Machine Learning for biology

V. Monbet

UFR de Mathématiques
Université de Rennes 1

Outline

Outline

Outline

Outline

Outline

Outline

Outline

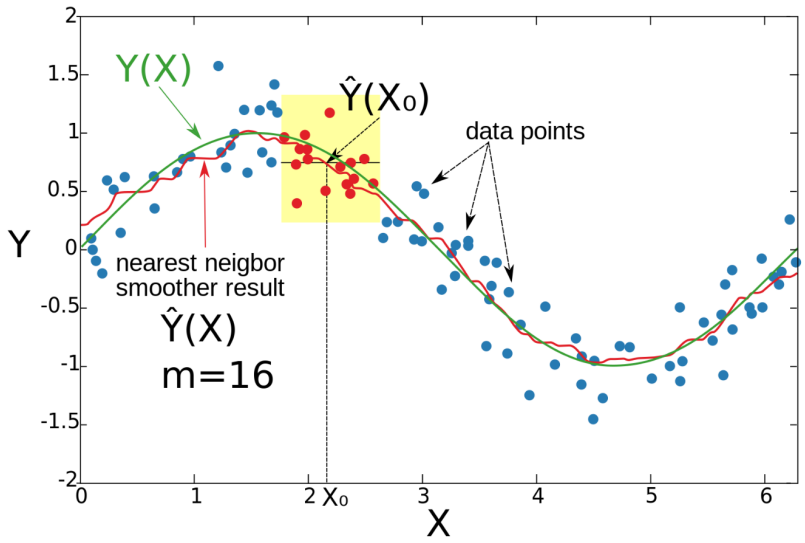
Outline

Making Predictions with kNN

- k nearest neighbors algorithm (kNN) makes data driven predictions.
- Predictions are made for a new instance $\mathbf{x}_0 \in \mathcal{X}$ by searching through the entire training set for the k most similar instances (the k neighbors) and summarizing the output variable for those k instances.

$$\hat{Y}(\mathbf{x}_0) = \frac{1}{k} \sum_{i|\mathbf{x}_i \in NN(\mathbf{x}_0)} y_i$$

where $NN(x_0)$ denotes the neighborhoor of x_0.



Finding the neighbors

- To determine which of the k instances in the training dataset $\mathcal{S}_{\mathcal{X},n} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ are most similar to a new input a distance measure is used.
- For real-valued input variables, the most popular distance measure is Euclidean distance.

$$d(\mathbf{x}_{i'}, \mathbf{x}_i) = \sqrt{\sum_{j=1}^{p} (x_{i'j} - x_{ij})^2}$$

- To deal with a mix of numerical and categorical inputs, a specific distance has to be used

$$d(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{p_{num}} (x_j - x'_j)^2 + \gamma \sum_{j=p_{num}+1}^{p_{num}+p_{cat}} \delta(x_j, x'_j)$$

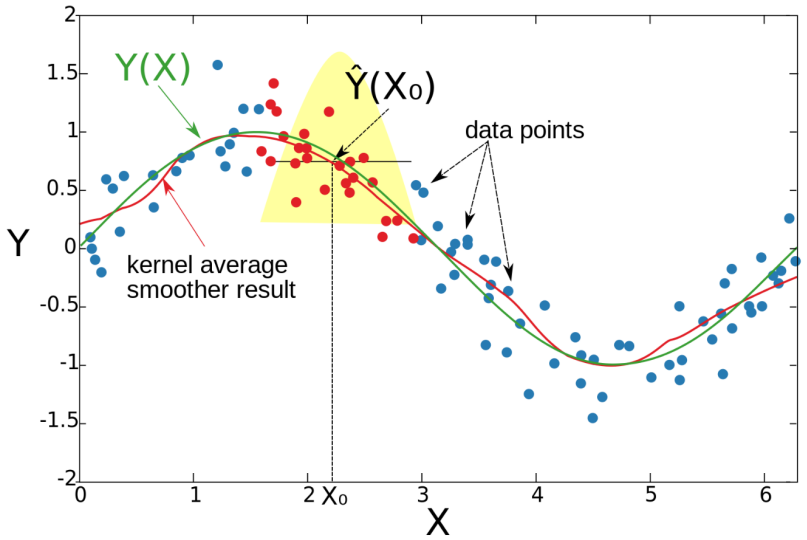
where $\delta(x_j, x'_j)$ is a distance for categorical variables and γ a constant to choose to correctly balance both distances.

Reference: Huang (1998), Data Mining and Knowledge Discovery 2, 283-304

kNN smoother with weights

- Smoother predictions can be obtained by the introduction of weights.
- Let K be a kernel defined by $K_\lambda(x, x') = \frac{1}{\lambda} K\left(\frac{x - x'}{\lambda}\right)$ prediction is given by

$$\hat{Y}(\mathbf{x}_0) = \frac{\sum_{i=1}^n K_\lambda(\mathbf{x}_0, \mathbf{x}_i) y_i}{\sum_{i=1}^n K_\lambda(\mathbf{x}_0, \mathbf{x}_i)}$$



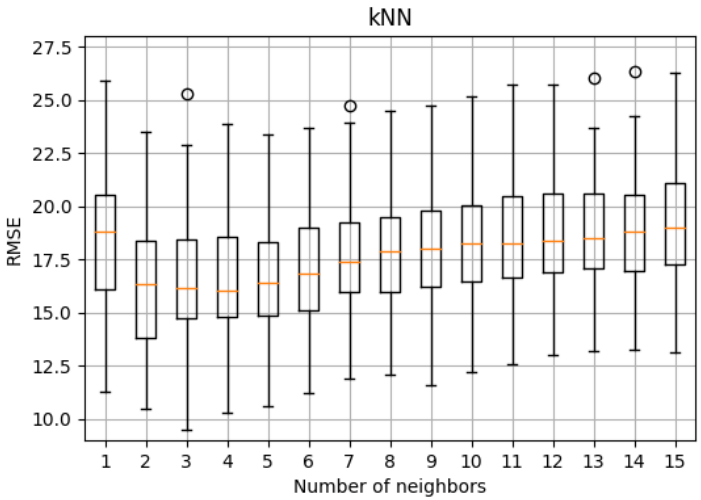
- The larger λ, the smoother the prediction.
- The Gaussian kernel is the most widely used in machine learning

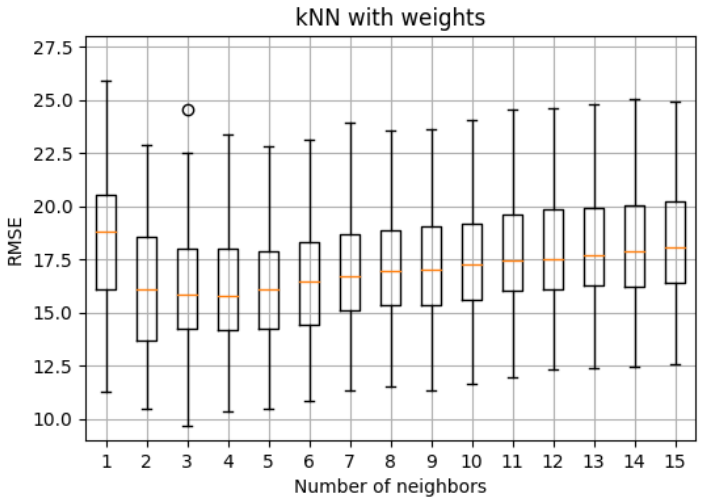
$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} x^2\right)$$

- The quality of the kNN prediction depends on the choice of k (resp. λ). It can be done by cross-validation.

Example: Ozone

- RMSE computed by Monte Carlo cross-validation (50 repetitions) for increasing number of neighbors (from 1 to 15), with uniform and Gaussian weights.
- Inputs: all the numerical variables (Euclidean distance)



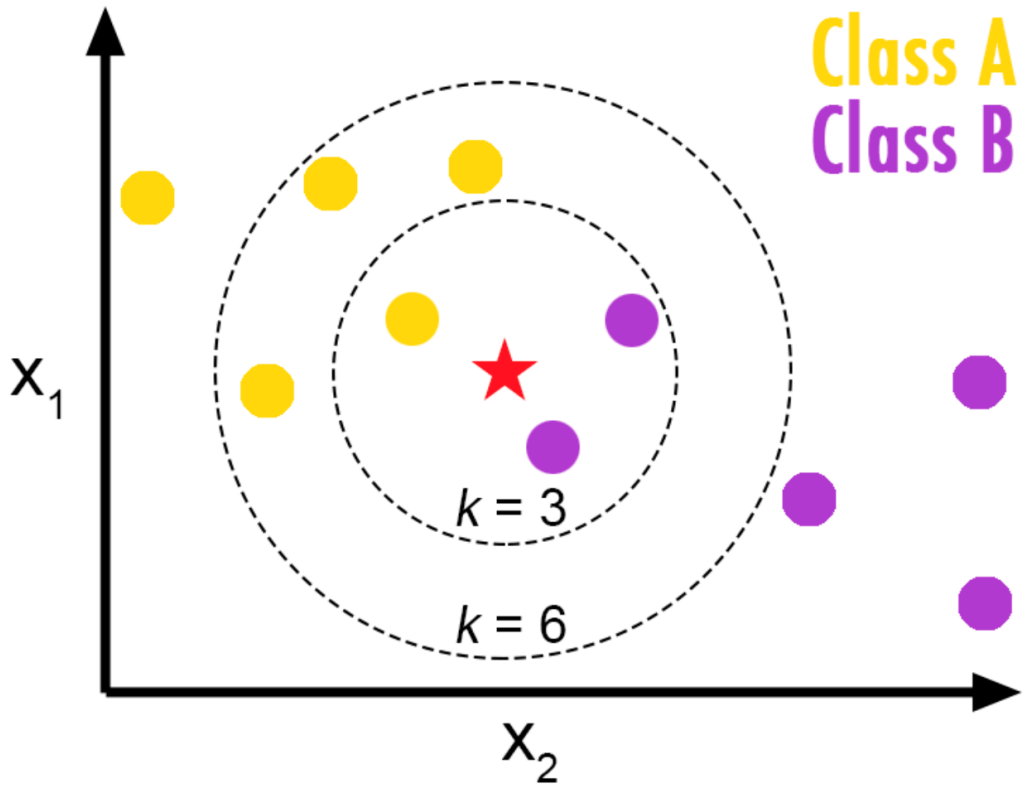


- Best number of neighbors $k^* = 3$

Prediction of a categorical variable

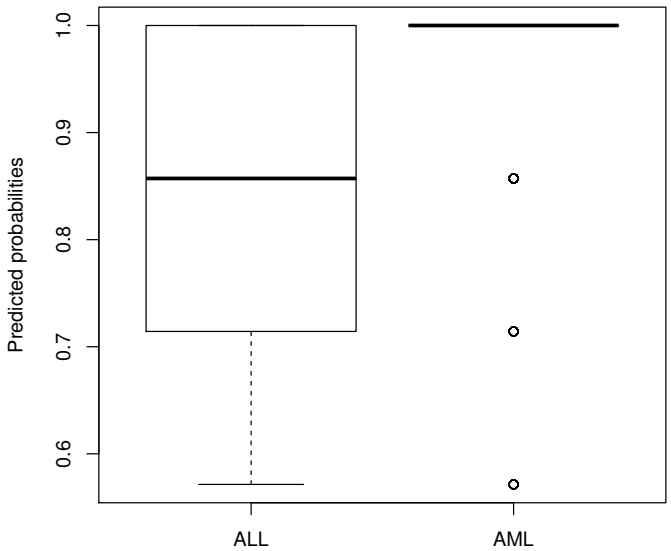
- If Y is qualitative (or categorical) and takes its values in $\{1, \cdots, K\}$, one predicts $\hat{y} = arg\max_k P(Y = k|\mathbf{X} = \mathbf{x})$
- The kNN algorithm

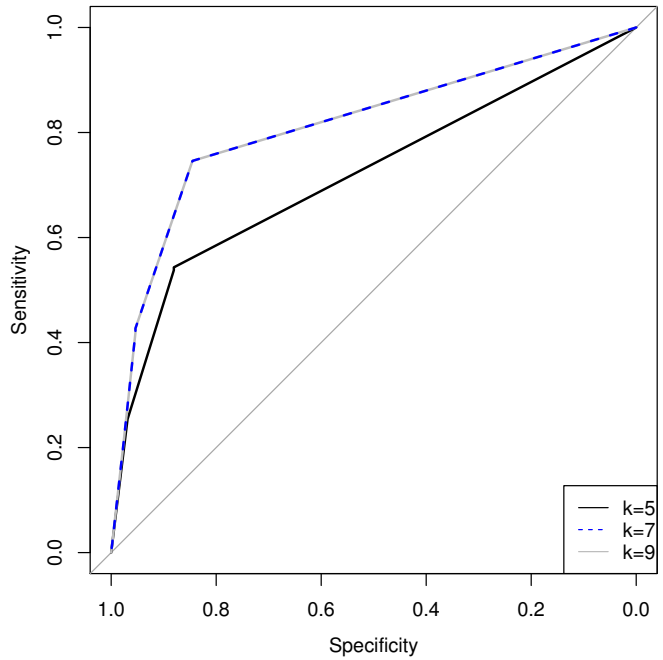
$$\hat{y} = arg \max_{k \in \{1,\cdots,K\}} \text{Card}\{\mathbf{x}_i \in NN(\mathbf{x}) \ \& \ y_i = k\}$$



knn classifier, example

- Leukemia genes expression
- Here, we keep the genes with highest absolute correlation with class AML/ALL ($|\rho| > .7$)
- Monte carlo cross validation, n.train = 25, 100 runs.
- For k=7 and k=9, AUROC = 0.81.





