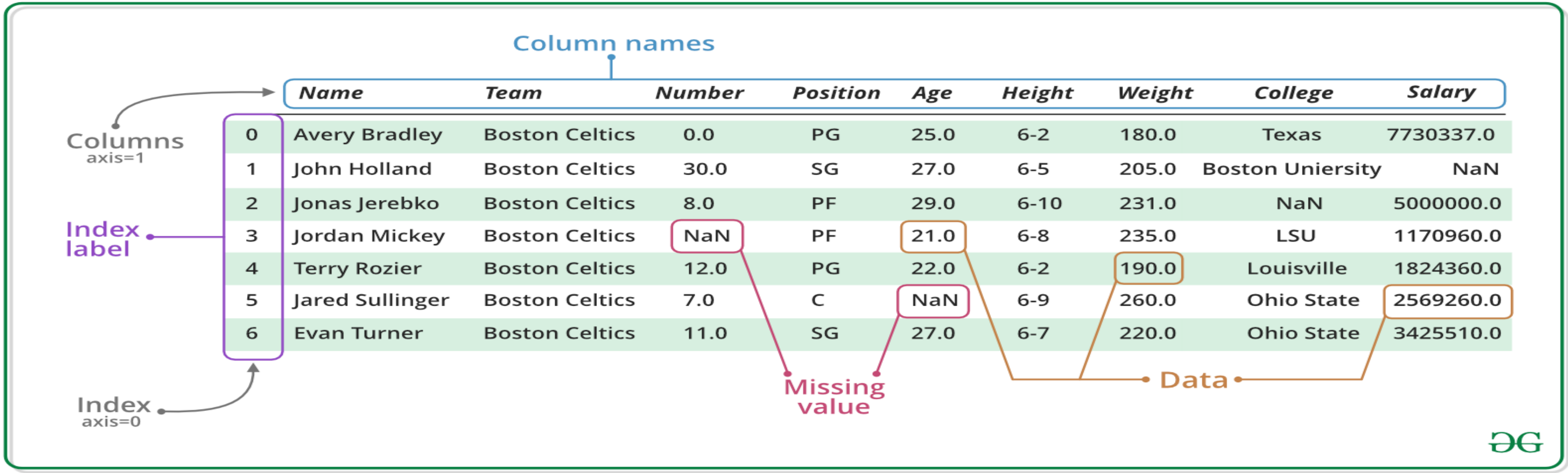
Demystifying Gradient Boosting

Joseph Eddy
Senior Data Scientist, Metis

What is Gradient Boosting?

- State of the art machine learning algorithm for **tabular prediction problems**; especially those with messy, missing data and high degrees of feature interaction and nonlinearity
- In this introduction we'll focus on **regression**, but the general algorithm can be extended to many tasks, including classification

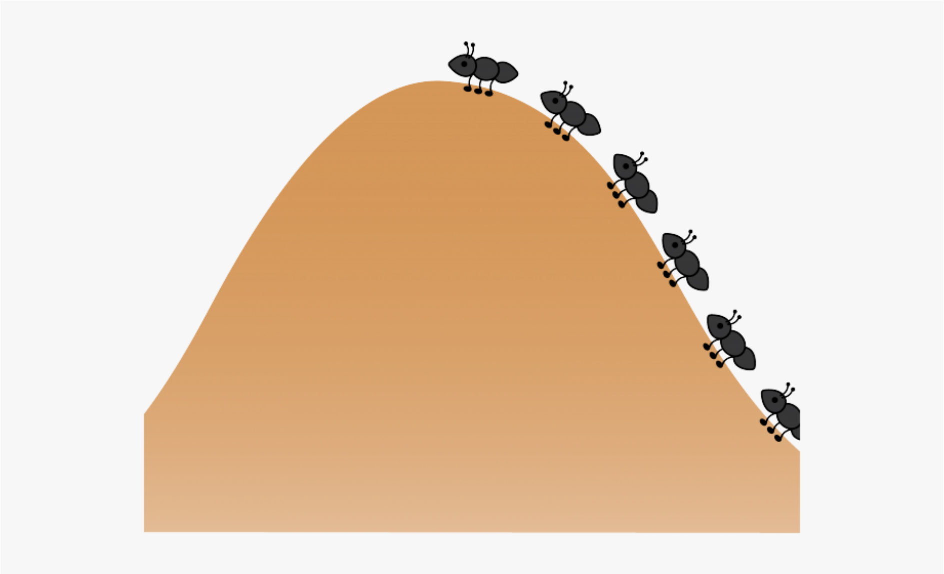


	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston Uniersity	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

