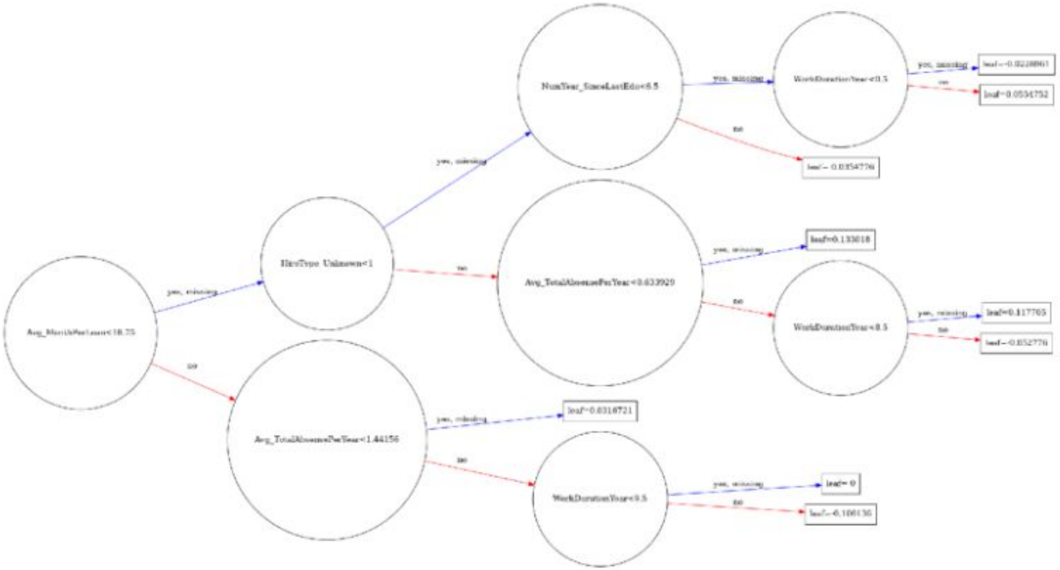
Random Forest and Boost Trees



Agenda

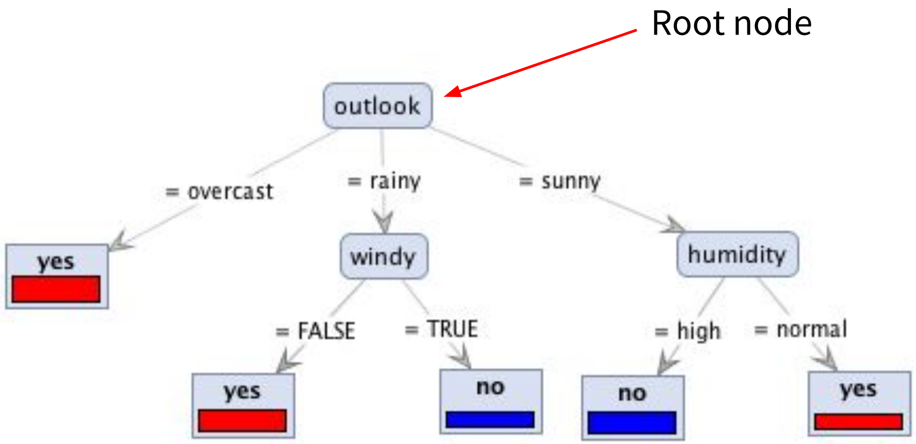
Decision Trees

- Tree Ensemble (Random forest)
- XGBoost

Intro to Neural Networks

Decision Trees

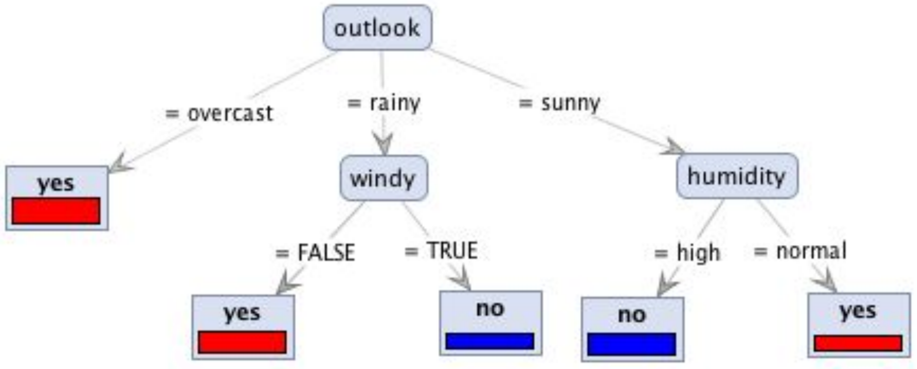
A tree structure that separates data into groups by the feature attributes
Can be used for classification and regression



https://www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html

What's a good decision tree?