- AML is almost always well predicted. Errors occur for ALL class.
- By adding information, the dimension is increased (which may be a drawback for knn) but here : for $|\rho| > .5$, k=7 and k=9, AUROC = 0.88.

k-nearest neighbors

- kNN is easy to handle and has good performances if the learning data set is large.
- kNN approach may break down in high dimensions.
- Dimension reduction techniques can be combined with kNN.
- When the number of instances is large, the computational cost of finding the k nearest neighbors may be expensive: use specific algorithm like kd-trees, ...

Outline

Local methods in high dimension

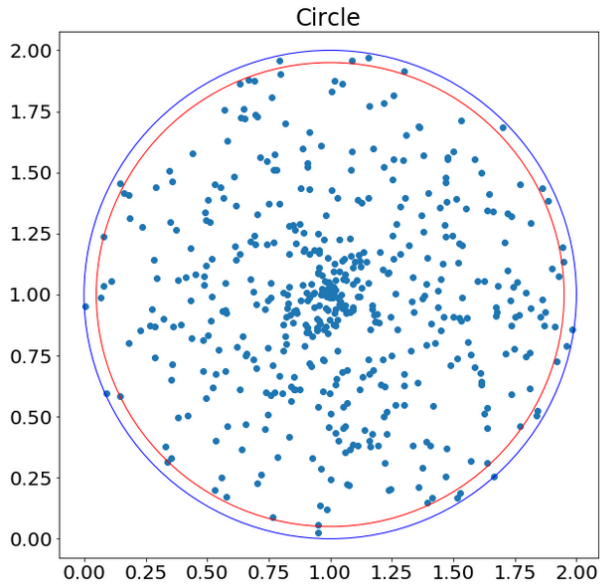
Curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces.

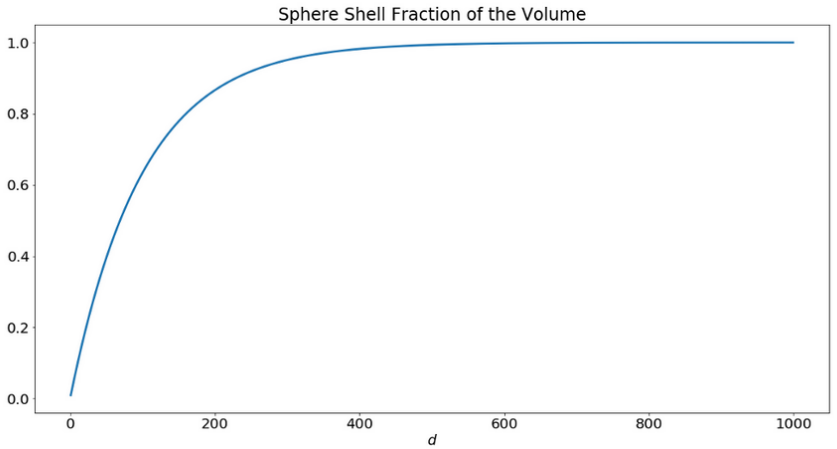
As the number of features or dimensions grows, the amount of data we need to generalize accuretely grows exponentially[1].

[1] Incomputer science, "exponentially" means "bad"...

Example: sphere in high dimension

Volume of a sphere in $\mathbb{R}^d$: $V = cr^d$.





Outside circle $r_o = 1$,
inner circle $r_i = .95$

$V = cr_o^d = c$,
$V_{\text{shell}} = c - cr_i^d$,

Sphere shell fraction of the volume:

$$\frac{V_{\text{shell}}}{V} = 1 - r_i^d$$

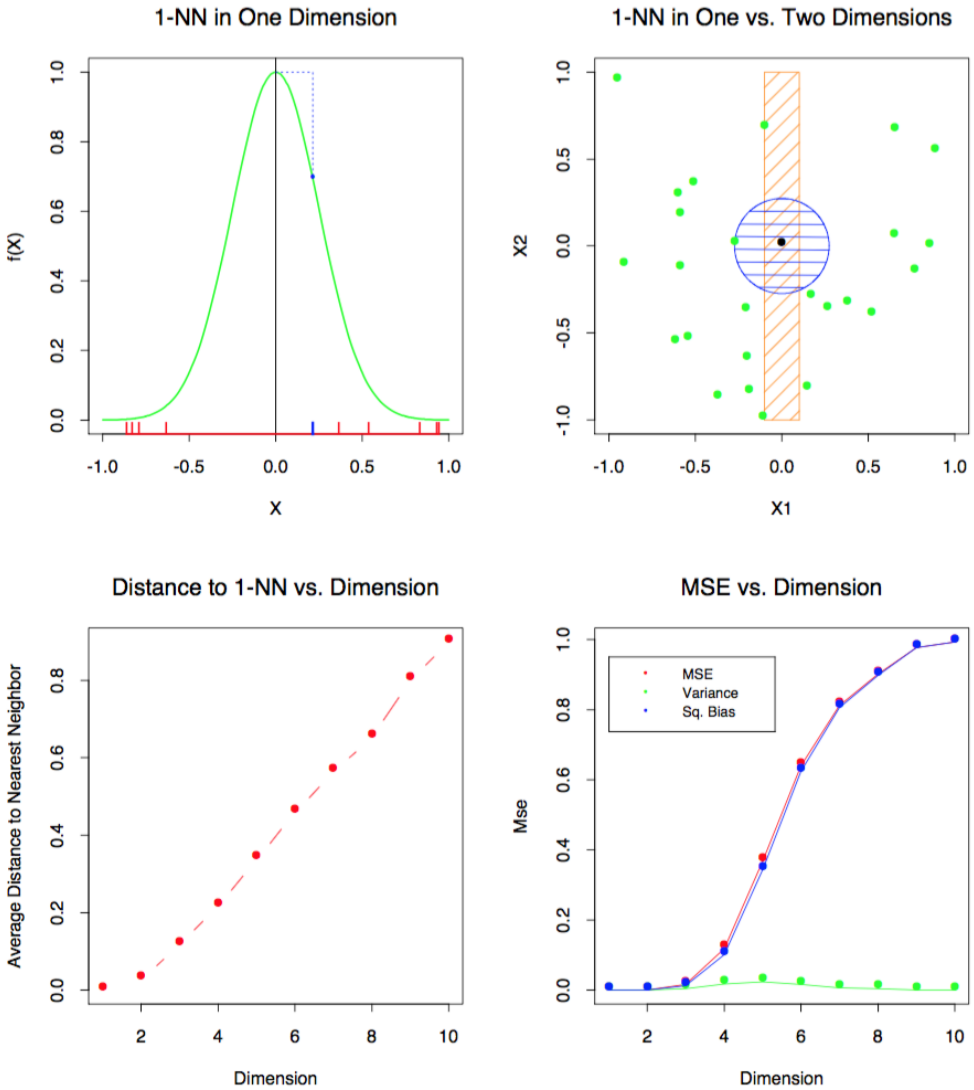
In high dimensional spaces most of the observed points belongs to the shell's fraction of the hypersphere's volume. The inner volume of the red shpere is almost empty...

It is hard to obtain a representative sample in this case !

Local methods in high dimension

Curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces.

- The input features are uniformly distributed in $[1,1]^p$ for $p = 1, \cdots, 10$.
- The top left panel shows the target function $f(X) = e^{||X||^2}$, and the error that 1-nearest neighbor makes in estimating $f(0)$.
- The top right panel illustrates why the radius of the 1-nearest neighborhood increases with dimension p.
- The lower left panel shows the average radius of the 1-nearest neighborhoods.
- The lower-right panel shows the MSE, squared bias and variance curves as a function of dimension p.

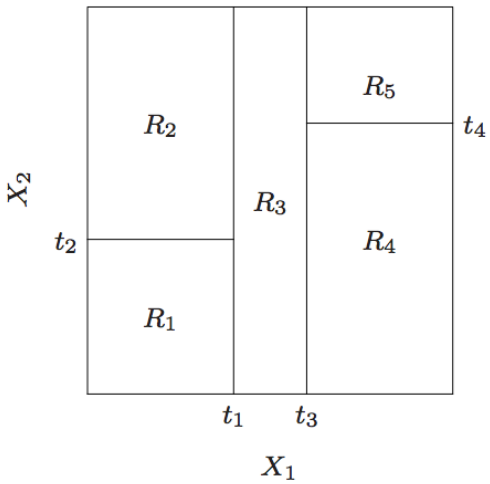


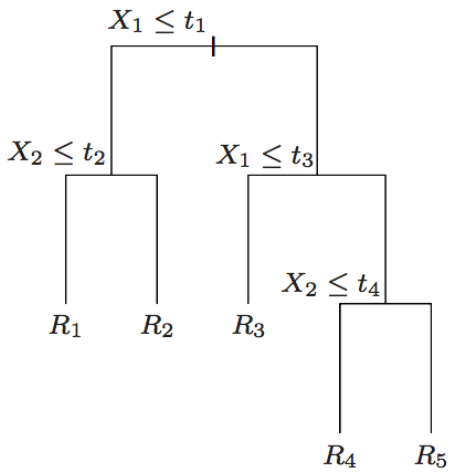
Outline

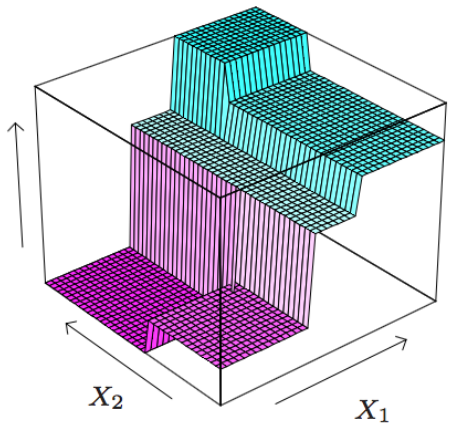
Introduction

- Tree-based methods partition the feature space into a set of rectangles, and then fit a simple model (like a constant) in each one.
- They are conceptually simple yet powerful.
- Let's consider a regression problem with continuous response $Y \in \mathcal{Y}$ and inputs X_1 and X_2, each taking values in the unit interval. In each cell, Y is predicted by the mean value c_m of the observations belonging to the cell R_m.

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{I}_{R_m}(x_1, x_2)$$







Figures from Hastie's book.

Construction of a regression tree: greedy algorithm

- Data: $\mathcal{S}_n = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ with $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})$
- The trees are build by an iterative algorithm.
- The algorithm automatically decides on the splitting variables and split points. Splitting rule: decrease the variance such that the weighted combination of the child node variance is smaller than the variance of the parent.
- Once the partition R_m is fixed, the regression function for an observation $\mathbf{x} \in \mathcal{X}$ is given by

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathbb{I}_{R_m}(\mathbf{x}), \quad c_m = \sum_{\{i | \mathbf{x}_i \in R_m\}} y_i$$

- Now finding the best binary partition in terms of minimum sum of squares. Evaluating $\sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2$ is generally computationally infeasible. Hence trees proceed with a greedy algorithm.

