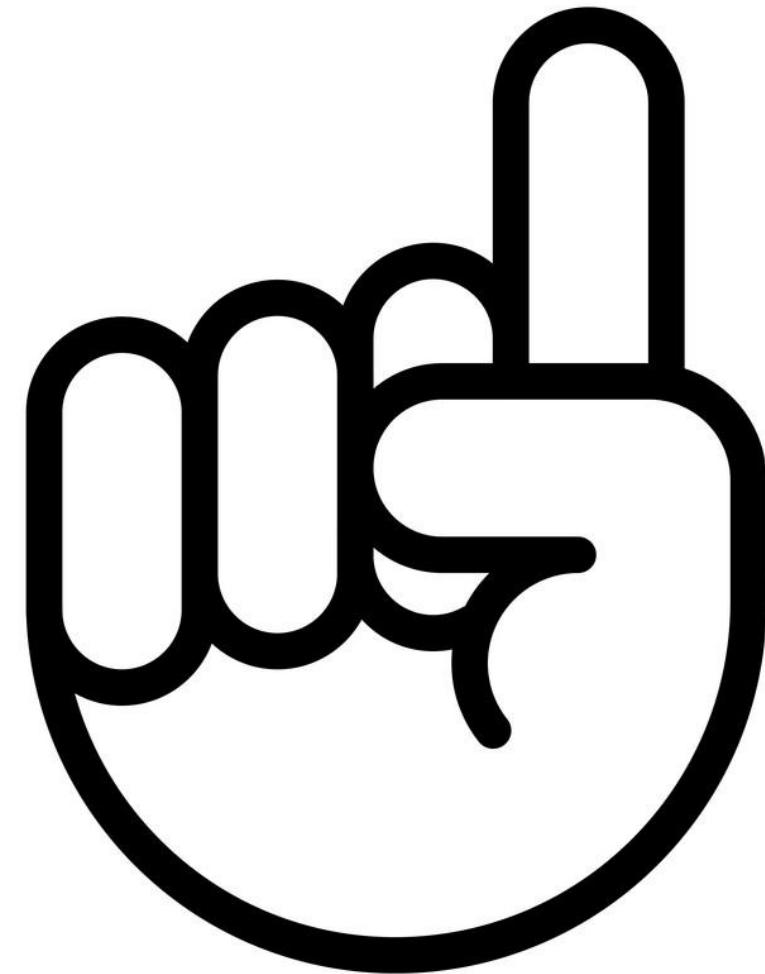
2. קידוד משתנים קטגוריאליים – כן צריך

- XGBoost לא תומך בתכונות קטגוריאליות באופן ישיר.
- לכן, חיבבים לקודד את העמודות הקטגוריאליות למספרים לפני ההזנה למודל.