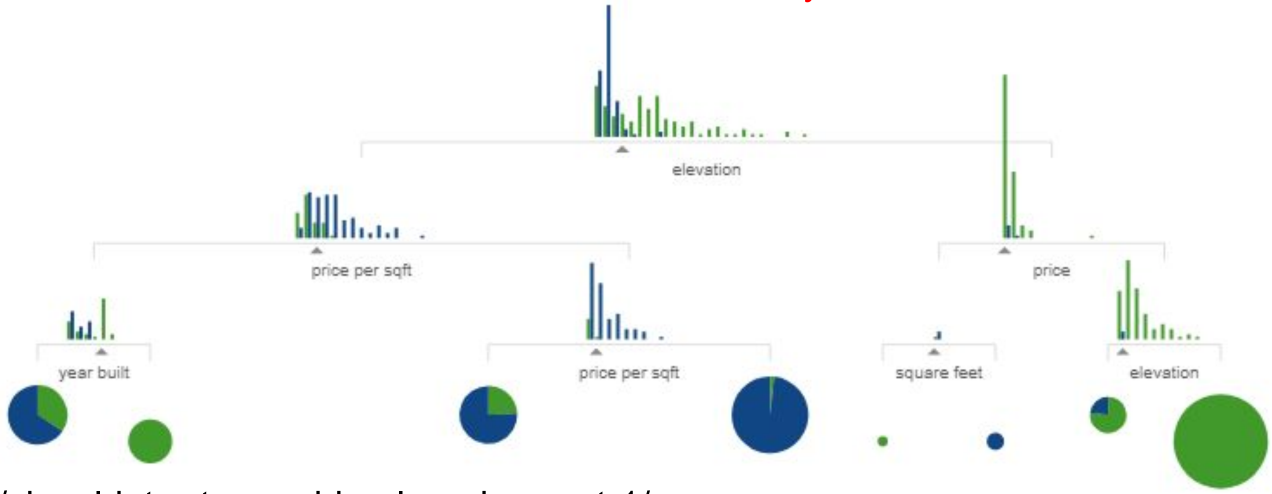
Separates the data nicely
Within a certain budget (smaller trees) - less overfitting



How to create a good decision tree?

Pick the attribute that best separates the classes
Keep doing it until a leave contains entirely one class or you decide it’s not worth it to add more nodes

How to determine the best attribute automatically?



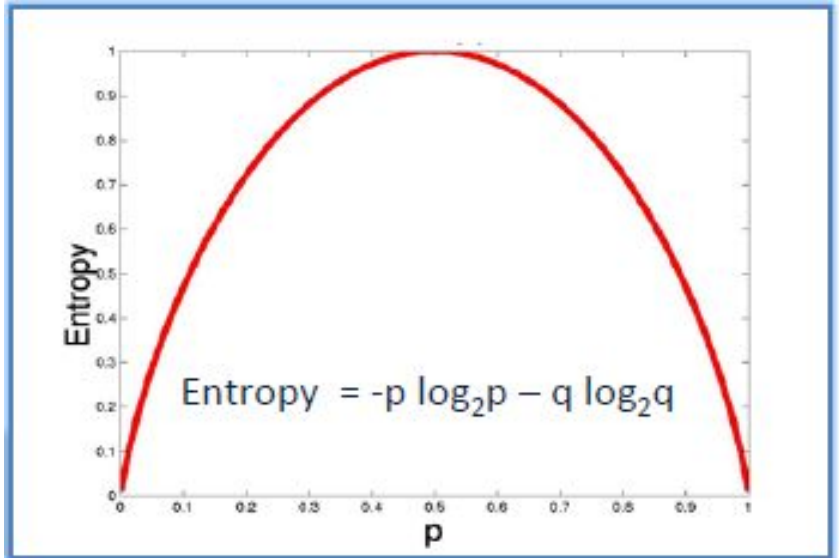
http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Entropy

