

Boosting Algorithm

Regressor

What is Boosting?

Boosting is an ensemble technique that combines multiple weak learners (models that perform slightly better than random guessing) to build a strong learner.

In regression, boosting builds models sequentially, where each new model tries to correct the errors of the previous models.

How does Boosting work in regression?

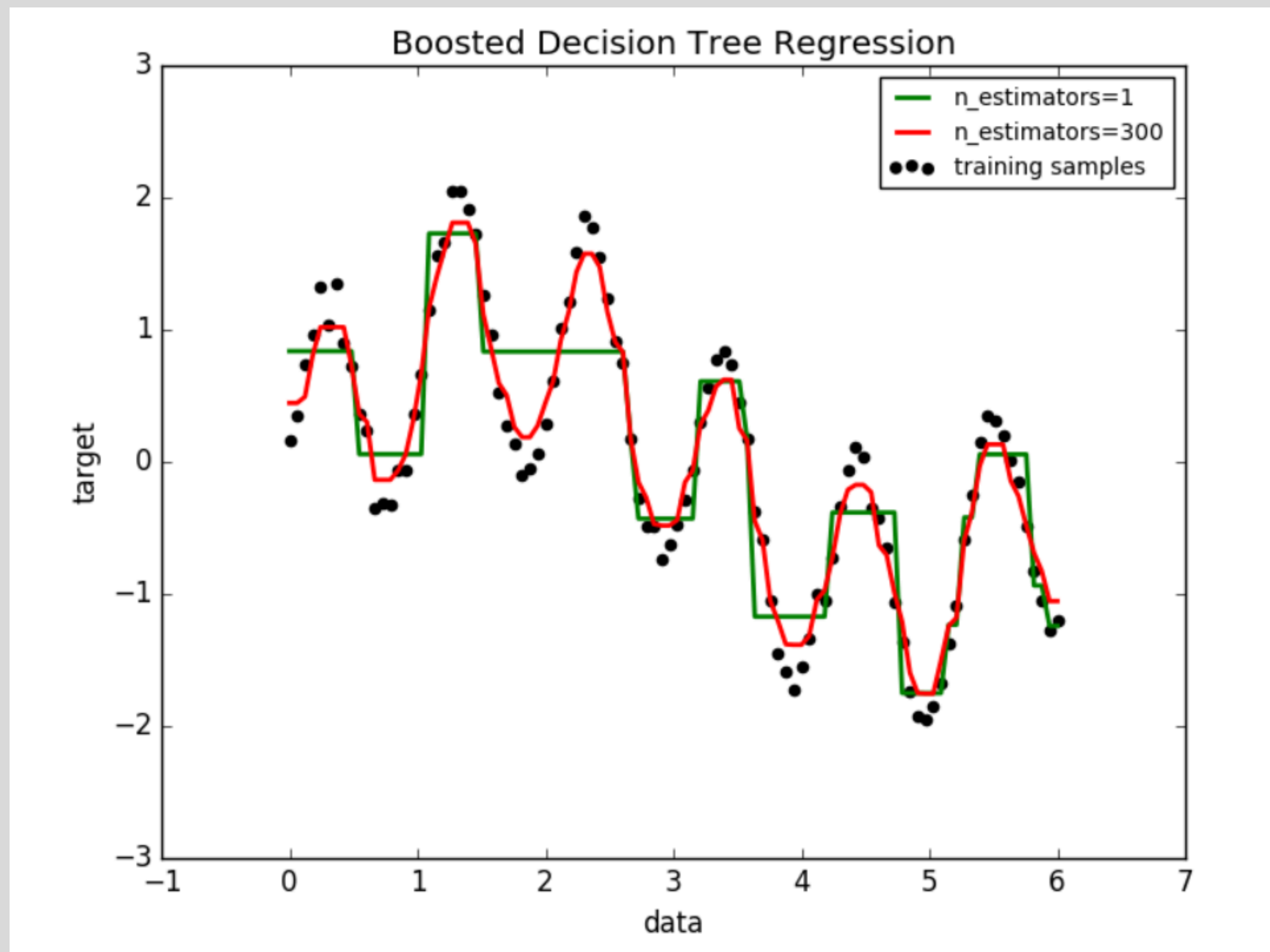
- Start with a base model (usually a weak learner) to predict the target.
- Calculate the residuals (errors) between predicted and actual values.
- Train the next model on these residuals to reduce the errors.
- Combine all models' outputs to produce the final strong prediction.

Types of Boosting Algorithm

1. AdaBoost Regressor (Adaptive Boosting)
2. Gradient Boosting Regressor
3. XGBoost (Extreme Gradient Boosting)
4. LightGBM (Light Gradient Boosting Machine)
5. CatBoost (Categorical Boosting)