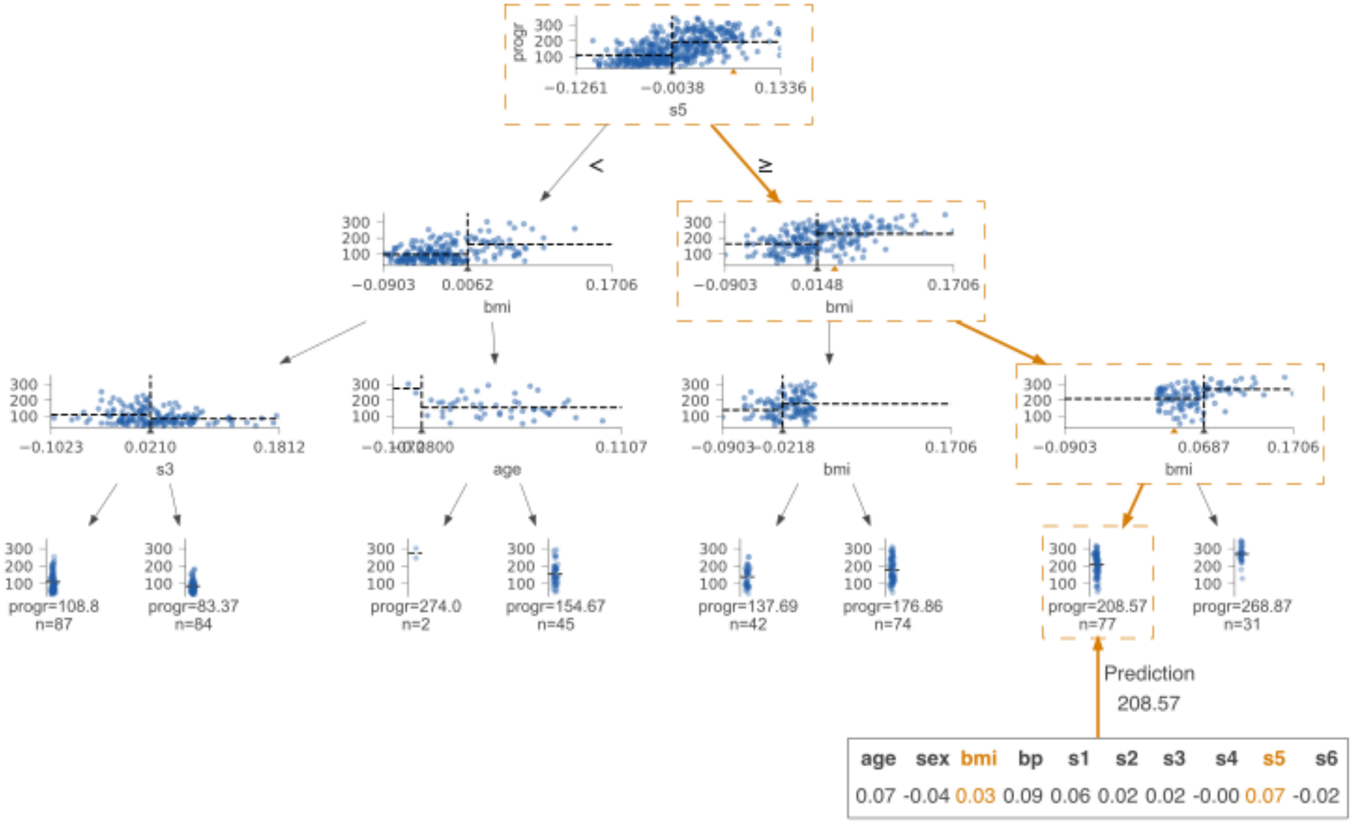
Gradient Boosting, Intuition?