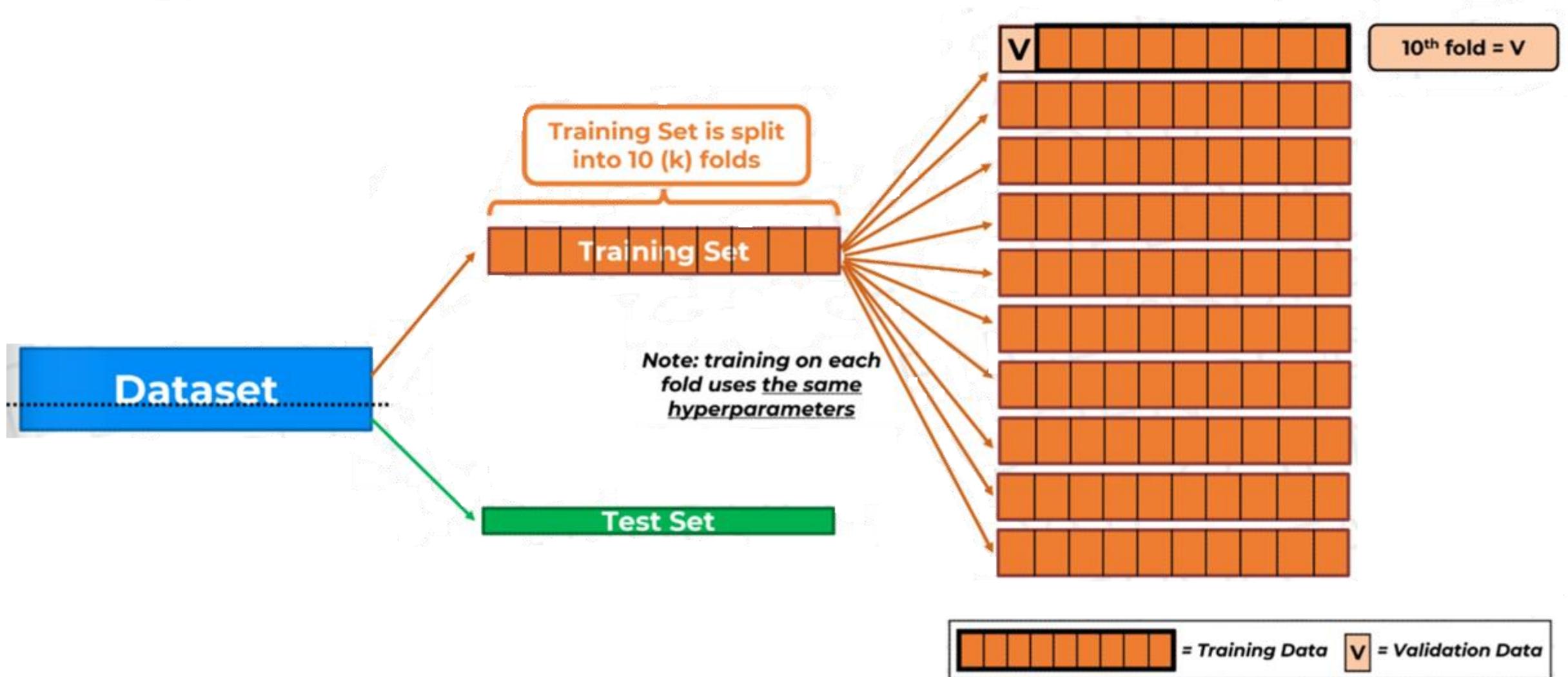
איך לקודד?

- מתאים אם יש מעט קטגוריות:
`.color_red, color_blue` הפקיד ל-2 עמודות: `color = ['red', 'blue']`
- לדוגמה: `[color_red, color_blue]` – מתאים לקטגוריות רכבות, אבל בזהירות:
`Label Encoding`
- XGBoost עלול לפרש את המספרים כמשמעויות (כלומר, ש² > 1), למרות שאין בכך היגיון.