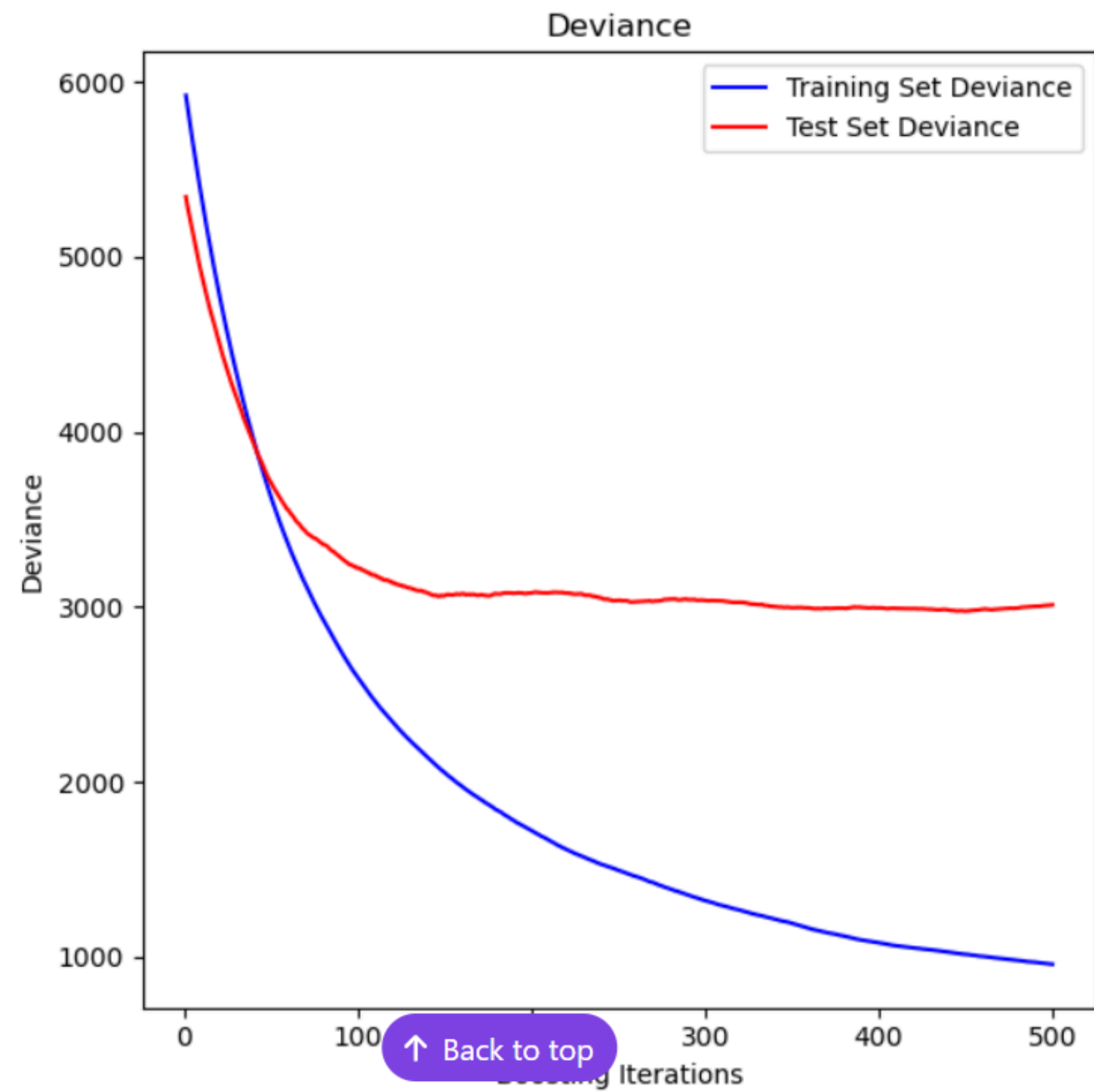
1. AdaBoost Regressor (Adaptive Boosting)

- Builds models sequentially.
- Assigns weights to training samples. Misclassified samples get higher weights for the next iteration.
- Final prediction is a weighted sum of all weak learners.
- Sensitive to outliers as they get higher weights in subsequent models.



2. Gradient Boosting Regressor

- Uses gradient descent to minimize the loss function.
- Each new model is trained to predict the residual errors (negative gradients) of the previous model.
- Final prediction = sum of all learners' outputs.
- More robust to overfitting if parameters are tuned well (e.g. learning rate, number of estimators).



3. XGBoost (Extreme Gradient Boosting)


- An optimized implementation of gradient boosting.
- Includes regularization (L1 & L2) to avoid overfitting.
- Efficient for large datasets due to parallel processing, tree pruning, and sparse-aware algorithms.
- Widely used in competitions (e.g. Kaggle).

4. LightGBM (Light Gradient Boosting Machine)

- Uses histogram-based learning for faster training.
- Grows trees leaf-wise (best leaf) instead of level-wise, leading to better accuracy but risk of overfitting on small datasets.
- Handles large data with high speed and low memory usage.

5. CatBoost (Categorical Boosting)

- Developed by Yandex for handling categorical variables natively without explicit encoding.
- Robust to overfitting, efficient, and easy to tune.
- Works well with less preprocessing compared to XGBoost and LightGBM.

Algorithm	Key Feature	Strength	Limitation	
AdaBoost	Weighted sample adjustment	Simple, improves weak learners	Sensitive to outliers	
Gradient Boosting	Minimizes residual errors	Accurate, customizable loss functions	Slower, prone to overfitting if not tuned	
XGBoost	Optimized Gradient Boosting	Fast, regularized, scalable	Slightly complex parameter tuning	
LightGBM	Histogram-based, leaf-wise growth	Very fast, handles large data	Overfits on small data	
CatBoost	Native categorical support	Easy to use, less preprocessing	Slightly slower than LightGBM	