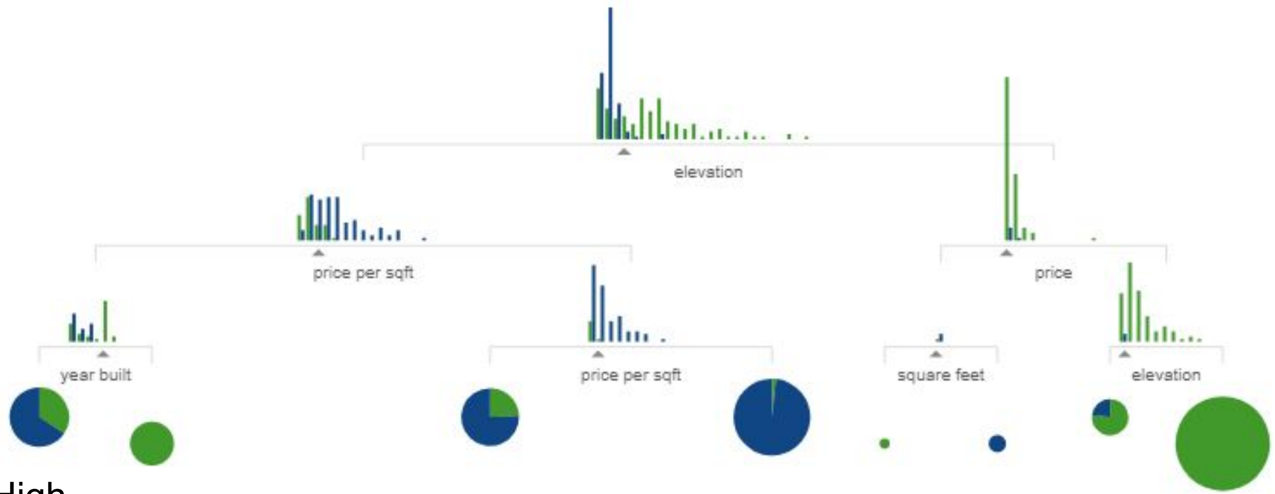
The whole sample space

A measure of randomness

$$E(S) = \sum_c -p(c)\ log_2 p(c) \text{ for } c \in C$$



Entropy



High
entropy

Low entropy

Information Gain (IG)

The whole sample space

A measure of how much entropy is reduced

All child nodes by that attribute

$$E = \sum_c -p(c)\, log_2 p(c) \text{ for } c \in C$$

$$\text{IG (parent,child)} = \text{E(parent)} - \sum_t p(t)E(t) \;\; \text{when } t \in T$$

Probability of going to that child node

Entropy of the child node

Information Gain (IG)

The whole sample space

A measure of how much entropy is reduced

All child nodes by that attribute

$$E = \sum_c -p(c)\, log_2 p(c) \text{ for } c \epsilon C$$

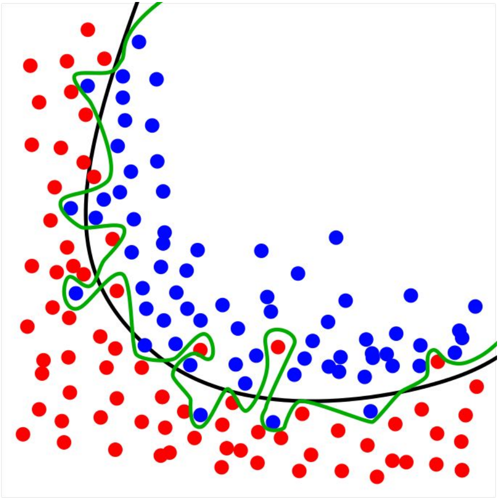
$$\text{IG (parent,child)} = \text{E(parent)} - \sum_t p(t)E(t) \text{ when } t \epsilon T$$

Find the way to split that maximizes IG

Problems with Decision Trees

Can overfitting easily

Susceptible to noise or badly labelled data



TREE ENSEMBLE MODEL

Tree ensemble model

Ensemble types are models that combine multiple models together

A group of experts voting on a subject

Can lead to less overfitting

Tree ensemble = Multiple trees = Random Forest!

Bagging

Create multiple subsets of data
Each subset is used to train a different tree
The final answer is the average or mode