Construction of a regression tree

- Starting with all of the data, consider a splitting variable x_j and split point s, and define the pair of half-planes

$$R_1(j,s) = \{\mathbf{x}|x_j \leq s\} \text{ and } R_2(j,s) = \{\mathbf{x}|x_j > s\}$$

- Then we seek the splitting variable x_j and split point s that solve

$$\min_{j,s}\left(\min_{c_1}\sum_{\{i|\mathbf{x}_i\in R_1(j,s)\}}(y_i-c_1)^2+\min_{c_2}\sum_{\{i|\mathbf{x}_i\in R_2(j,s)\}}(y_i-c_2)^2\right).$$

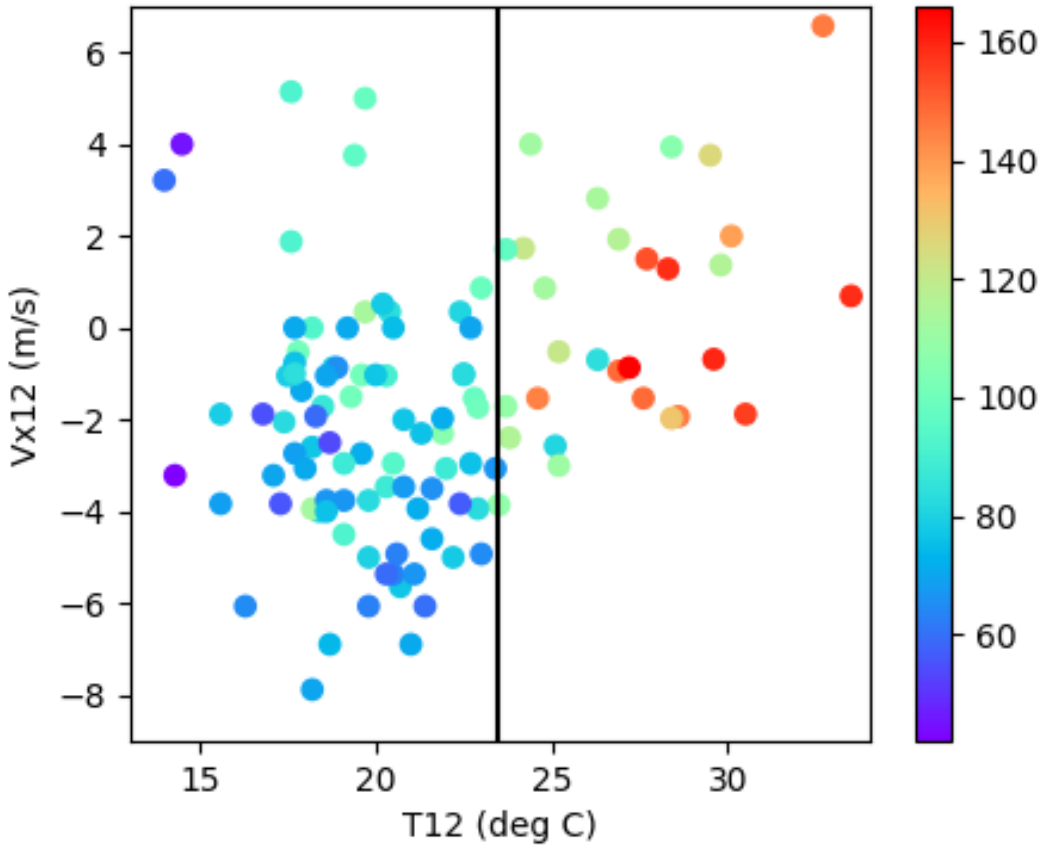
  where for any choice x_j and s, the inner minimization is solved by

$$\hat{c}_1 = \sum_{\{i|\mathbf{x}_i\in R_1\}} y_i \text{ and } \hat{c}_2 = \sum_{\{i|\mathbf{x}_i\in R_2\}} y_i$$

- For each splitting variable, the determination of the split point s can be done very quickly and hence by scanning through all of the inputs, determination of the best pair (j,s) is feasible.

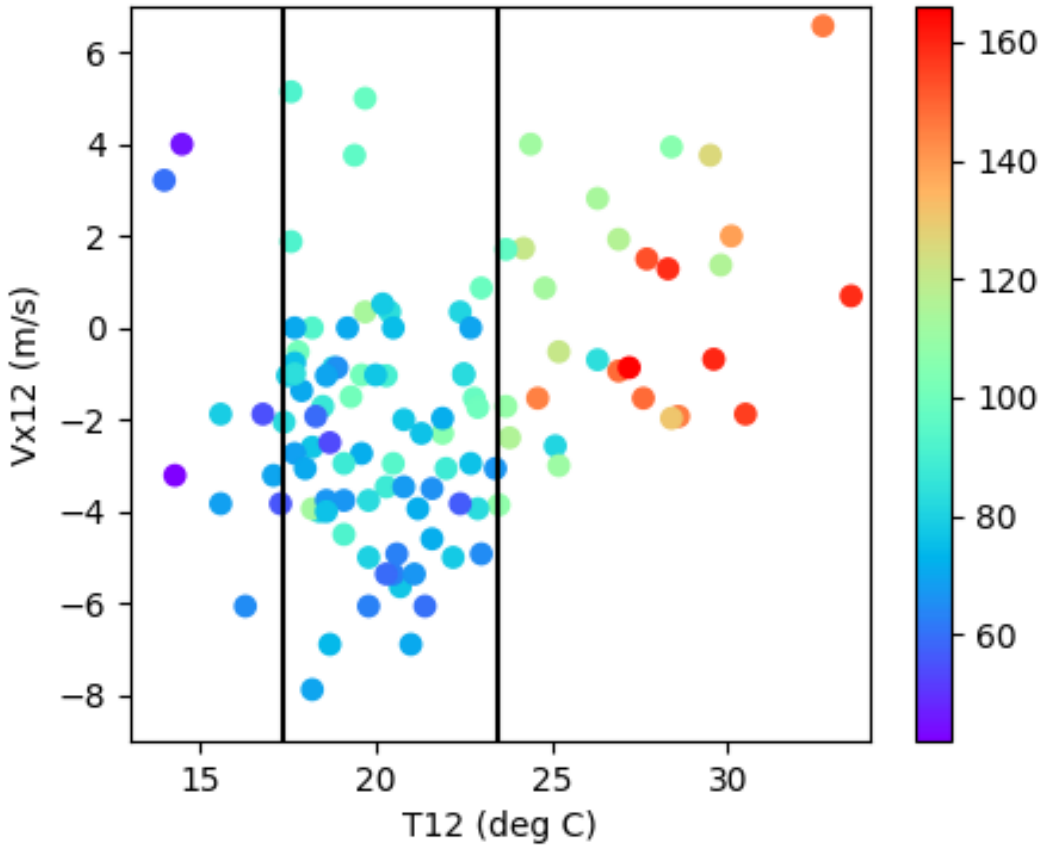
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



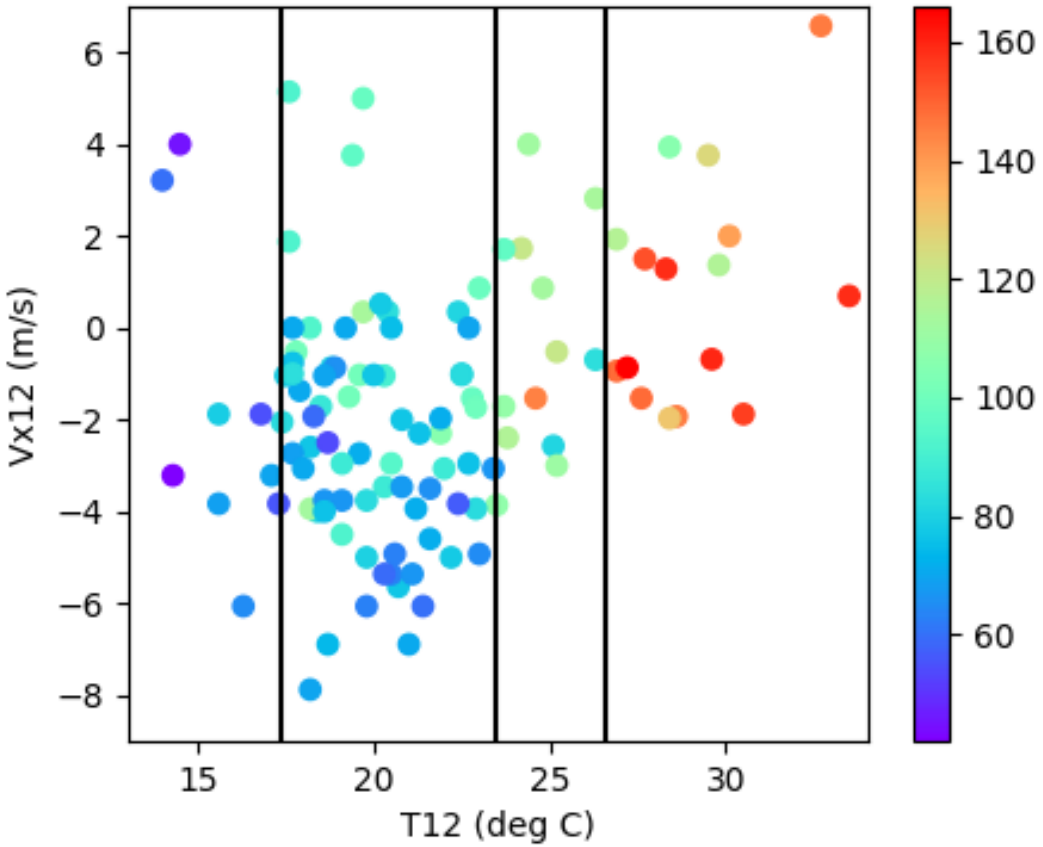
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



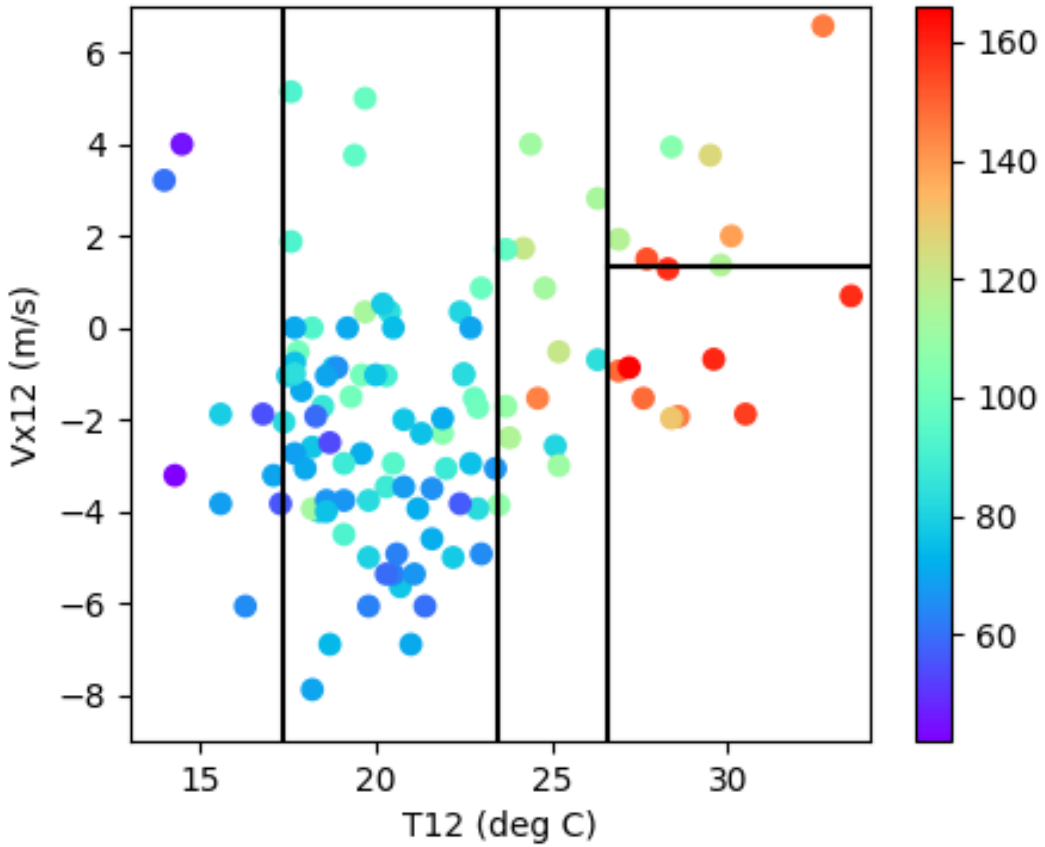
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



