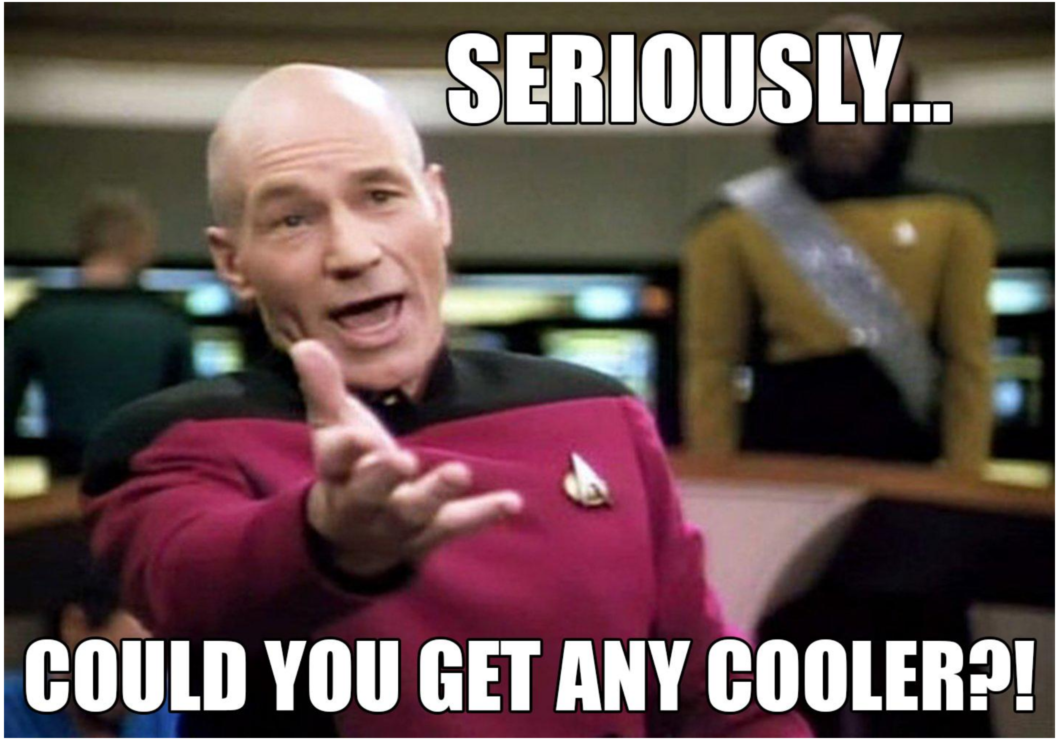
Feeling Boosted Yet?



Welcome to Week 8

Lecture 2!

Data Science in Python &
Machine Learning

Announcements

- Make sure that you have access & have viewed your feedback document.

- All assignments must be submitted by Friday at 9am PST

- All resubmits must be resubmitted by Friday at 9am PST

Today’s Agenda

- What is boosting?
- How is boosting different than other ensemble models?
- Conceptualizing Adaboost
- Codealong and breakout room challenge!

Boosting

What is boosting?

- Boosting is an ensemble method based on a simple decision tree
- It starts by making a simple decision tree and making the predictions
- But it doesn’t stop there!
- It then uses the residual errors (difference between the observed value and the predicted value) from the first tree as the target for the next tree
- Essentially, it improves based on the “mistakes” of the previous attempt

Gradient Boosting vs. other Tree models

An analogy for conceptualizing the differences:

Imagine you are hiring someone:

A simple decision tree is represented by you developing the criteria and making the choice.

Bagged trees is represented by a panel of interviewers who each have a vote in the final decision.

Random Forest is also represented by a panel, but in this case, each interviewer focuses only on a subset of the criteria.

Boosting is represented by a series of interviews in which the criteria used is altered based on feedback from the previous interviewer.

