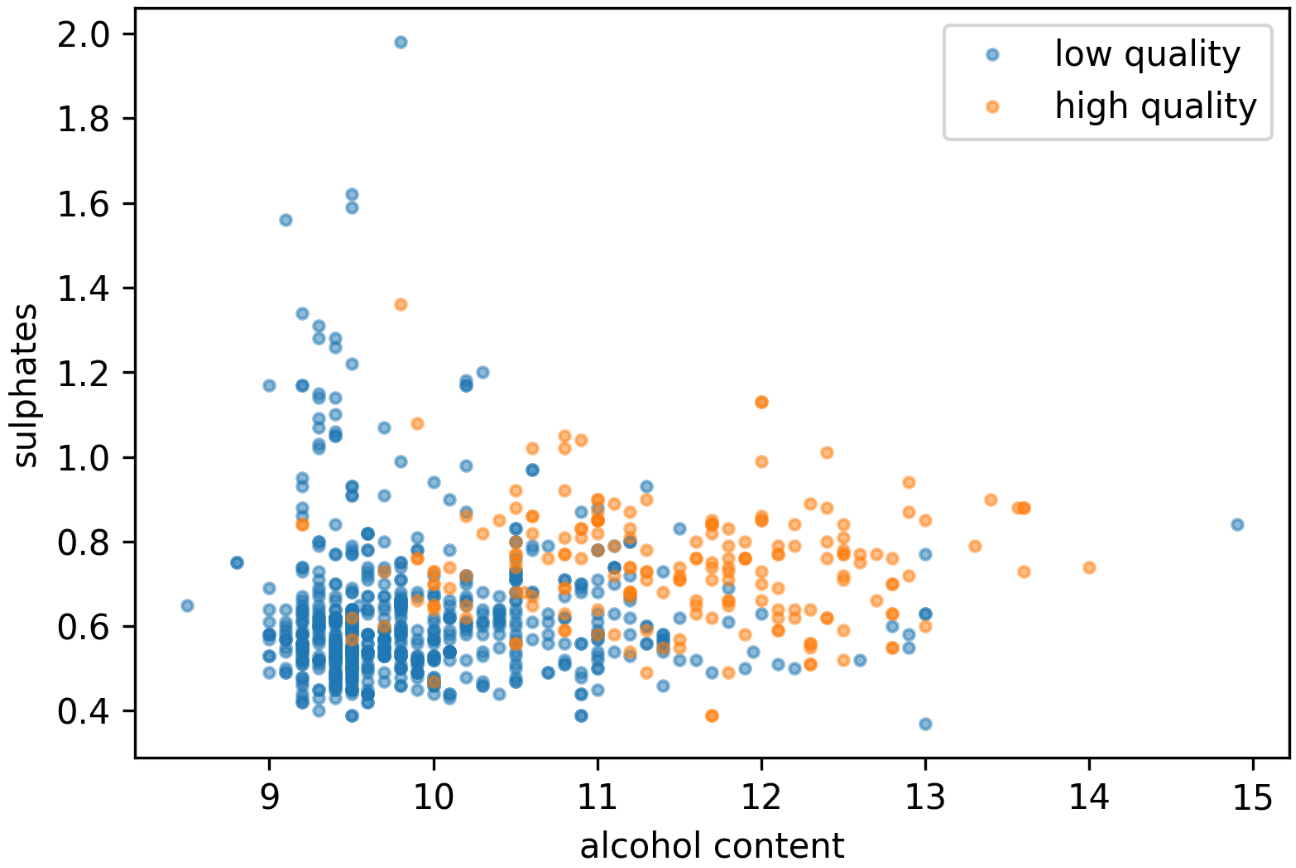