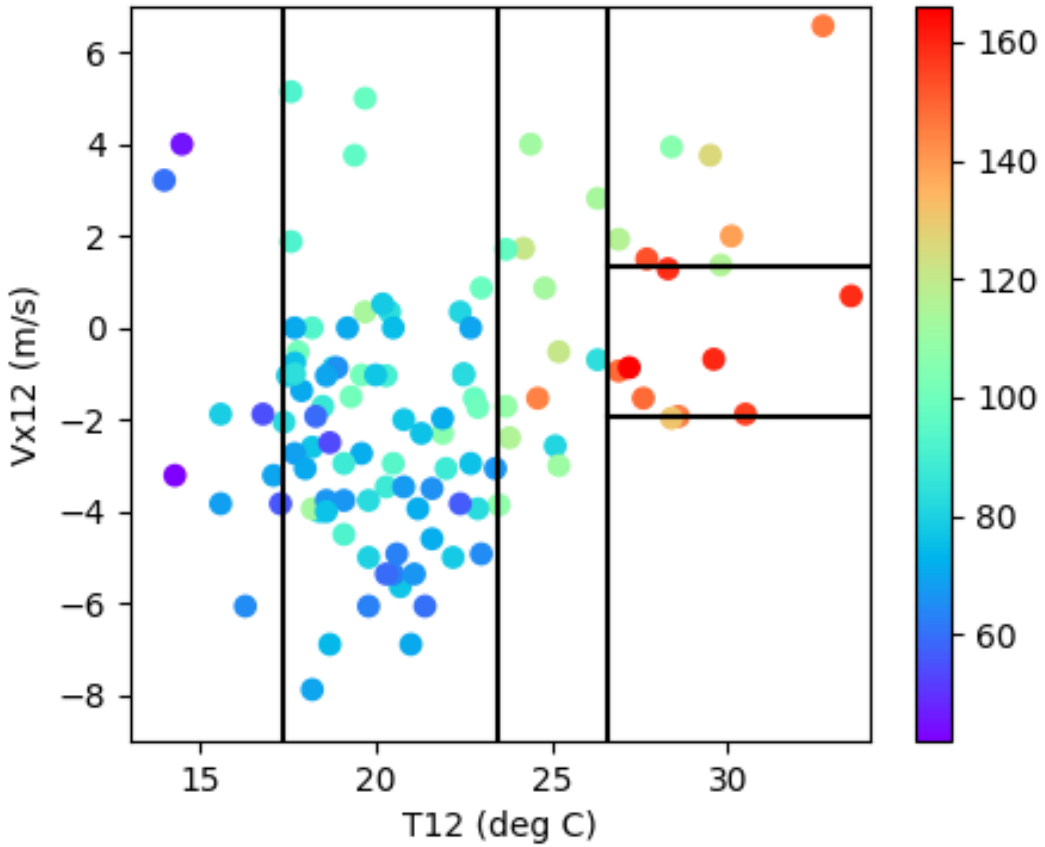
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



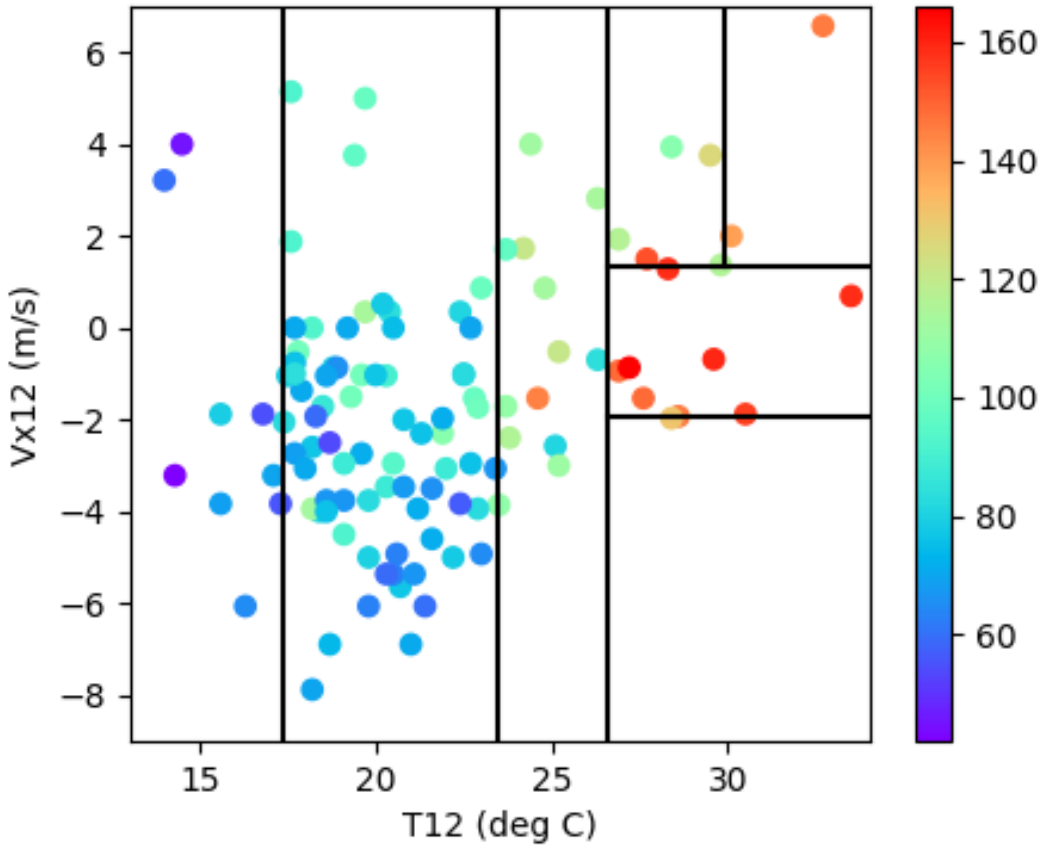
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



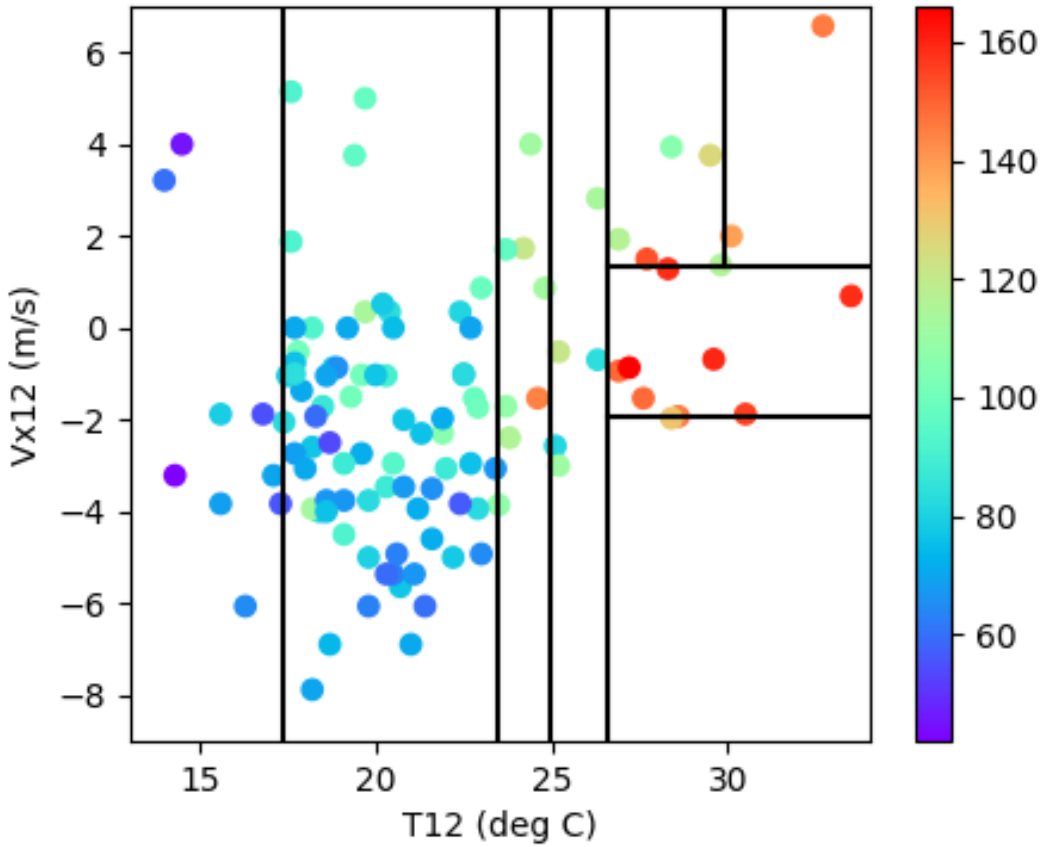
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



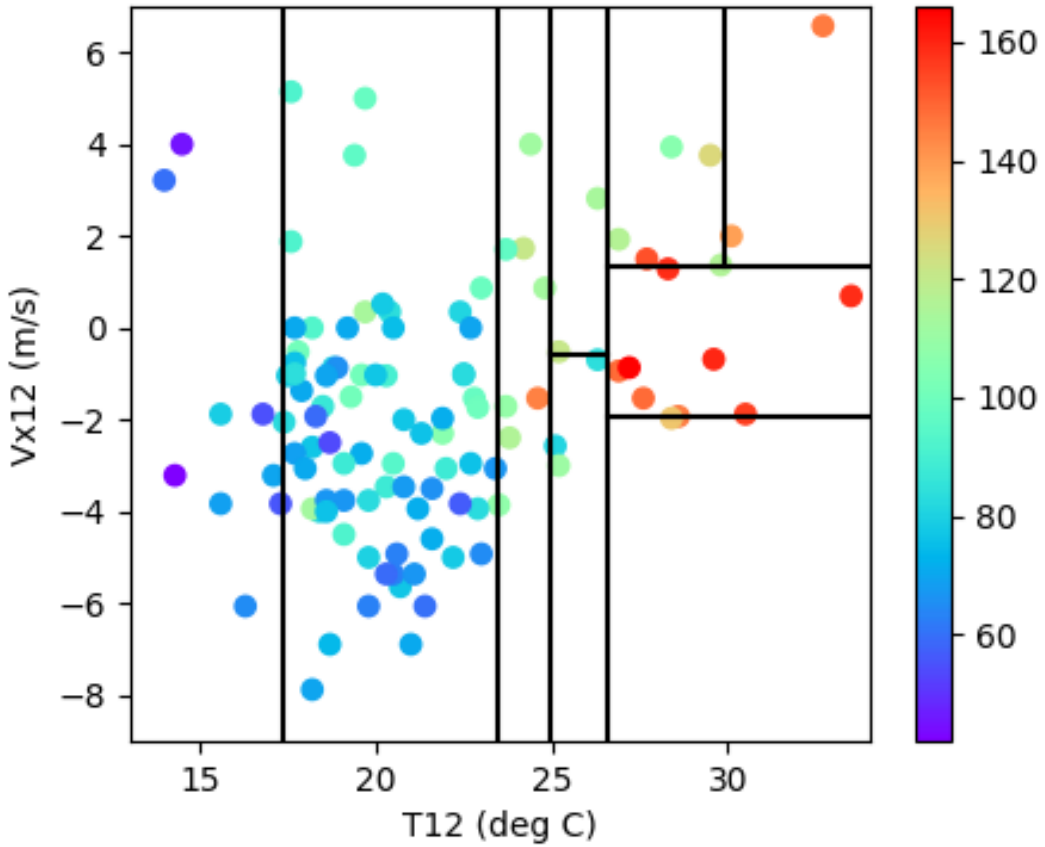
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