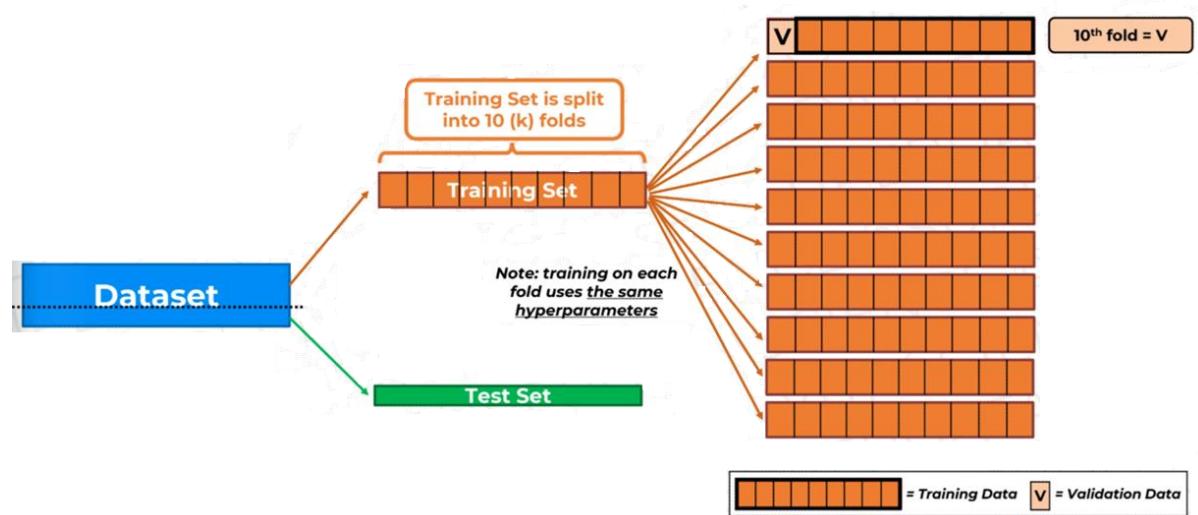
Xgboost Regressor



K-Fold Cross Validation



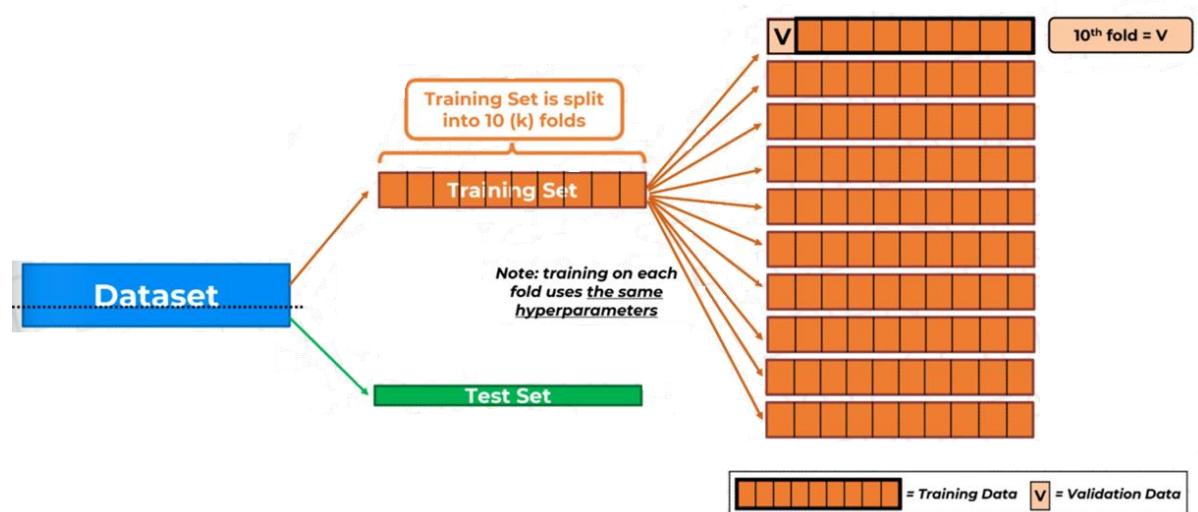
K-Fold Cross Validation



Dataset .1

- הנתונים מחולקים לשני חלקים:
- עברו אימון והצלבת תוקף (Cross Validation) •
- נשמר בצד, ומשמש רק לבדיקה הסופית לאחר בחירת המודל. •