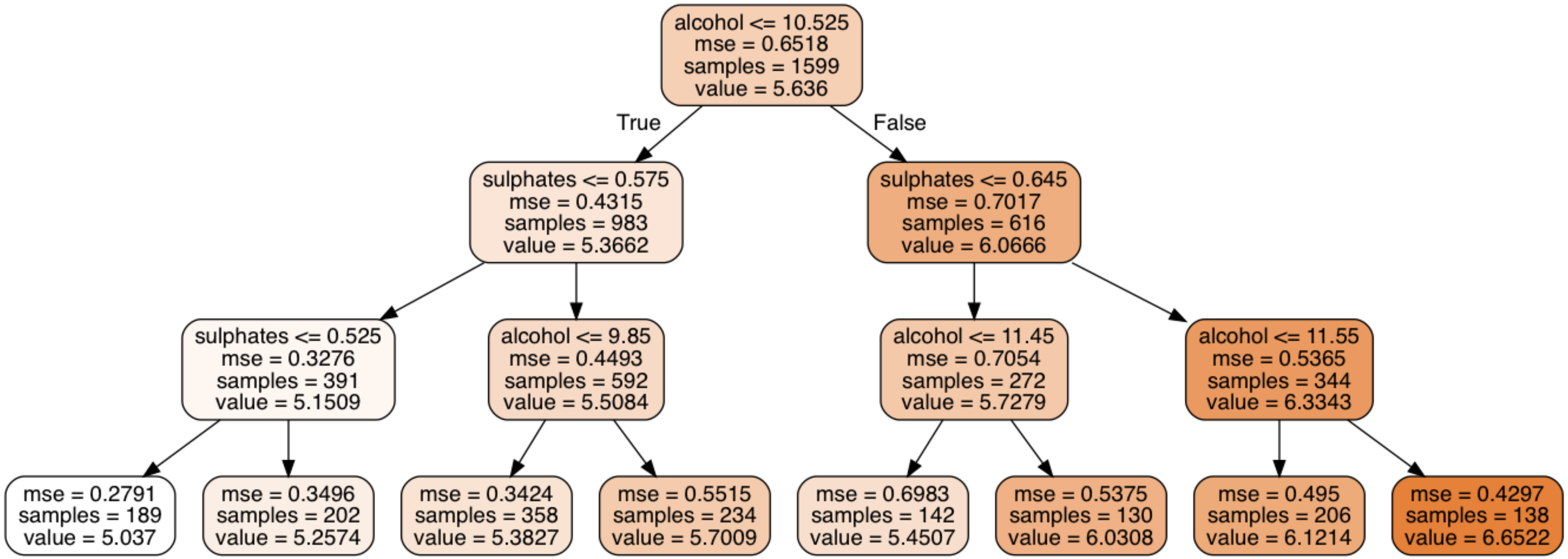
STK-INF3000/4000 - WEEK 10 - TREES