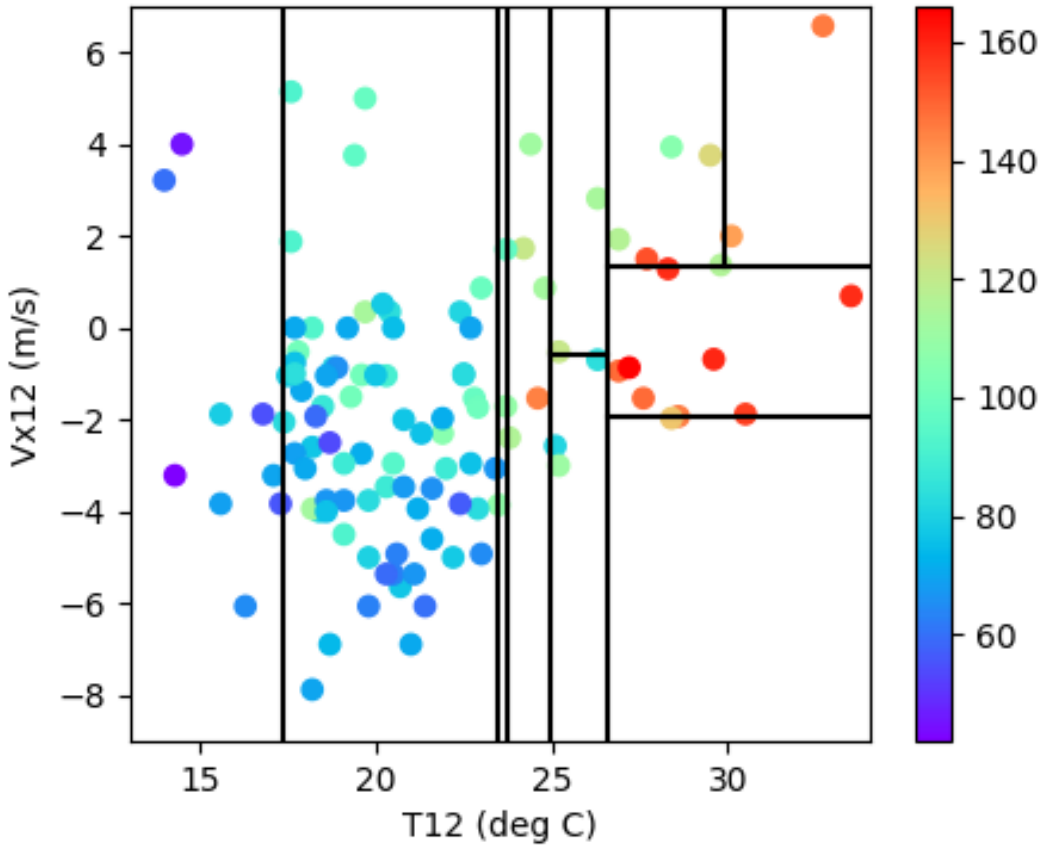
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



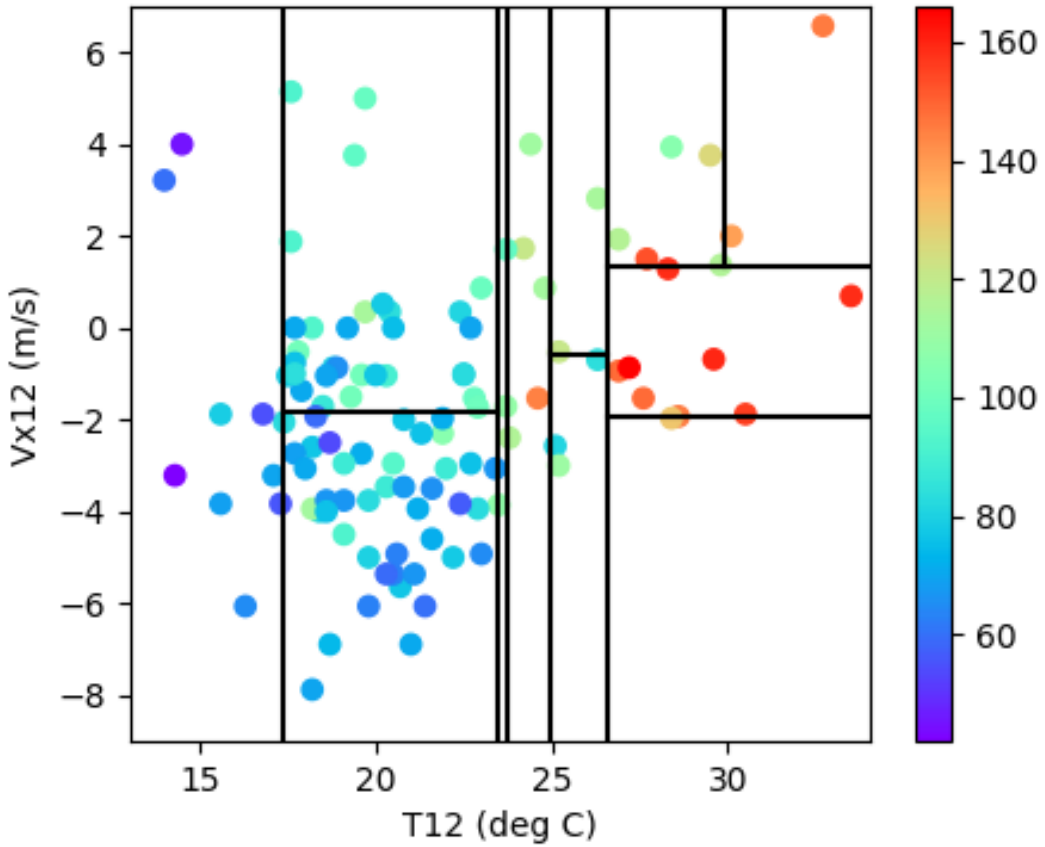
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



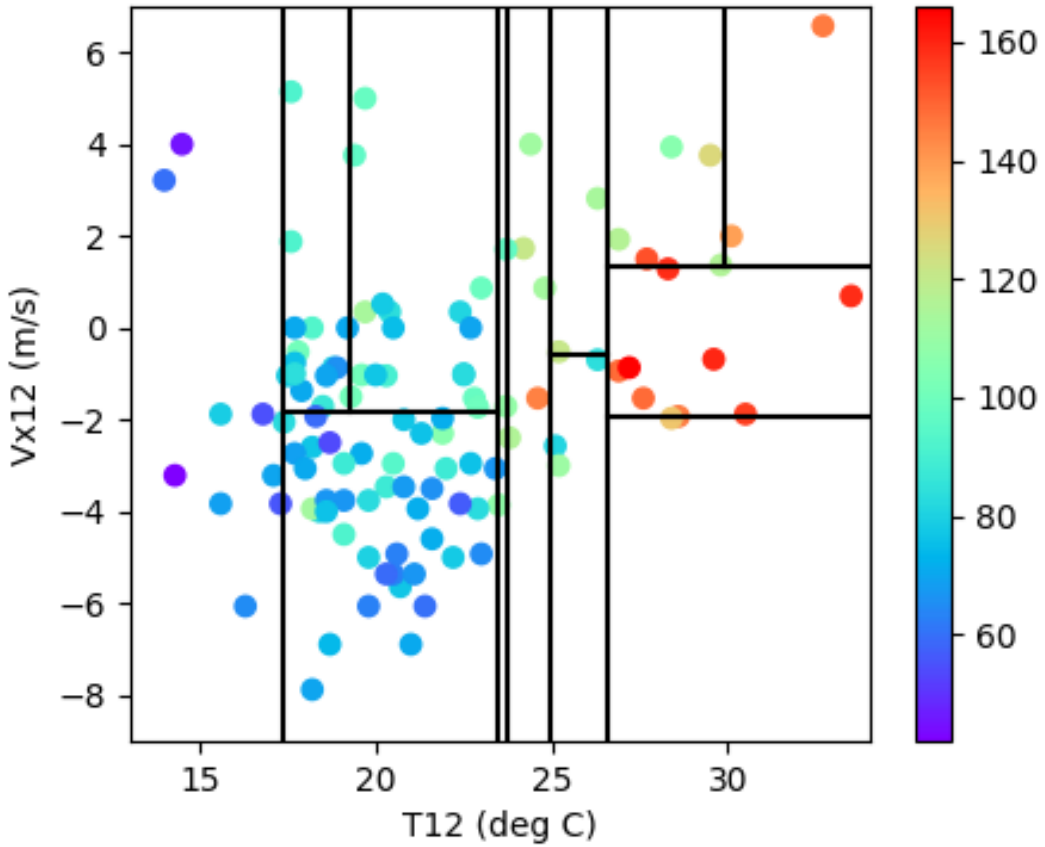
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



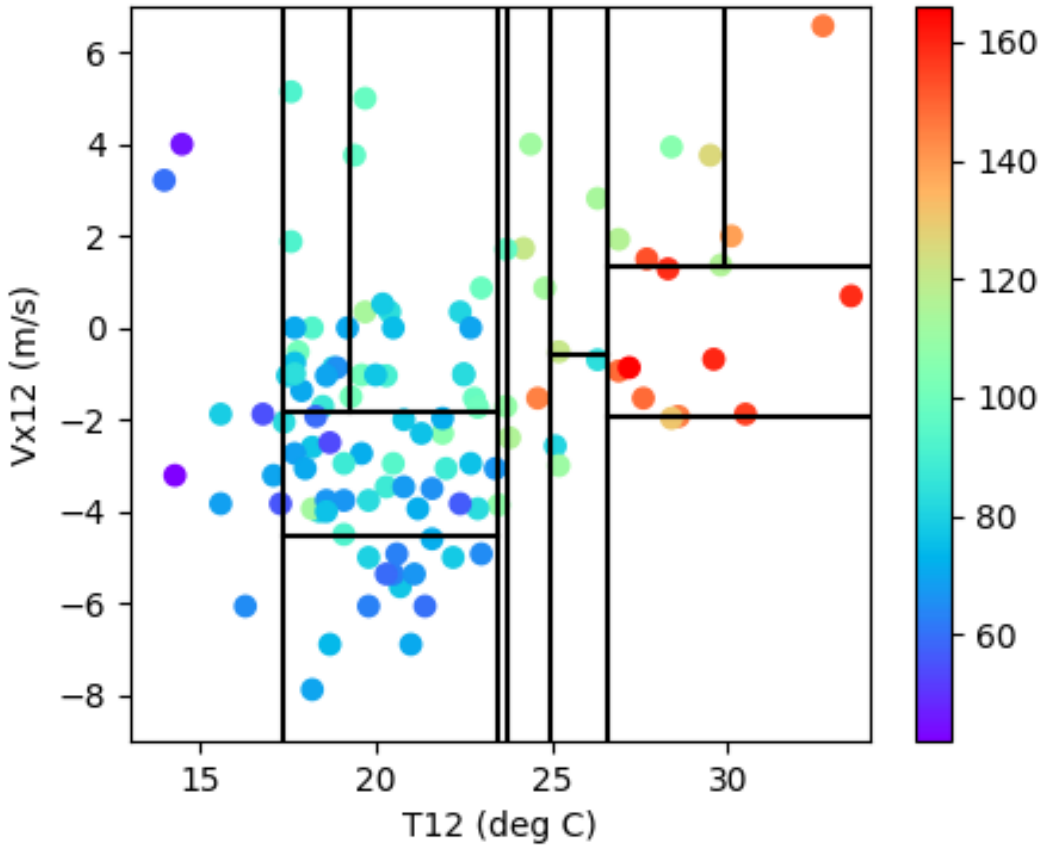
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



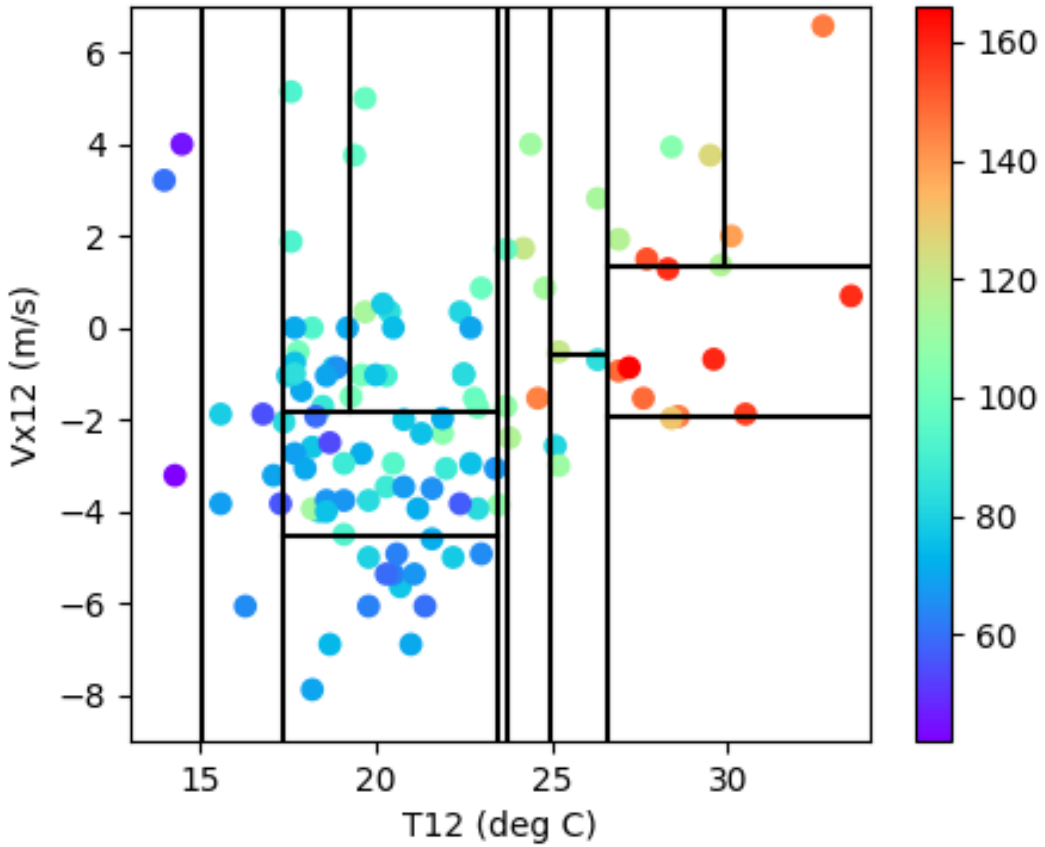
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