Less overfitting and can handle mislabeled data

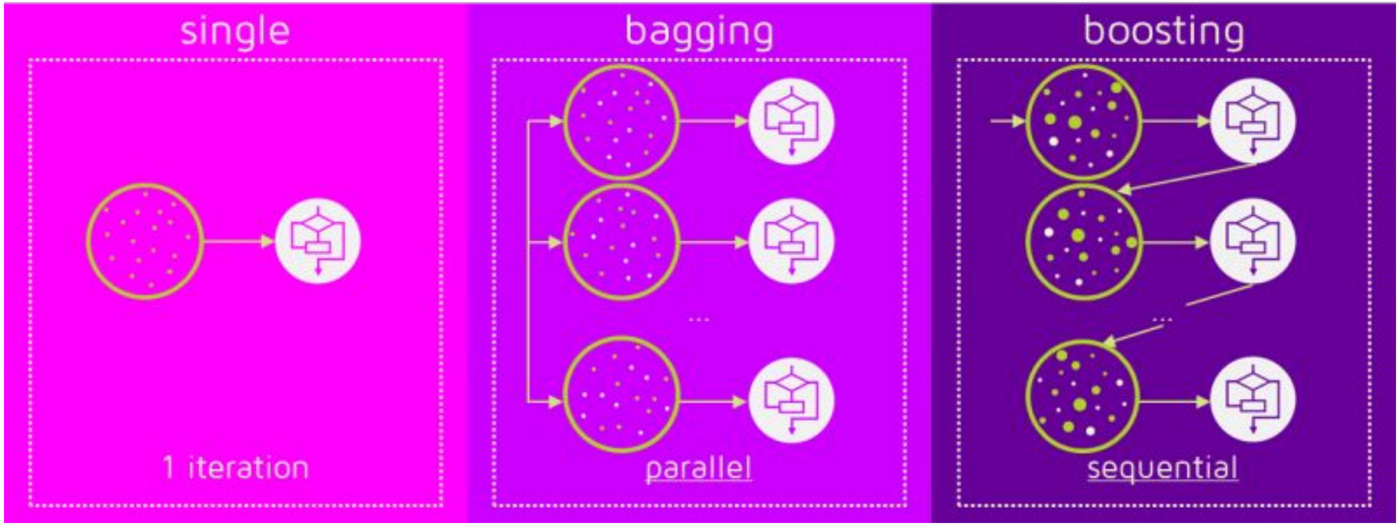
Random Forest

We can also use bagging on features
Each tree has different training samples AND set of features

Boosting vs Bagging

Boosting is another way to create multiple trees

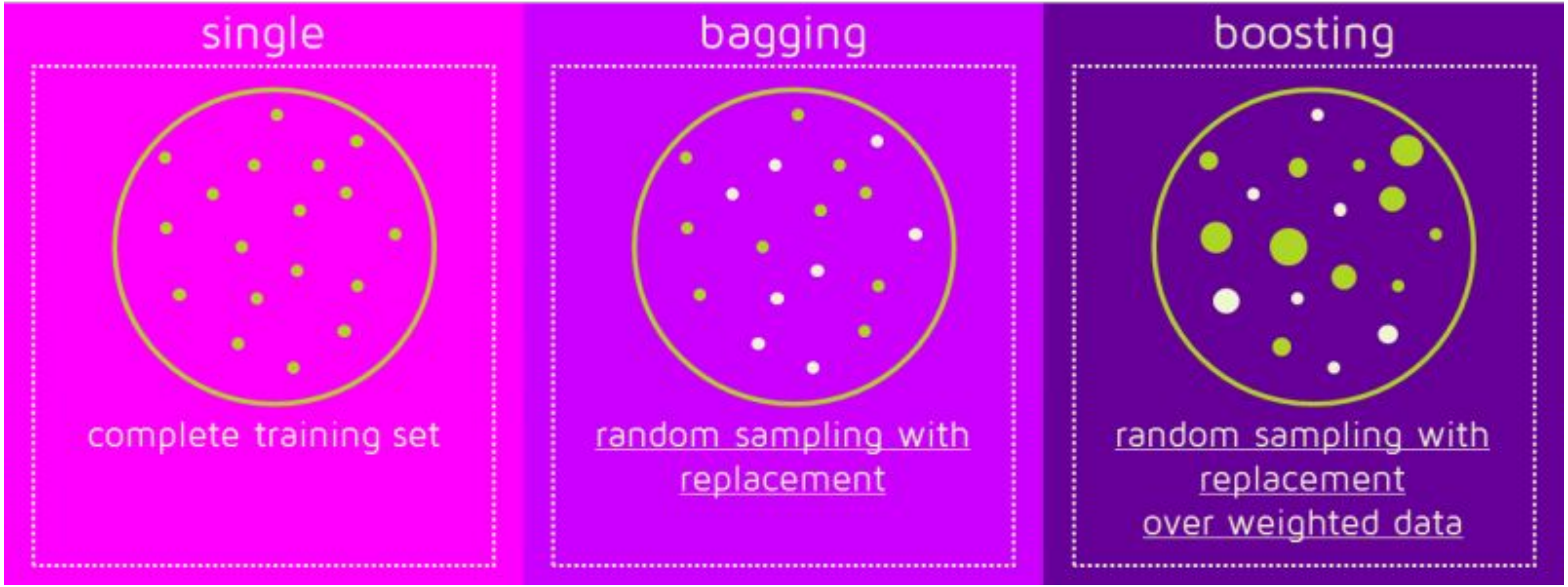
But boosting is iterative, the next tree is based on the errors from the previous trees