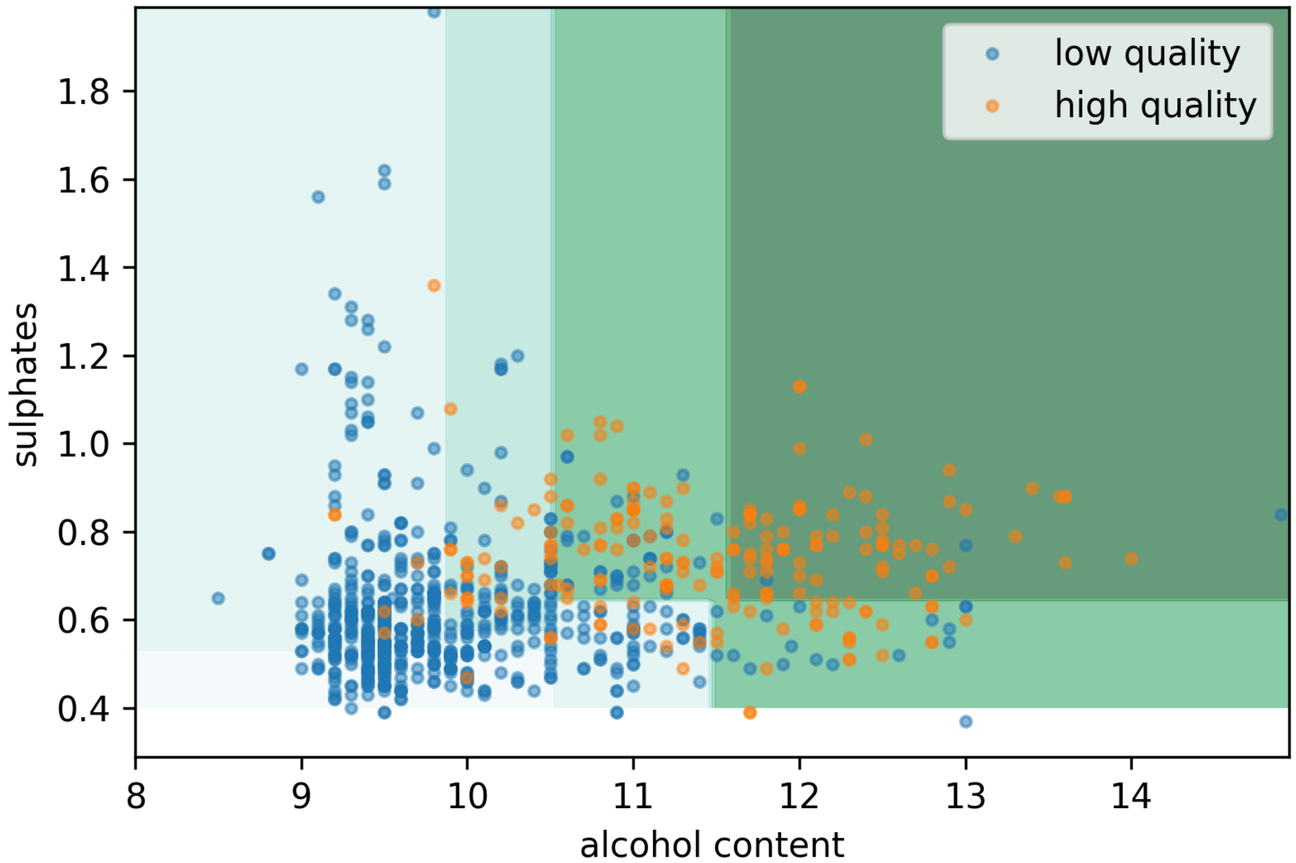
WINE QUALITY



WINE QUALITY TREE



WINE QUALITY REGRESSION TREE



WHY TREES?

- Simple.
- Easy to explain.
    - Especially to non-experts.
- Powerful.

CALCULATING TREES

- Divide your data $R_L(j,s) = \{X | X^{(j)} \leq s\}$, $R_R(j,s) = \{X | X^{(j)} > s\}$.
- Find the best a_R, a_L, j, s to minimize

$$\sum_{i, x_i \in R_L(j,s)} (a_L - y_i)^2 + \sum_{i, x_i \in R_R(j,s)} (a_R - y_i)^2$$

- For given j, s, we find that $a_{R,L} = \underset{i, x_i \in R_{R,L}}{\operatorname{avg}} y_i$.
- Repeat on the sub-sets.
  - Until a maximum depth is reached.
  - Until a minimum number of samples is reached.