Ensemble Models

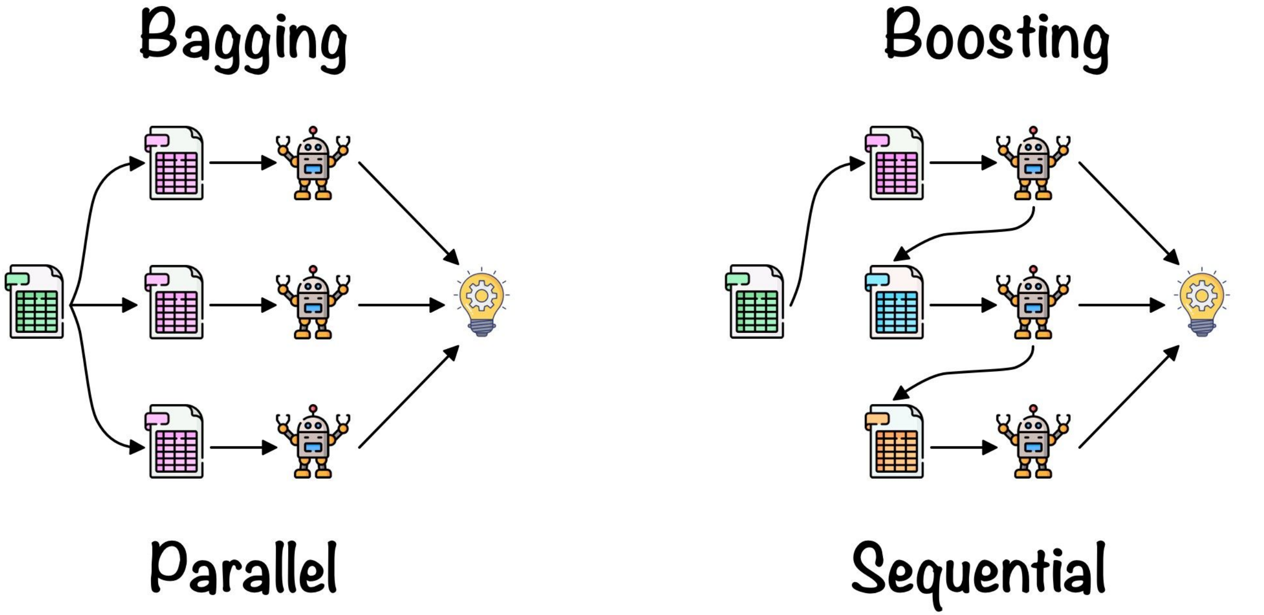


Image courtesy of Fernando Lòpez’s blog

Adaboost

1st model predicts everything with x < 1 is a circle.

2nd model focuses on 1st model's errors, the red circles. It predicts everything with y > 4 is a circle.

3rd model focuses on 2nd model's errors, the blue square. It predicts everything with x > 5 is a square.

Add the weak models together and we get good predictions.

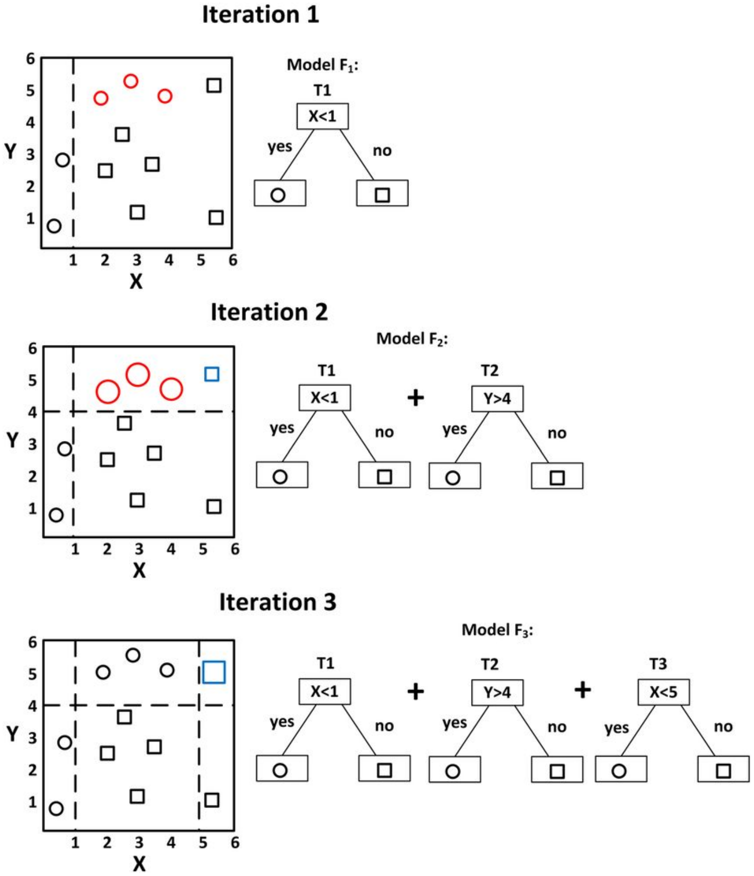


Image courtesy of Rafael del Rìo

Why Boosting?

- It works! Boosting models perform very well
- It resists overfitting

Downsides of Boosting

- Computationally complex
- Sometimes slow to train

Boosting in Python

CLASSIFICATION: Use them like any sklearn model.

- from sklearn.ensemble import AdaBoostClassifier
- from sklearn.ensemble import GradientBoostingClassifier
- From sklearn.ensemble import SGDClassifier
- from lightgbm import LGBMClassifer
- from xgboost import XGBClassifier

Also come in regression flavors!!!

Your CodeAlong Today!!!

Try 3 different boosting classifiers on this Churn dataset from Kaggle to predict whether a customer will close their account at a bank (churn).

Your target column is ‘Exited’

Today’s Colab Notebook

Is this a classification or regression problem?

Challenge Notebook

Challenge Dataset