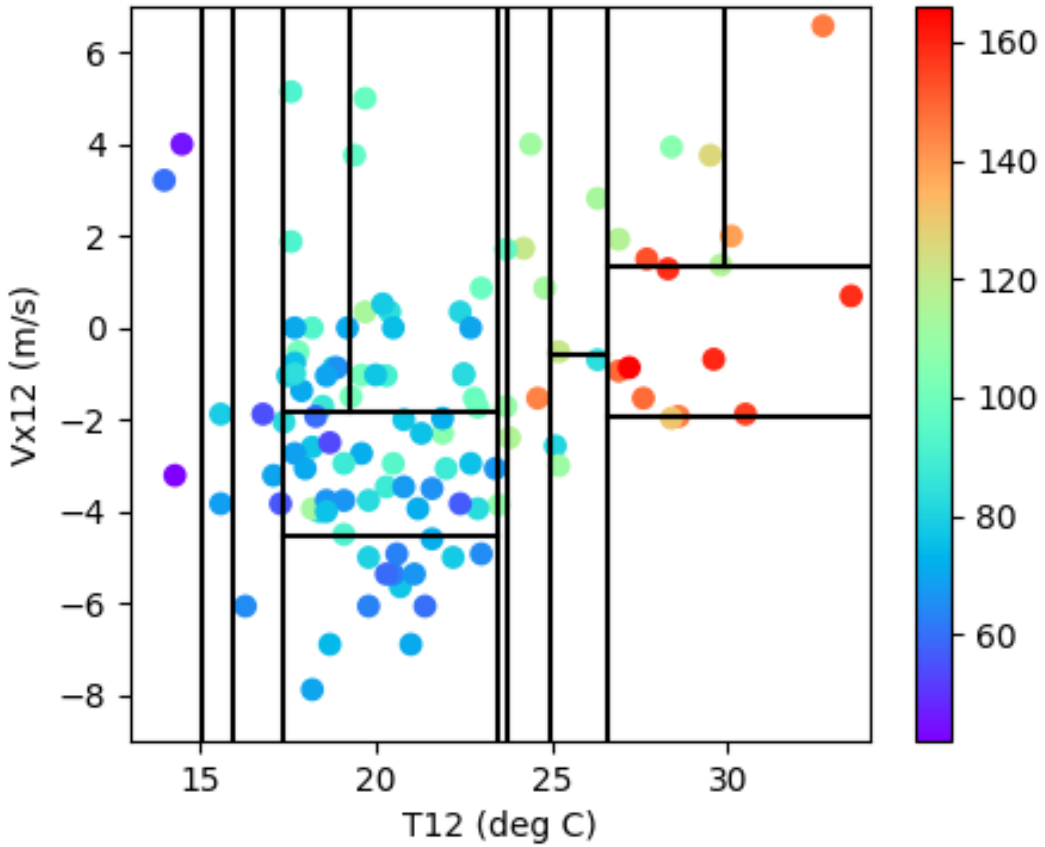
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



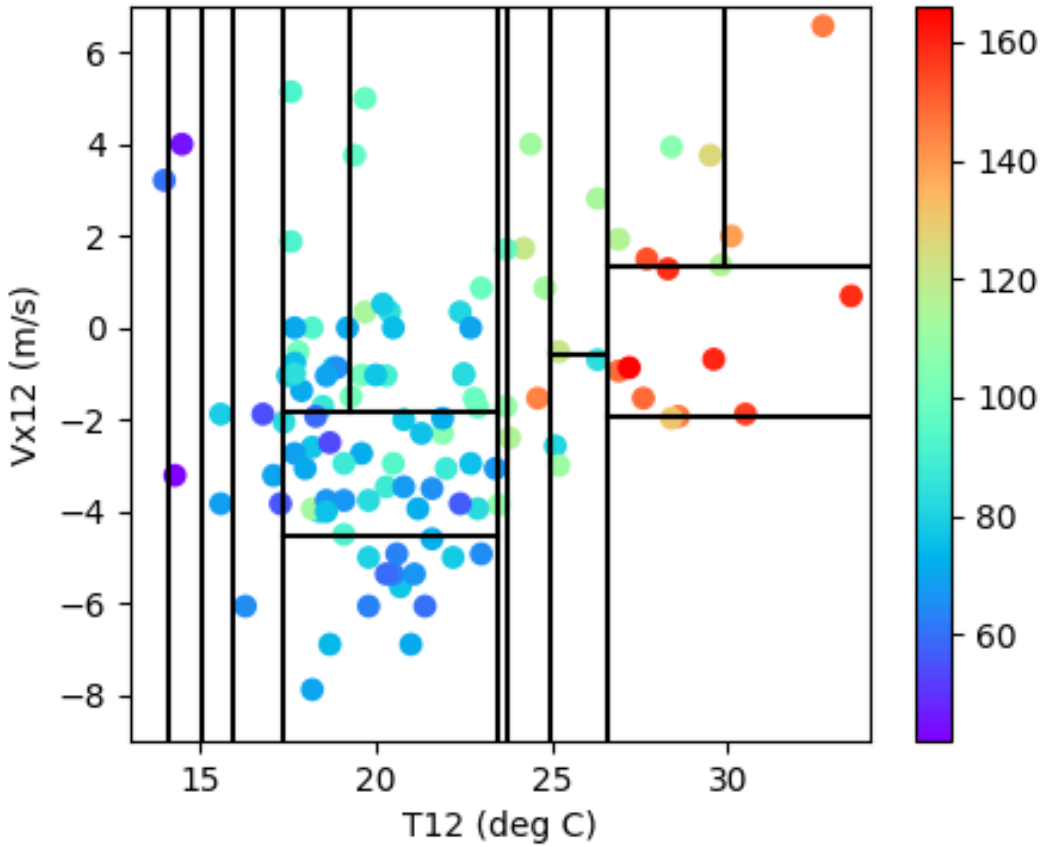
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



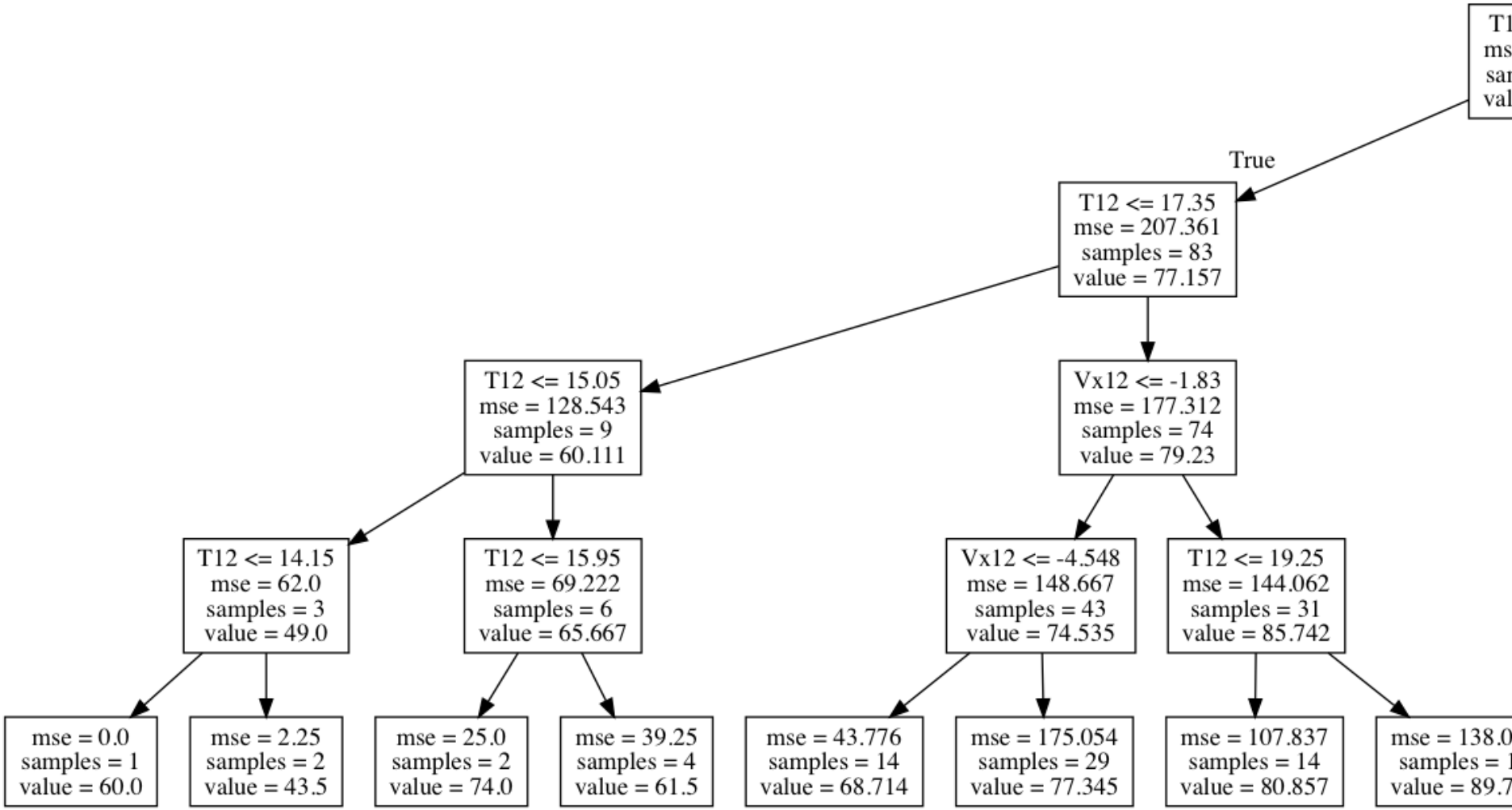
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



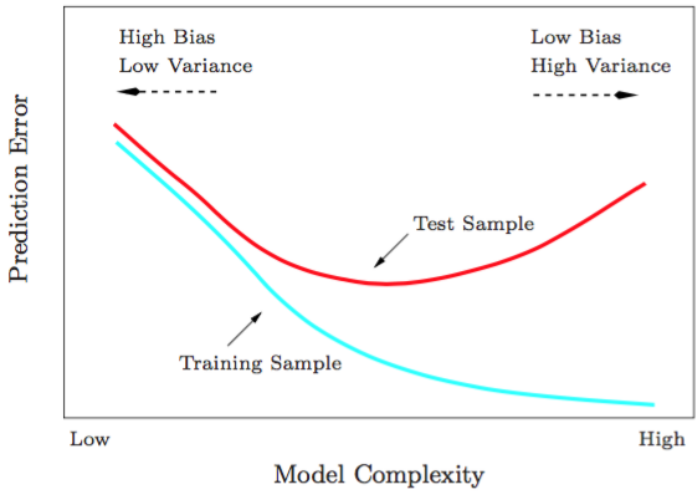
Construction of a regression tree, Ozone

- maxO3 with respect to T12 and Vx12



How large should we grow the tree?