Gradient Boosting, Intuition!

- Gradient boosting is an **ensemble of weak learners**
- Base models are typically **shallow decision trees**
- Like an ant colony, each model makes a small contribution to **residual correction** that builds to a complex prediction system



What are Decision Trees?

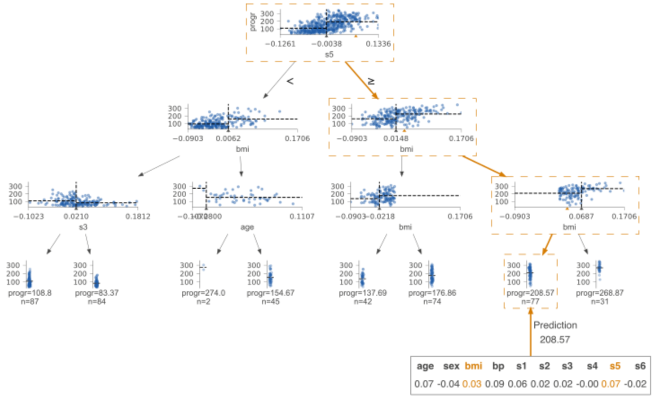


What are Decision Trees?

- Trees generate predictions based on a series of **if/else binary decisions** that recursively partition the feature space

- In regression, these splits are chosen to **minimize variance** within the resulting subsets (nodes)

- Each data point will fall within a **leaf node** that has no further splits, and its target prediction will be the mean of that node

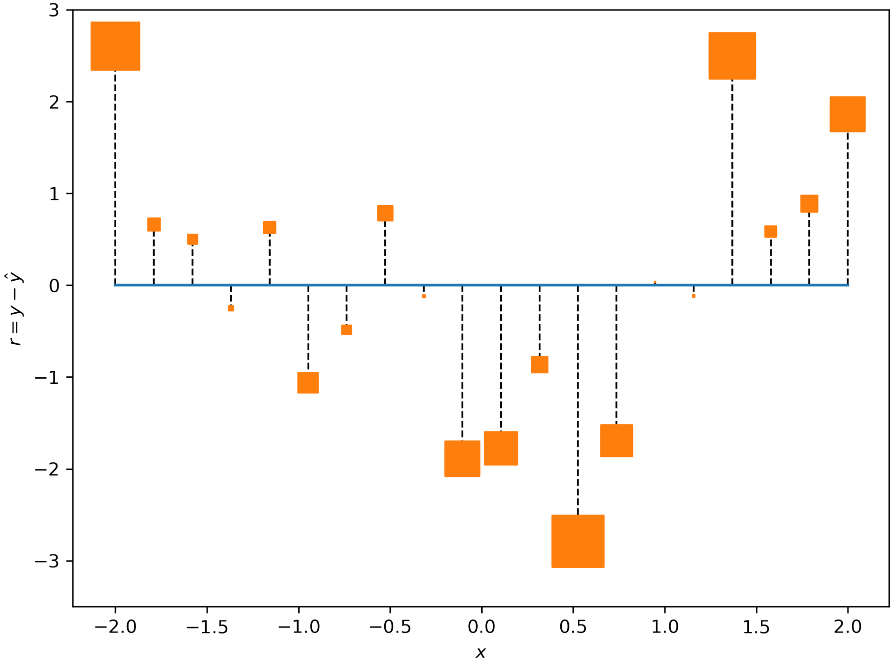


What are Residuals?