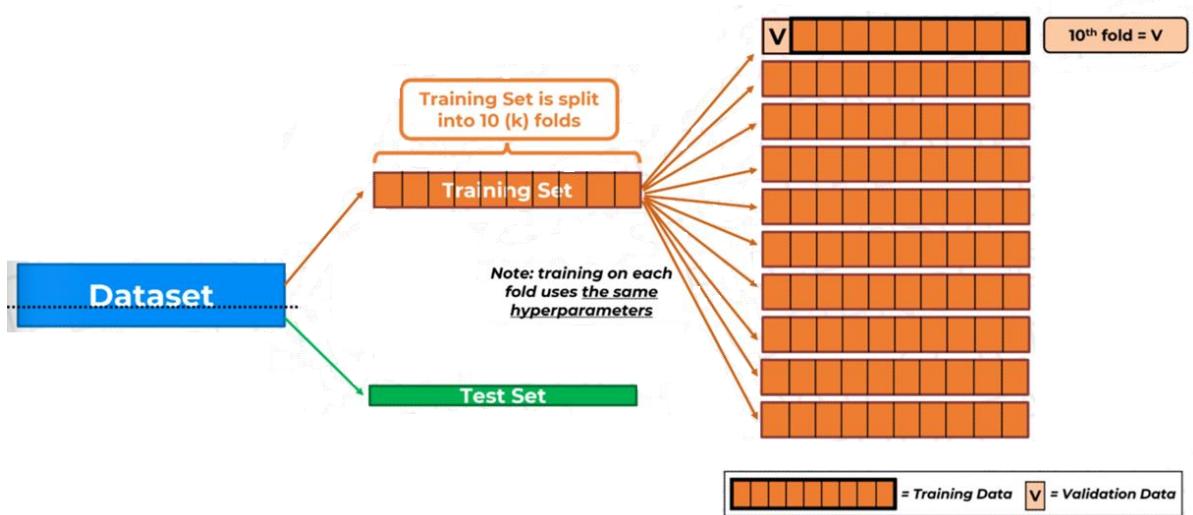
K-Fold Cross Validation



Training Set Split into K Folds ■ .2

- קבוצת האימון (Training Set) מחולקת ל-K תתי-קבוצות שוות בגודל, בתרמונה: (10-Fold) . $K = 10$
- כל אחת מהתתי-קבוצות תיקרא **fold**.

K-Fold Cross Validation

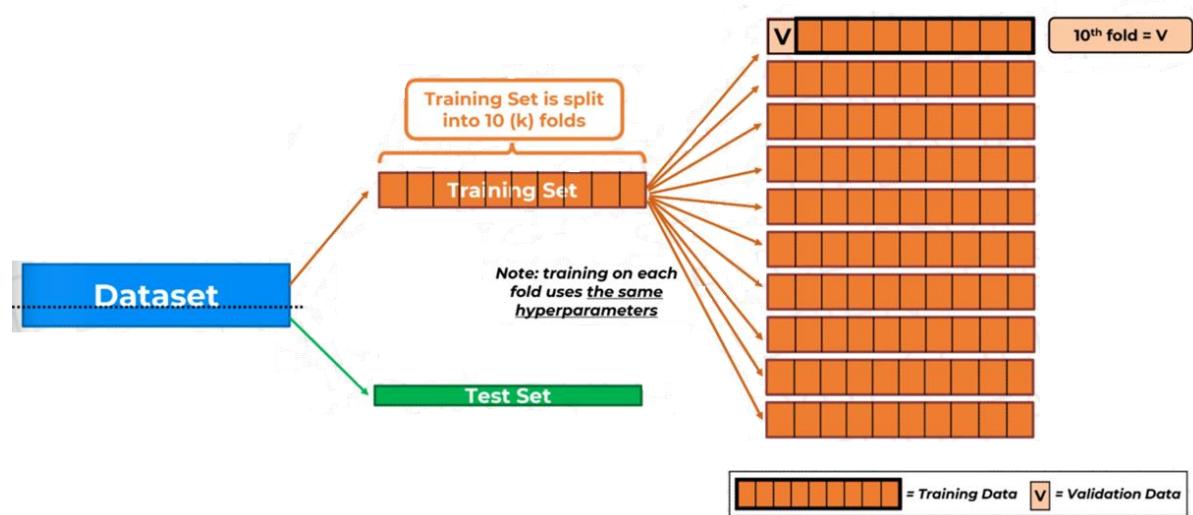


3. תהליכי הצלבה (Cross Validation Loops)

- נכצע K איטרציות של אימון:
- בכל איטרציה:
 - נבחר Fold אחר להיות קבוצת האימונים (Validation Set).
 - כל שאר folds-1-K משמשים כאימון (Training Set).
- התוצאה: נבנו K מודלים, וכל אחד מהם נבחן על חלק שונה מהנתונים.