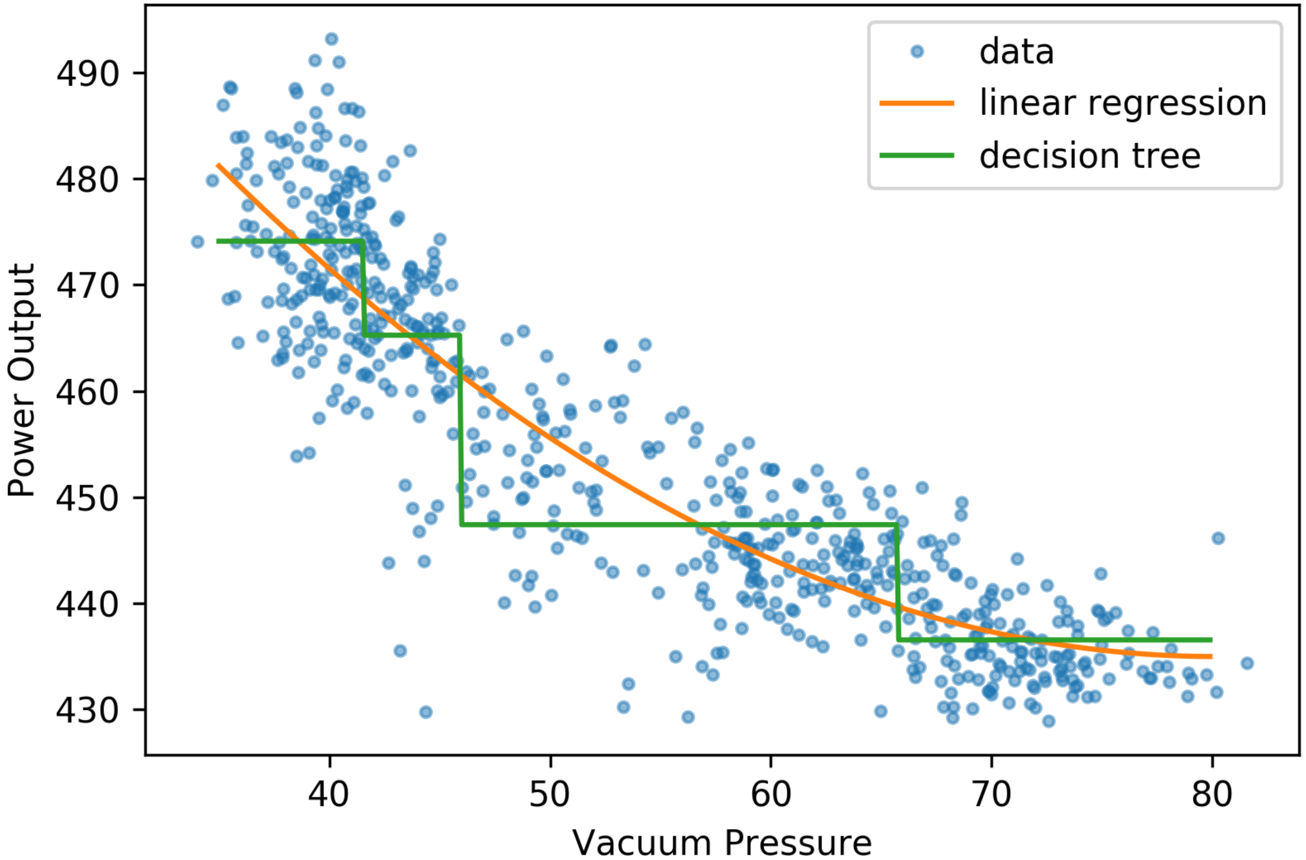
REGRESSION TREE

Our resulting model reads

$$\hat{f}(X) = \sum_m c_m I\{X \in R_m\}.$$

Hence trees are an example of a general class of *additive models*.