https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

Boosting vs Bagging

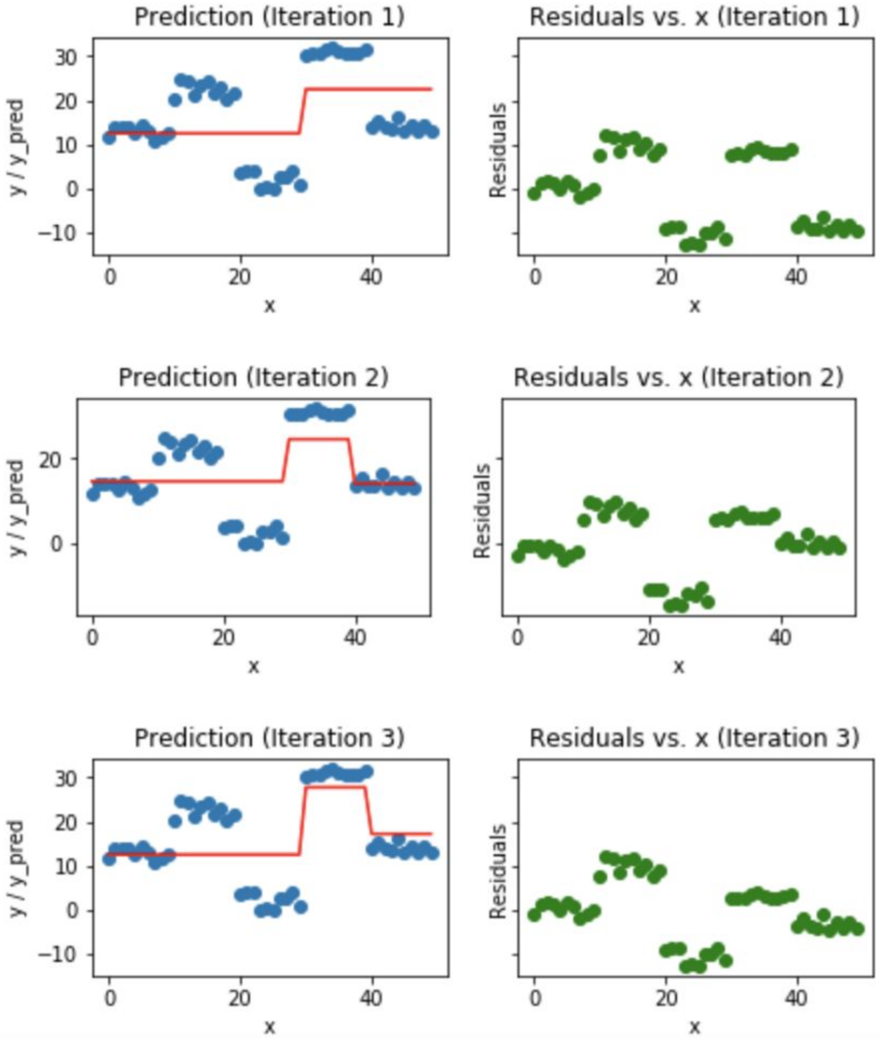
The selection of training data (bagging process) is based on the previous errors