לדוגמא, באיטרציה ה-3:
3 Fold משמש כ-Validation, ו-1,2,4,...,10 Folds משמשים לאימון.

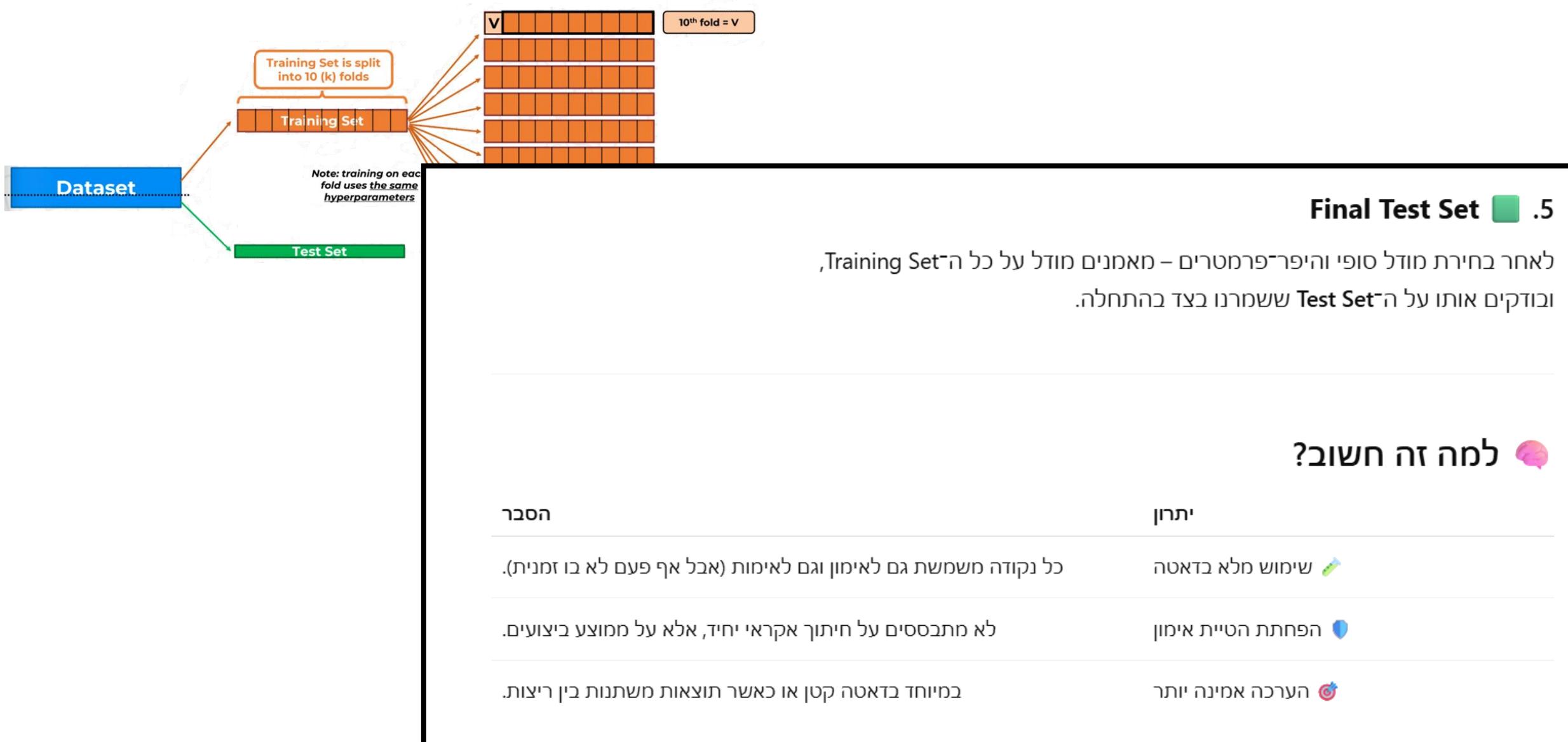
K-Fold Cross Validation



4. **סקולול התוצאות**

- בסוף, מחשבים ממוצע של ביצועי המודלים (לדוגמה: דיק, RMSE, AUC וכו') על כל h -folds.
- זה נותן הערכה **יציבה** ואובייקטיבית לביצועי המודל על הדאטה.