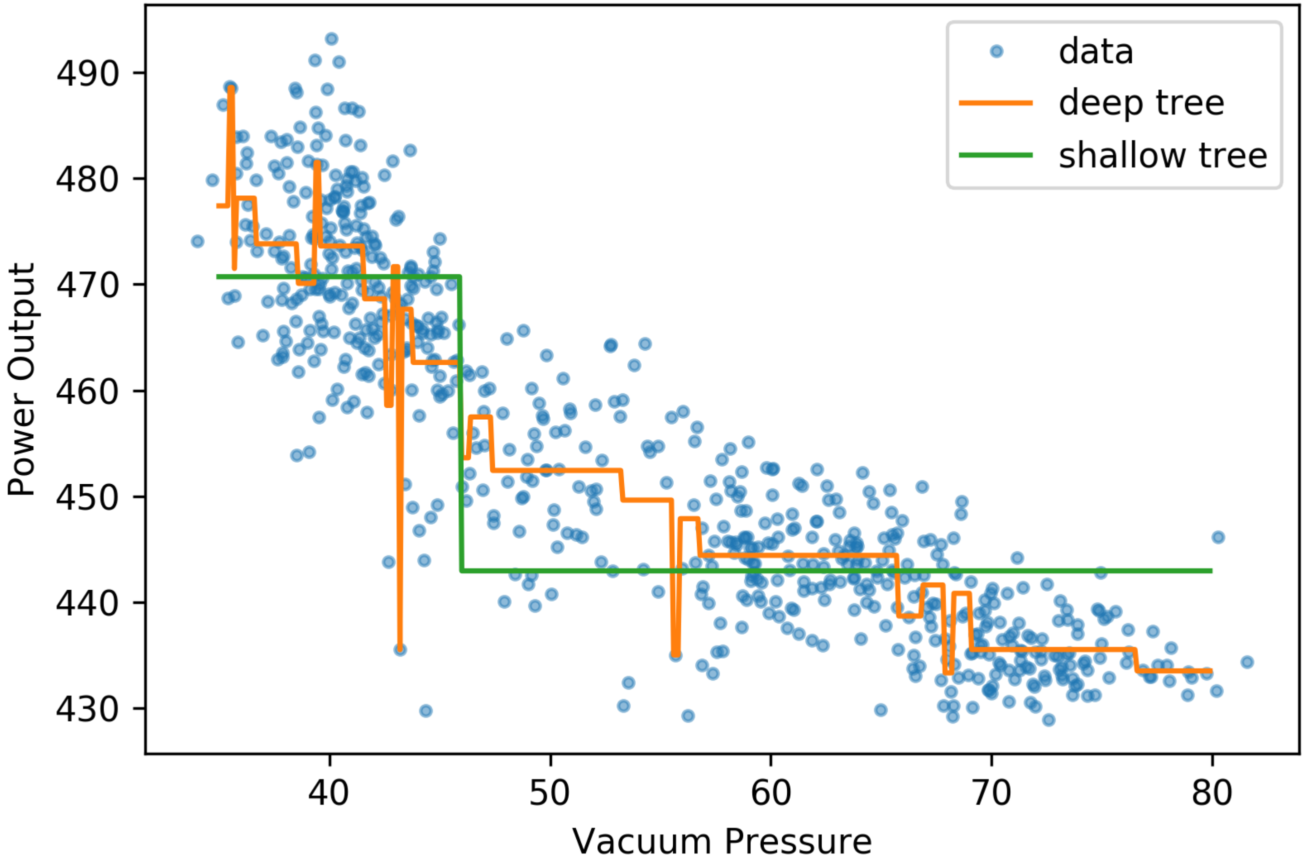
TREE VS LINEAR REGRESSION



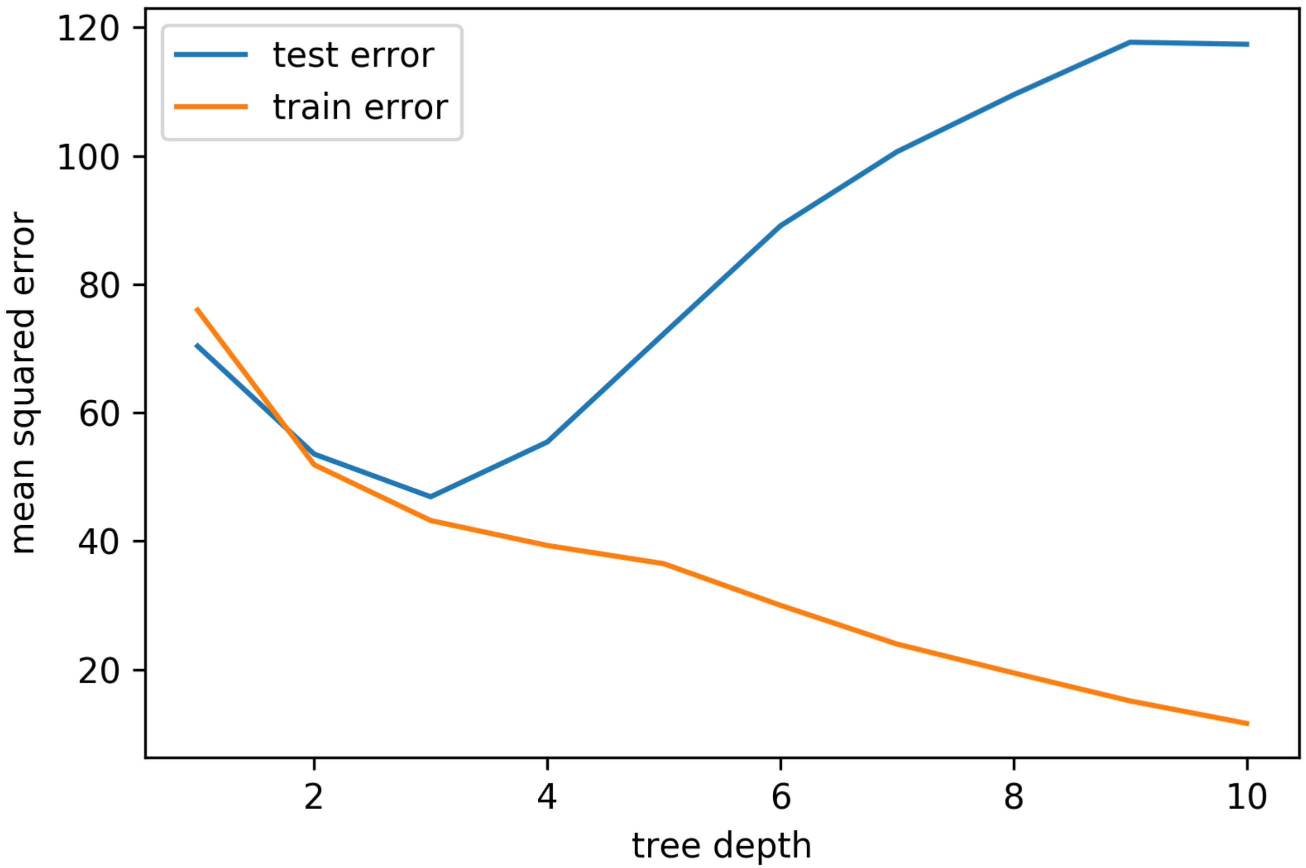
HOW DEEP SHOULD YOU GO?

- Deep trees have many degrees of freedom and hence high **variance**.
- Too shallow trees can't capture the *shape* of the data.
  - Hence have high **bias.**