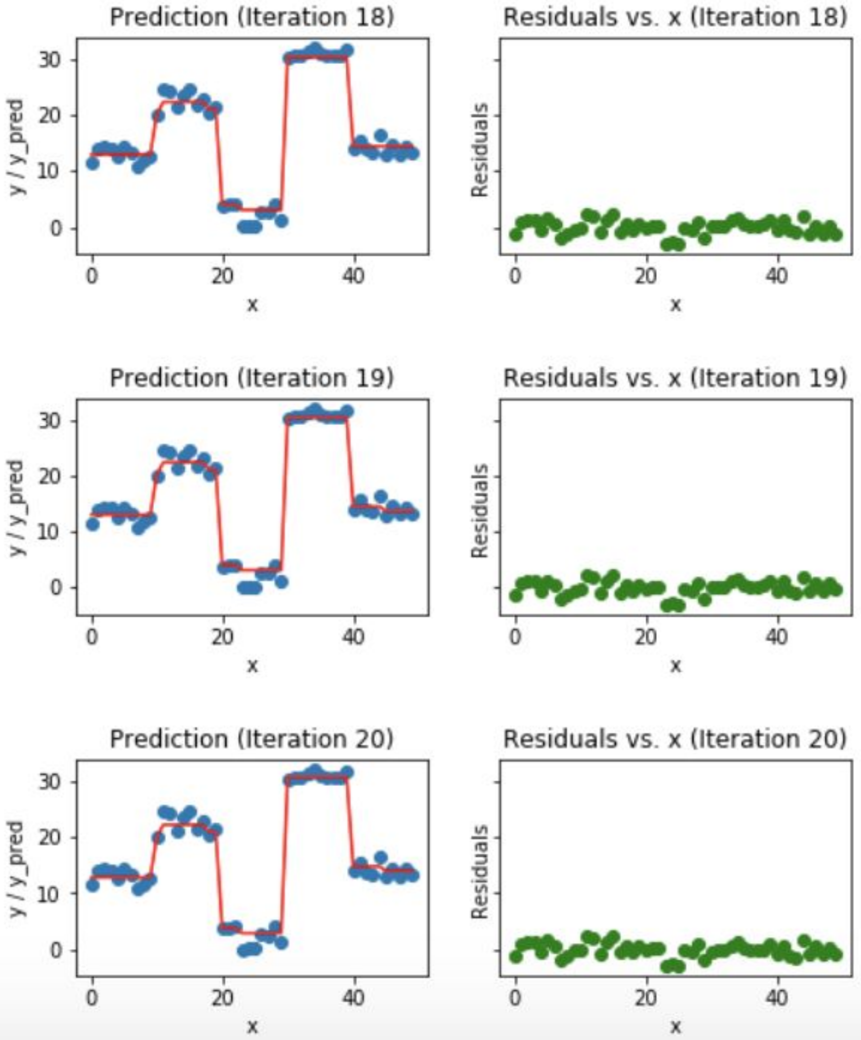


https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

Gradient Boosting

A method of boosting that use gradient-based methods

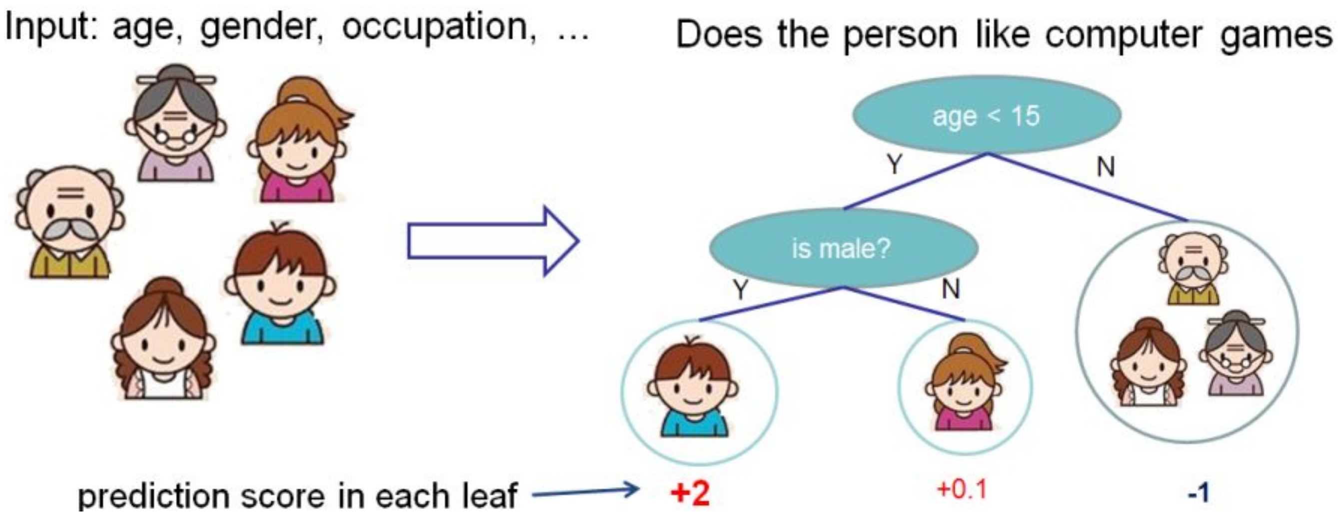

Prediction (Iteration 1)
Residuals vs. x (Iteration 1)
Prediction (Iteration 2)
Residuals vs. x (Iteration 2)
Prediction (Iteration 3)
Residuals vs. x (Iteration 3)
y / y_pred
Residuals
x


Prediction (Iteration 18)
Residuals vs. x (Iteration 18)
Prediction (Iteration 19)
Residuals vs. x (Iteration 19)
Prediction (Iteration 20)
Residuals vs. x (Iteration 20)
y / y_pred
Residuals
x

Tree Gradient Boosting

Similar to decision tree

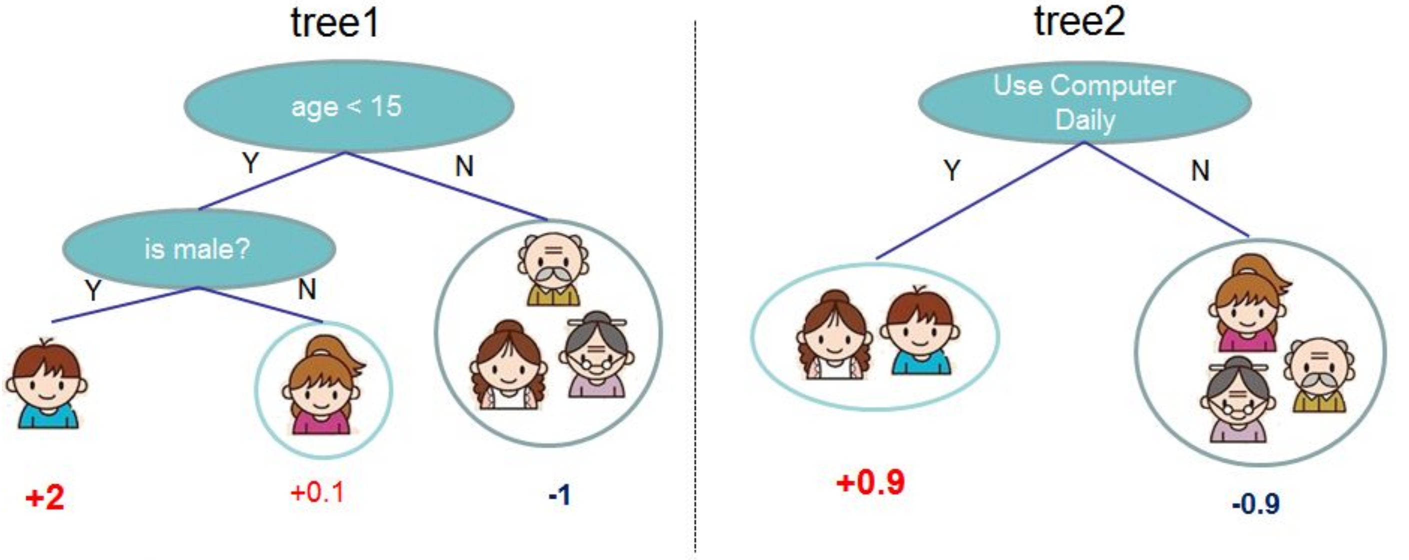
Difference is the leaf node contains a score