K-Fold Cross Validation



Gradient Boosting - xgboost

Feature Importance

XGBoost Feature Importance 💬

- XGBoost מאפשר-Calculating Feature Importances. באמצעות טכנייה מתקדמת שנקראת "gradient boosting".
- חשיבותו של מושג זה הוא ביכולתו לסייע בהחלטות הפלט של המודל.
- שימושי ל-:
- הסבר המודל (Explainability)
- צמצום משתנים (Feature Selection)
- שיפור ביצועים והבנת השפעות עסקיות

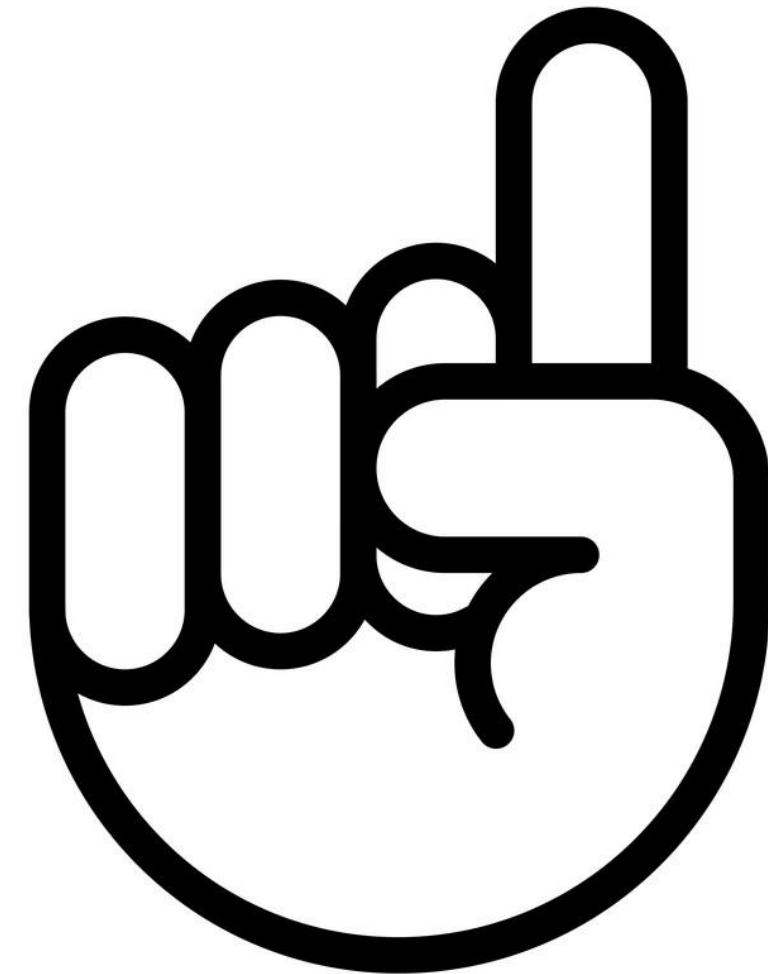
השוואת ביצועים:

לאחר בניית שני המודלים – עם כל הפיצרים ועם רק הפיצרים החשובים – נבדוק את ביצועי המודל לפי מדדים סטנדרטיים:

- **RMSE** – שורש ממוצע השגיאה הריבועית
- **MAE** – ממוצע השגיאה האבסולוטית
- **R²** – אחוז השונות שהמודל מסביר

אם הביצועים כמעט זהים – נעדיף את המודל המצוומצם: פשוט יותר, מהיר יותר, וקל לפרשנות.

Xgboost Regressor



xgboost_classifier