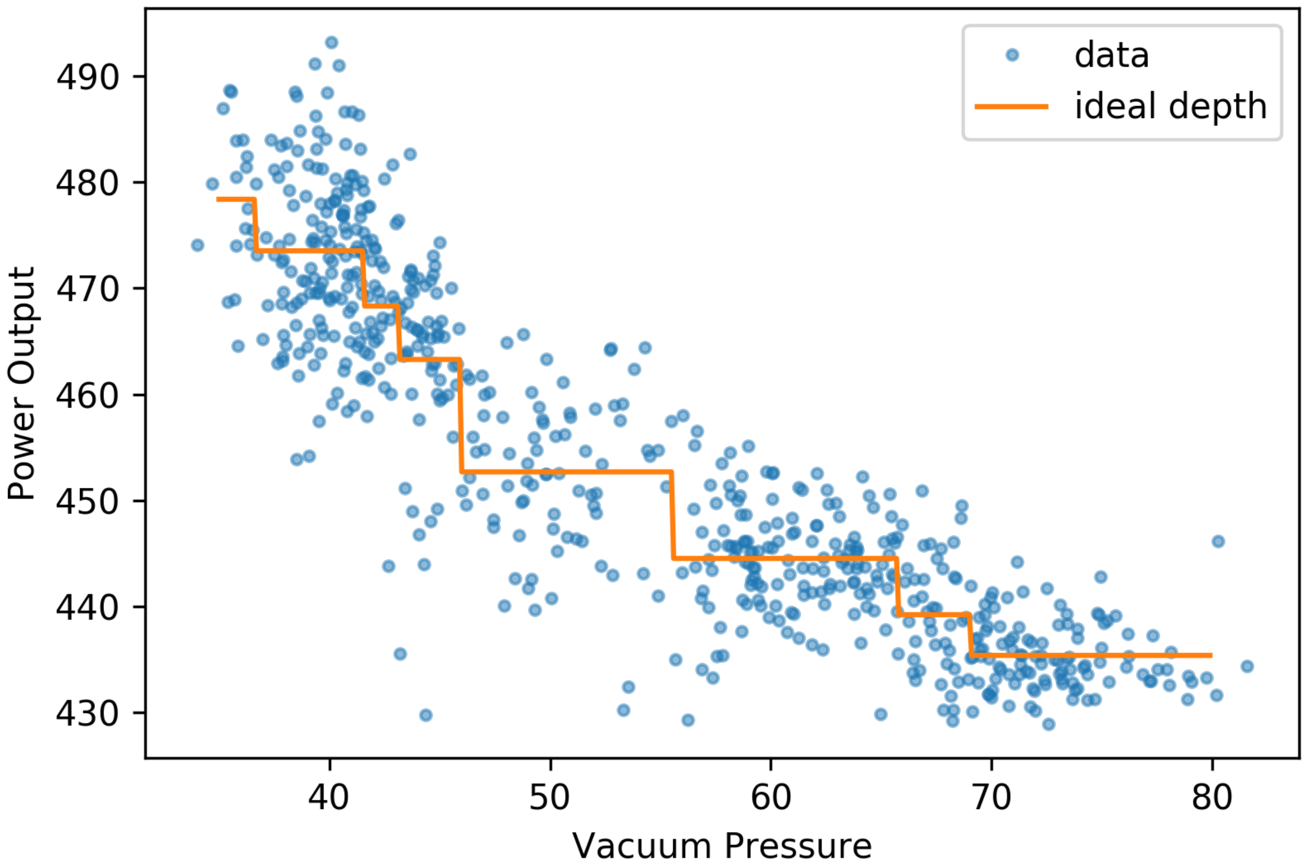
BIAS-VARIANCE TRADE-OFF FOR TREES



TRAINING AND TEST ERROR



THE BEST TREE



TREES FOR CLASSIFICATION

Just *modifying* our tree formulas to use the **mode**

$$a_{R,L} = \underset{i, x_i \in R_{R,L}}{\text{mode}} \; y_i$$

yields a classification algorithm.

HOW FIND THE SPLITS FOR CLASSIFICATION?