https://xgboost.readthedocs.io/en/latest/tutorials/model.html

Tree Gradient Boosting

Multiple trees with different rules. The subsequent tree try to correct the errors from the previous trees



f() = 2 + 0.9= 2.9 f()= -1 - 0.9 = -1.9

Extreme Gradient Boosting (XGBoost)

Super popular Tree Boosting library

Highly recommended for spreadsheets type of input data

```
model = XGBClassifier(
    n_jobs=16,
    n_estimators=400,
    max_depth=4,
    objective="binary:logistic",
    learning_rate=0.07,
    subsample=0.9,
    min_child_weight=6,
    colsample_bytree=.9,
    scale_pos_weight=0.8,
    gamma=8,
    reg_alpha=6,
    reg_lambda=1.3)
```

Objective <- type of problem you want to solve

Max_depth <- max depth of tree, higher more overfitting
Min_child_weight <- how strong must the leave be, higher less overftting
Gamma <- when to stop splitting early

Reg_alpha, reg_lambda <- reduce overftting

Scale_pos_weight <- weight for class imbalance

https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

Notes on feature encoding

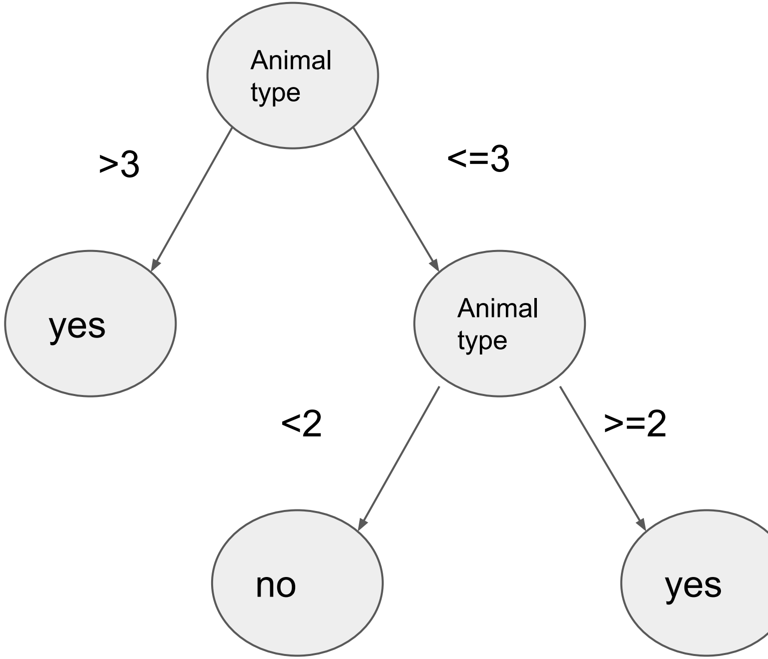
Categorical features does not mean anything

Type of animal

Animal type =
 1 if mouse
 2 if bird
 3 if dog
 4 if insect

Makes it hard to do decision trees

Is it green?