- Residuals measure the **difference between actual and predicted** target values (those generated by a model)

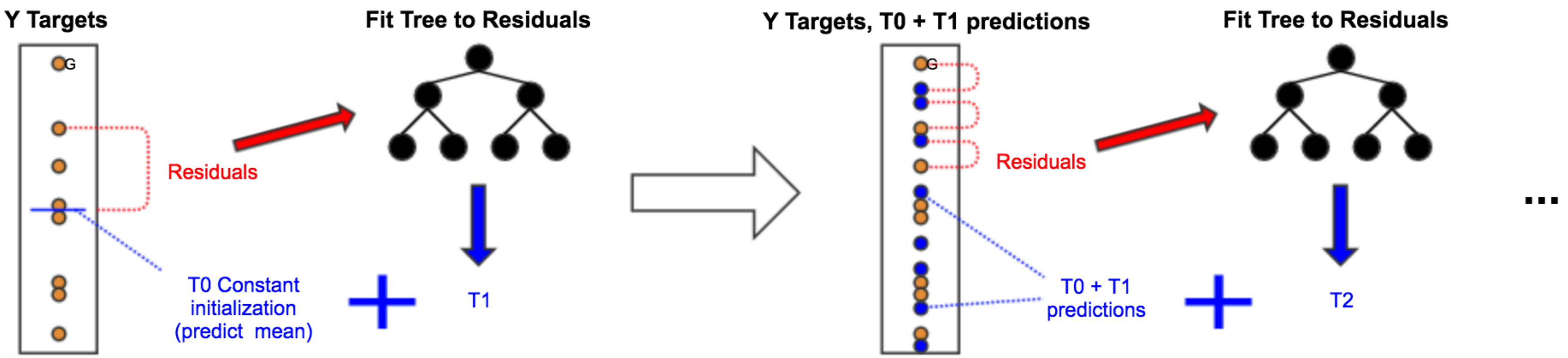
- The **smaller the total magnitude of residuals**, the better a model describes a dataset

- Imagine that we started with a bad model like the **constant model** on the right, but combined it with one that perfectly predicted its error…



Gradient Boosting as a Diagram

- We iteratively reduce our model's current mistakes (residuals) by approximating them with a decision tree, and adding that tree to the ensemble



Model: $F(x) = T0 + T1(x) + T2(x) + ... + Tk(x)$

Gradient Boosting as a Formal Algorithm

- After choosing **hyperparameters k** (number of trees) and **d** (max depth of each tree):

1. Set $T_0 = mean(y)$
2. For $m = 1, \ldots, k$:

   A. Set $r_{m-1} = y - (T_0 + \sum_{j=1}^{m-1} T_j(X))$

   B. Fit max depth d tree T_m with features X, target r_{m-1}

Obtain final model: $F(X) = T_0 + T_1(X) + \ldots + T_k(X)$

Gradient Boosting as a Python Class: Initialization

```
# need these to help construct the model
import numpy as np
from sklearn.metrics import r2_score
from sklearn.tree import DecisionTreeRegressor

class GradientBooster():

    # select hyperparameters at initialization
    def __init__(self, n_estimators=10, max_depth=3):

        self.n_estimators = n_estimators
        self.max_depth = max_depth
```

Gradient Boosting as a Python Class: Fitting

```
def fit(self, X, y):

    # start with constant prediction (mean)
    self.C = np.mean(y)
    self.estimators = []

    resids = y - self.C

    # repeatedly fit to, predict, and update current errors
    for _ in range(self.n_estimators):

        est = DecisionTreeRegressor(max_depth=self.max_depth)
        est.fit(X, resids)
        resids -= est.predict(X)

        self.estimators.append(est)
```

Gradient Boosting as a Python Class: Prediction and Scoring

```python
# predict by summing across all trees and adding original constant prediction
def predict(self, X):

    return self.C + np.sum([est.predict(X) for est in self.estimators], axis=0)

def score(self, X, y):

    return r2_score(y, self.predict(X))
```

```python
booster = GradientBooster(n_estimators=100, max_depth=3)
booster.fit(X_train, y_train)
print('%.3f' % booster.score(X_test, y_test))
```

```
0.796
```

Thank You!

https://github.com/jeddy92/

https://jeddy92.github.io/

METIS