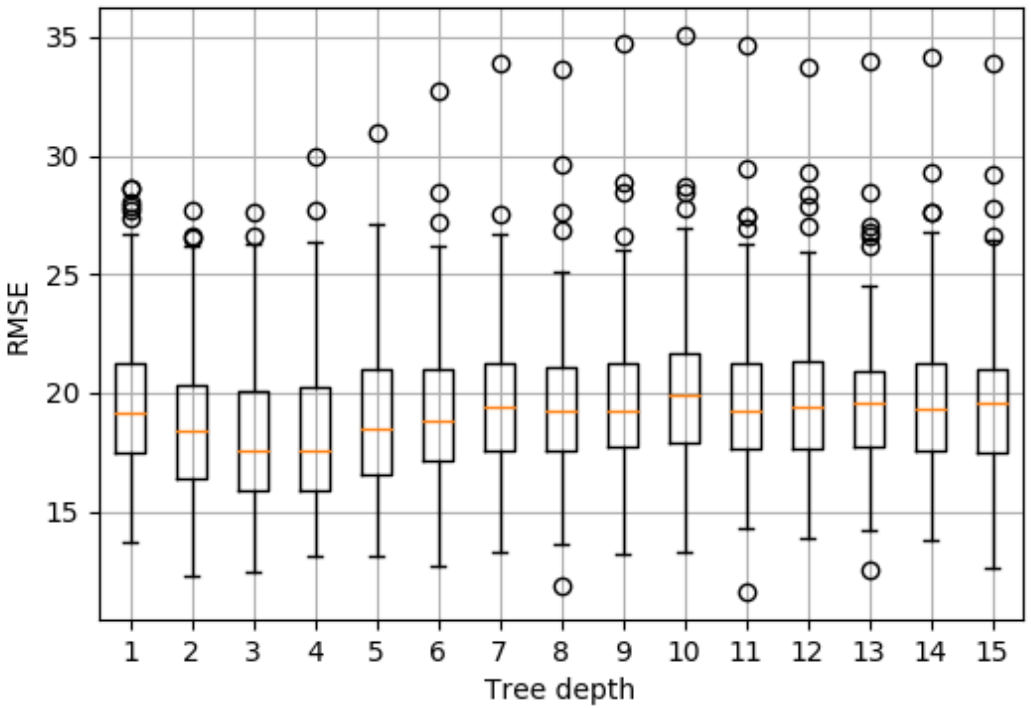
- Bias-Variance Tradeoff: Very large tree might overfit the data while a small tree might not capture the important structures.



- The depth of the tree drive the compromise between bias and variance
  -Low depth (= few splits) leads to stable trees that have low variance but high bias
  -High depth (= many splits) leads to stable trees that have low bias but high variance

Example: Ozone

- Tree depth selection (Monte Carlo cross validation)



How large should we grow the tree?

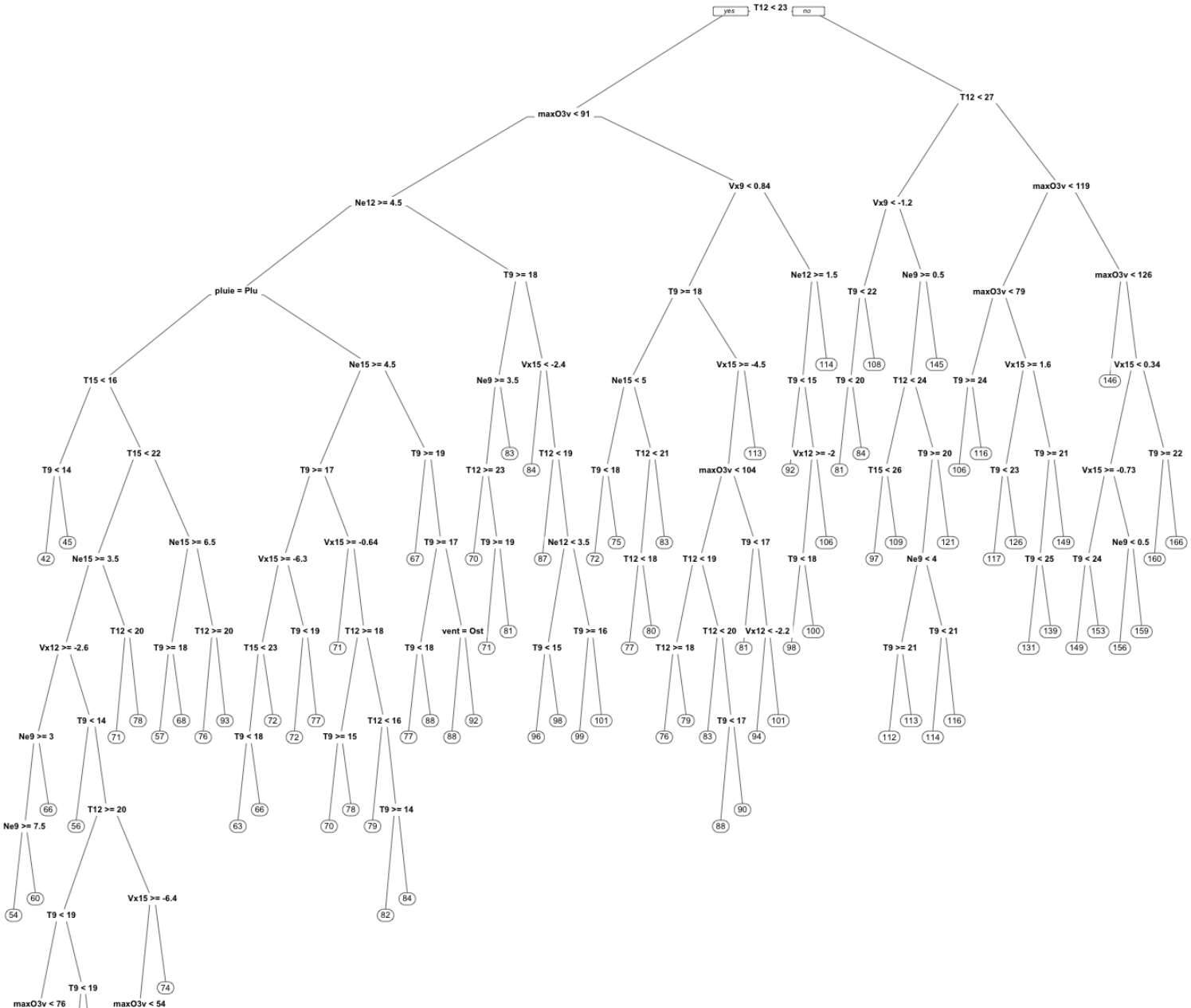
- The strategy is to grow a large tree $\mathbb{T}_0$, stopping the splitting process only when some minimum node size (say 5) is reached. Then this large tree is pruned using cost-complexity pruning.
- The cost-complexity criterion is defined by

$$C_\alpha(\mathbb{T}) = \underbrace{\sum_{m}^{|\mathbb{T}|} N_m Q_m(\mathbb{T})}_{\text{cost}} + \underbrace{\alpha|\mathbb{T}|}_{\text{complexity}}$$

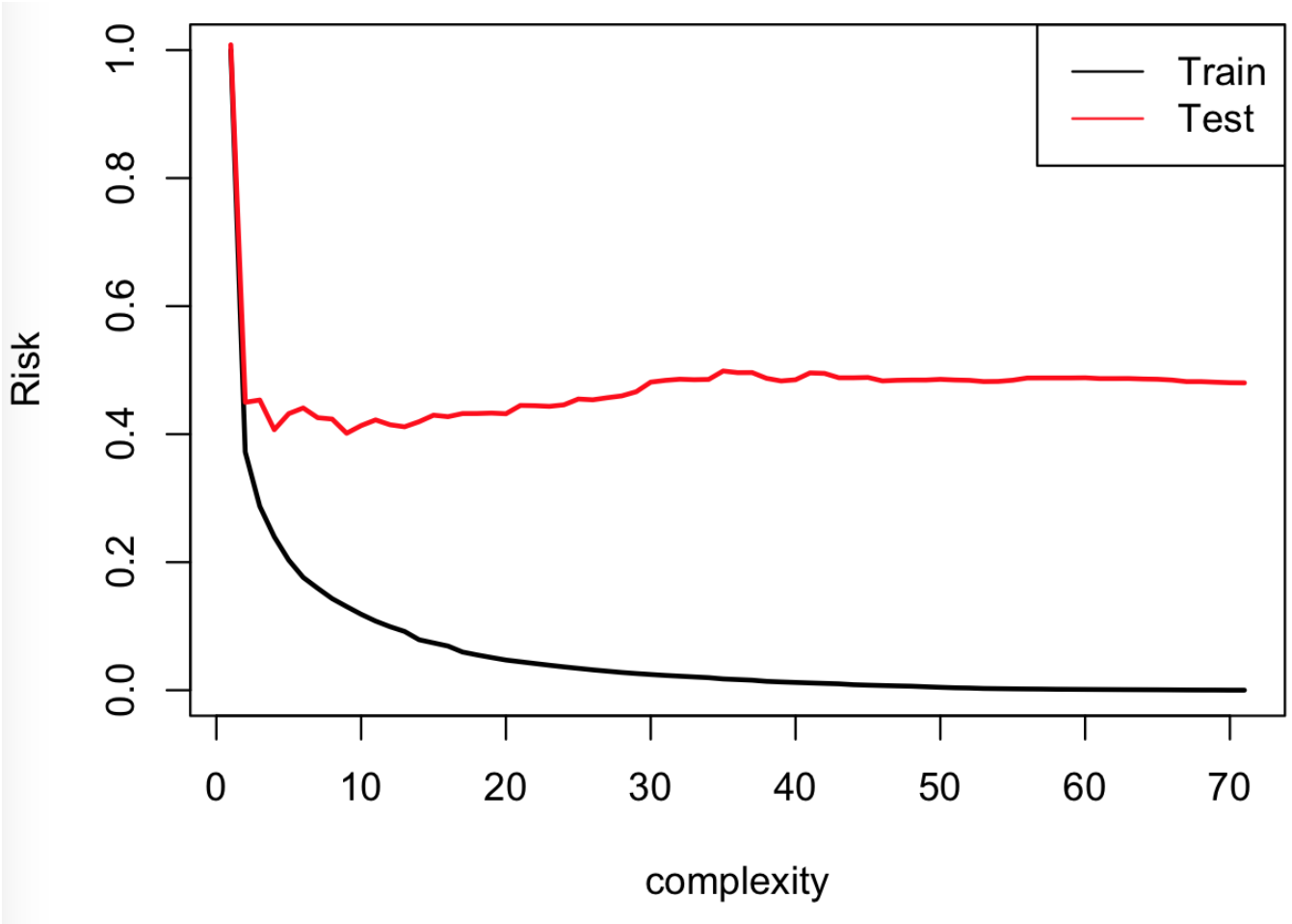
where $N_m = \text{Card}(\{\mathbf{x}_i \in R_m\})$ and $Q_m(T) = \frac{1}{N_m} \sum_{\{i|\mathbf{x}_i \in R_m\}} (y_i - \hat{c}_m)^2$.

- The idea is to find, for each α, the subtree $\mathbb{T}_\alpha$ of $\mathbb{T}$ to minimize $C_\alpha(\mathbb{T})$.
- α is referred to as complexity parameter.
- Estimation of α is achieved by 5- or 10-fold cross-validation.