One hot encoding

Split categorical features into multiple binary features

Type of animal (as one hot)

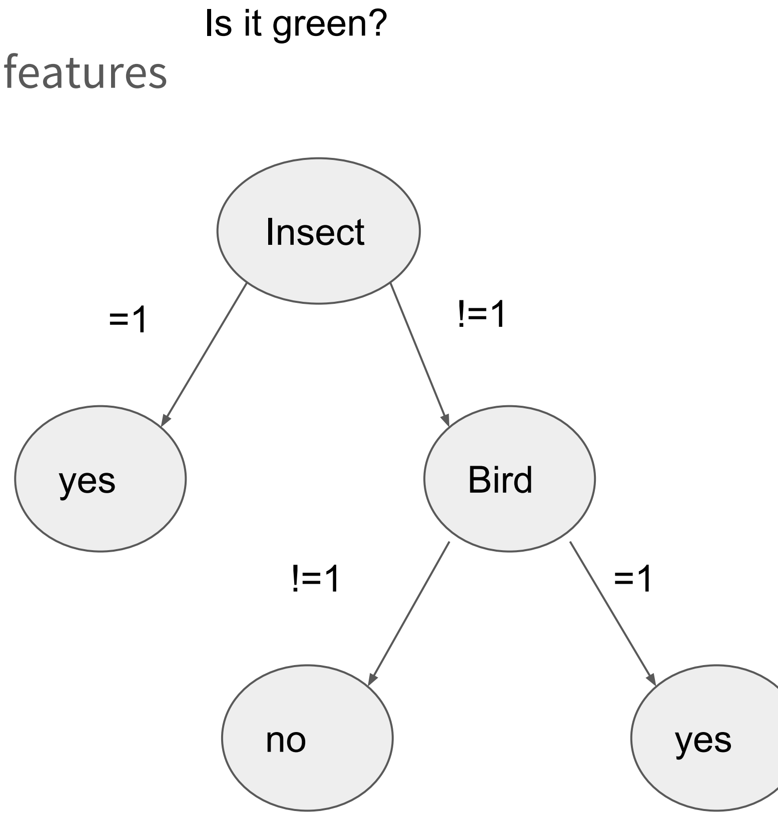
Is_mouse = (0,1)

Is_bird = (0,1)

Is_dog = (0,1)

Is_insect = (0,1)

Doesn’t change much



Target encoding

Encode information by looking at how the feature correlates with the final answer

$$\textit{Encoded feature} = P(\textit{answer} = \textit{yes} \mid \textit{feature value})$$

Animal type =

0 if mouse

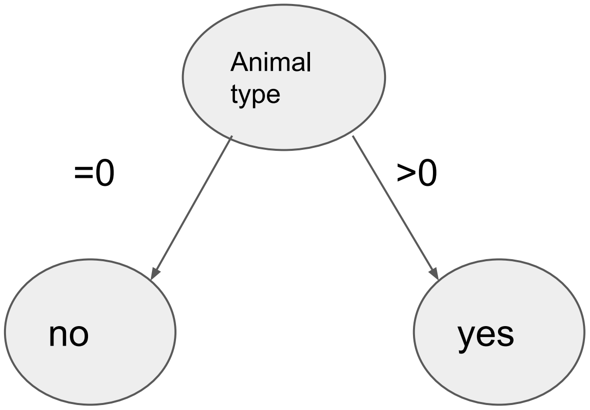
0.3 if bird

0 if dog

0.5 if insect

Need some further smoothing to improve this.

Is it green?



https://dl.acm.org/citation.cfm?id=507538

Other XGBoost variants

LightGBM

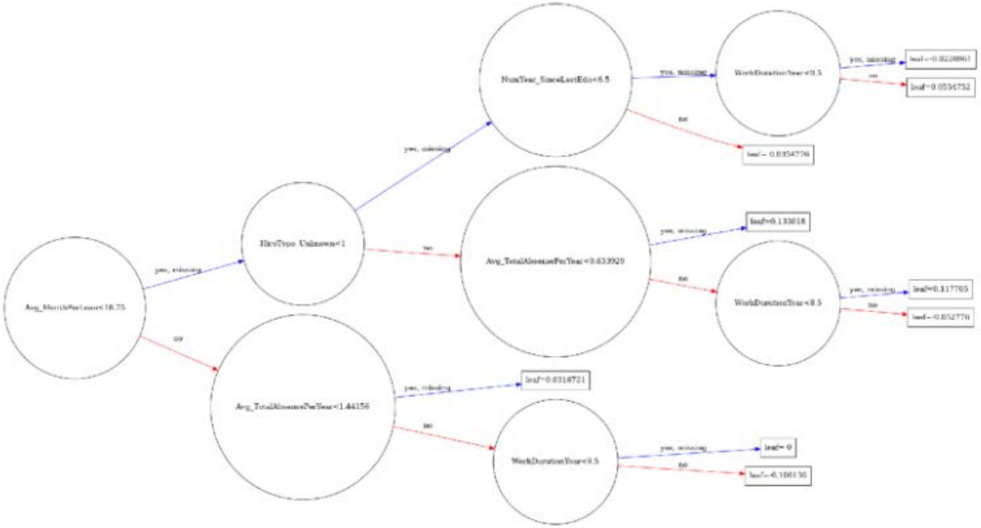
CatBoost

Different ways to handle categorical encoding.

Different ways to do node splitting (faster)

https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db

Lab



HR data

Class imbalance

XGboost

- Encoding
- Visualizing trees and feature importance