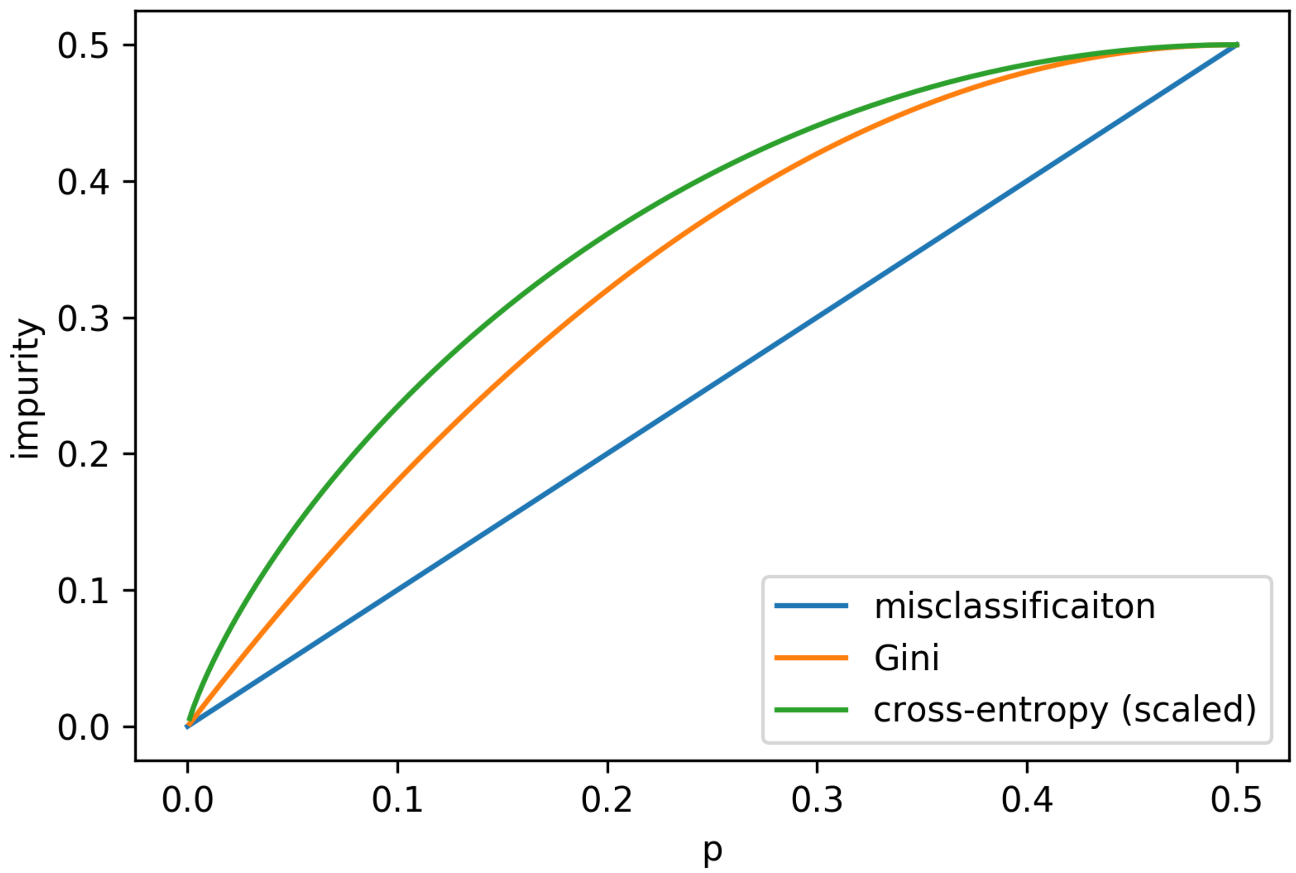
Define

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{i; x_i \in R_m} I(y_i = k),$$

such that $k(m) = \mathrm{argmax}_k \, \hat{p}_{mk}$

- Misclassification: $1 - \hat{p}_{mk(m)}$.
- Gini index: $\sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$.
- Cross-entropy: $-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$.

TWO-CLASS IMPURITY MEASURES



TREE PRUNING

- Stopping criteria:
  - Depth d.
  - Terminal node size.
  - Maximum split size.
  - Minimum impurity.
- Stopping at given d or impurity threshold might miss good splits later on.
- Often better to stop at e.g. minimal node size 10.
- Prune resulting tree.

PRUNING BY COMPLEXITY

- Fitted tree T_0, let T be subtree with $|T|$ terminal nodes R_m.

$$
\begin{aligned}
N_m &= |\{x_i \in R_m\}| \\
\hat{c}_M &= \frac{1}{N_m} \sum_{x_i \in R_m} y_i \\
Q_m(T) &= \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2 \\
C_\alpha(T) &= \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|
\end{aligned}
$$

- For each α, $\exists$ unique smallest subtree minimizing C_α.

PRACTICAL CONSIDERATIONS.

- Categorical inputs.
  - *Many* possible splits.
  - Easy for binary targets.
- Loss matrix.
  - L_{kl} loss for classifying k as l.
  - Classify $k(m) = \mathrm{argmin}_k \sum_l L_{lk} \hat{p}_{ml}$.
- Missing values.
- Multiple child nodes.
- Smoothness.
- Variance.

ENSEMBLE METHODS

- Reduce over-fitting and increase accuracy by using *multiple* models.
  - Reduce variance.
  - Possibly increase bias.
- Most commonly used:
  - Boosting.
  - Bagging.