Maximal tree, Ozone

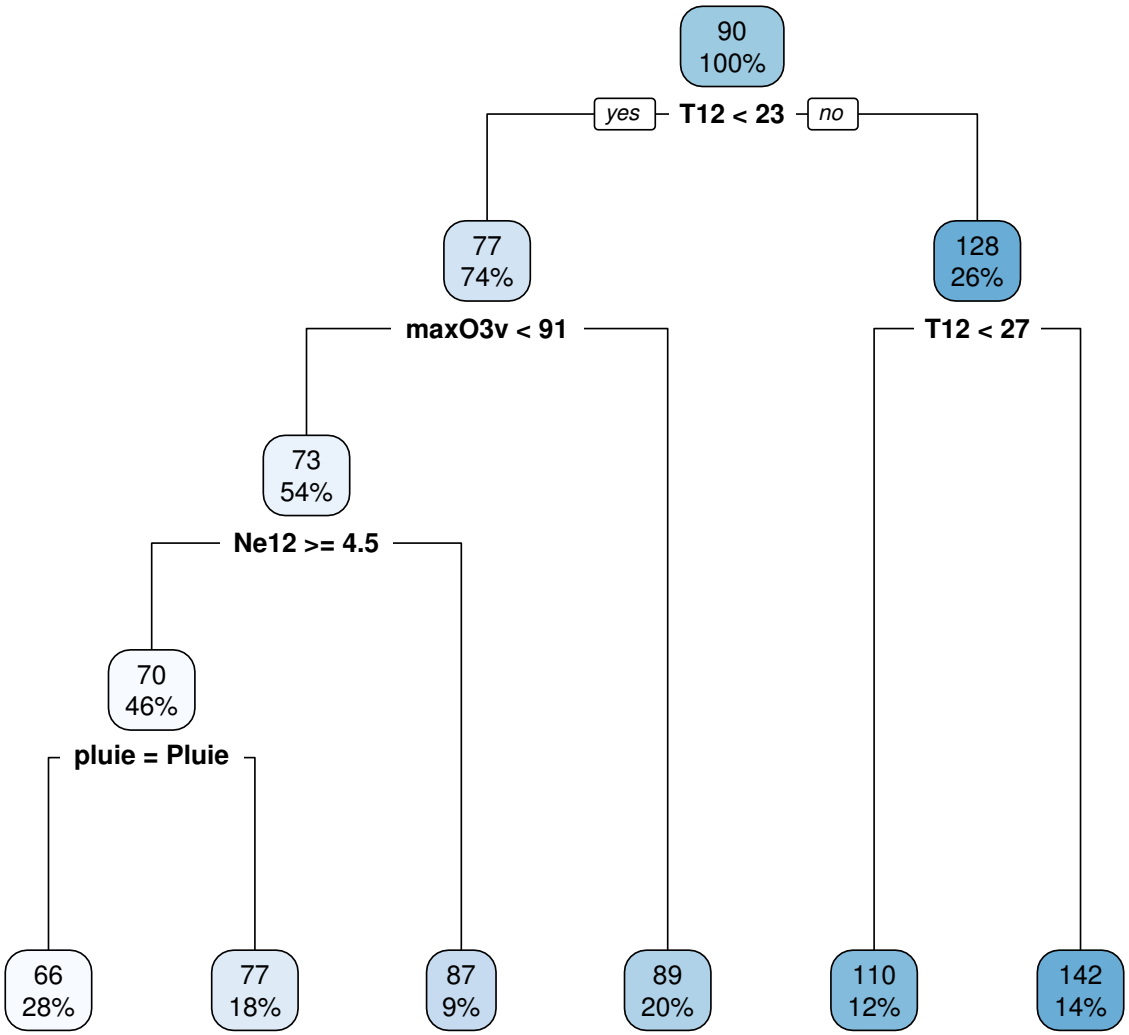


Complexity/risk compromise, Ozone



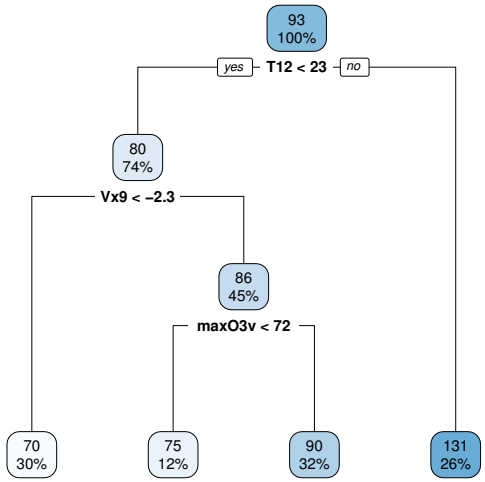
Complexity/risk compromise, Ozone

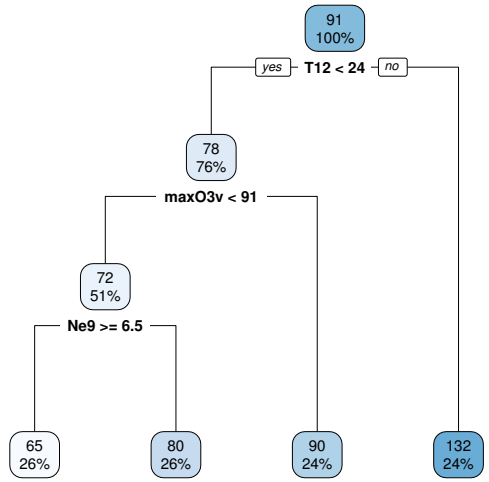
- Complexity/risk compromise leads to the following tree.



Tree instability, Ozone

- Tree instability: trees fit on two different learning sets are different.





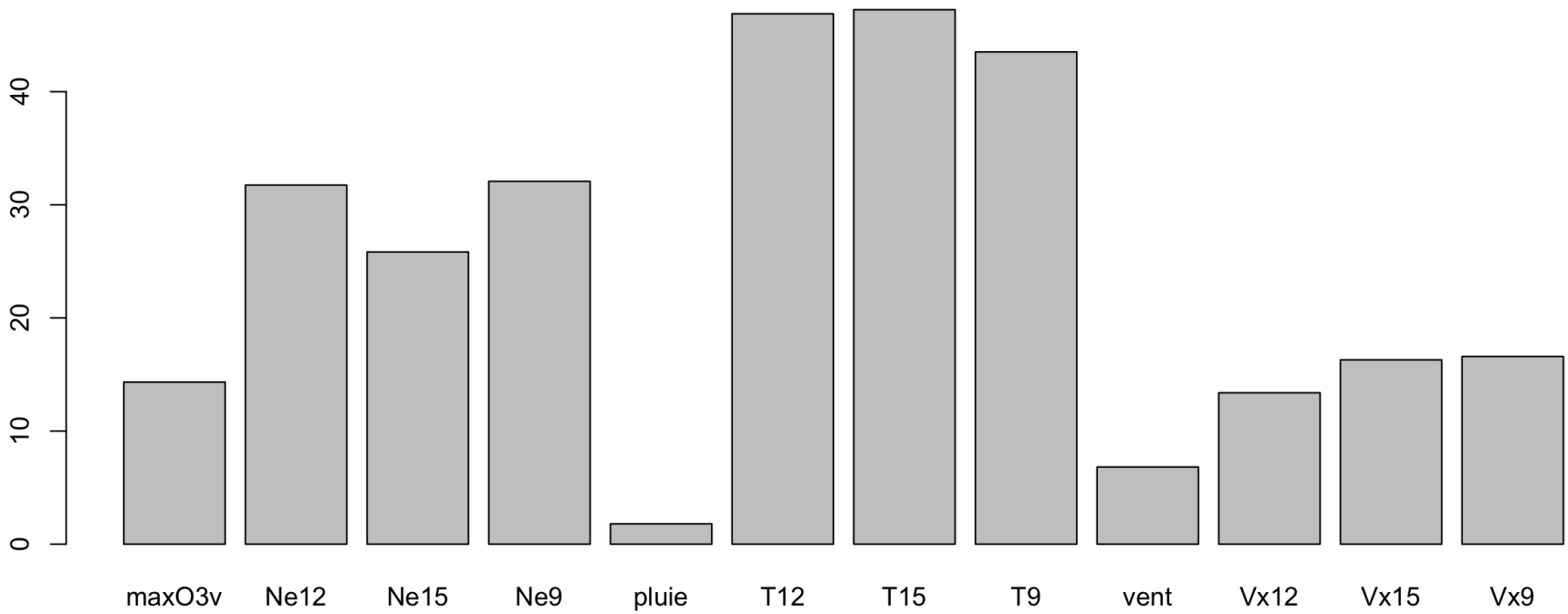
Variable importance

- A question that frequently arises among tree users is: which variables are the most important?
- To measure the importance of a variable X_j, the idea is to compute, for each split of the tree, the deviance gain (improvement) obtained if the optimal split is replaced by a split based on X_j.
- In a regression tree, the improvement of a split is measured by

$$i^2(R_\ell, R_r) = \frac{w_\ell w_r}{w_\ell + w_r}(\bar{y}_\ell - \bar{y}_r)^2$$

  where ℓ (resp. r) referred to as a left (resp. right) node.
- In practice the computation of variable importance is not expensive since it just consists in storing the results of the calculus which are done to build the tree.

Variable importance, Ozone



Classification trees

- If $\mathcal{Y} = \{1, \cdots, K\}$ i.e. Y is a categorical variable, the trees algorithms are very similar.
- The main differences are that,
  - for an instance $x \in \mathcal{X}$, $\hat{y} = arg\max_{k \in \mathcal{Y}} \hat{p}_{mk}$ where

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{\text{ß} | \mathbf{x}_i \in R_m} \mathbb{I}(y_i = k)$$

  - the quality of the tree is measured by one of the impurity criteria

Misclassification error: $\frac{1}{N_m} \sum_{\text{ß} | \mathbf{x}_i \in R_m} \mathbb{I}(y_i = k) = 1 - \hat{p}_{mk}$

Gini index[2]: $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$

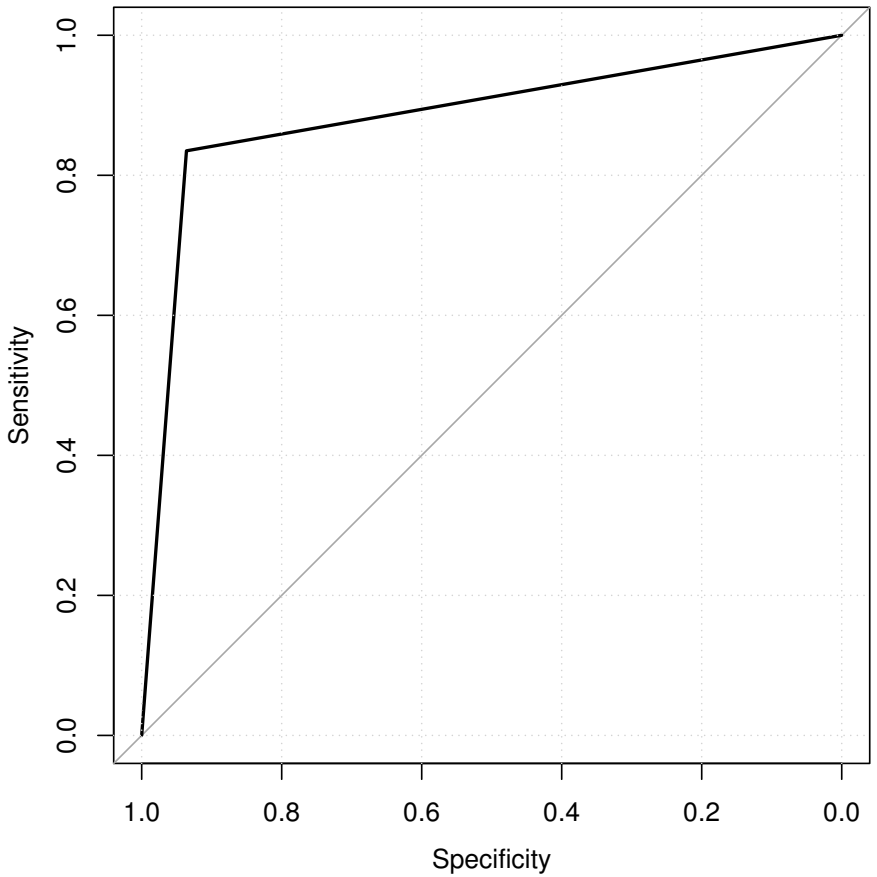
Entropy[3]: $-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

[2] Gini index is a measure of inegalities. It is equal to 0 for a total equality and to 1 if the situation is totally unequal. ex: income inequalities.

[3] Entropy can be interpreted as a measure of disorder.

Classification trees, Leukemia

- Monte Carlo cross validation, 100 runs.
- AUC = 0.902 is quite high for a regression tree.



Trees

- Decision trees are easy to use and they lead to interpretable representations.
- They easily deal with missing values because algorithms work variable by variable.
- Instability
  One major problem with trees is their high variance. Often a small change in the data can result in a very different series of splits.
- Lack of smoothness
  Trees lead to constant by pieces boundaries.
- Random forests (combination of trees) provide an interesting alternative to address the two last points (see below).

Concluding remarks

- Data driven model such as kNN and Trees lead to prediction models which are locally optimal.
  They require quite a lot of data to be learned.
  They are affected by curse of dimensionality.
  They are might not be convenient for extrapolation.

  But they are powerfull is enough data are available and the relationship betwenn X and Y is complex!

- Linear models such as linear regression, logistic regression, discriminant analysis are globally optimal.
  If only few data are available, they should be prefered.