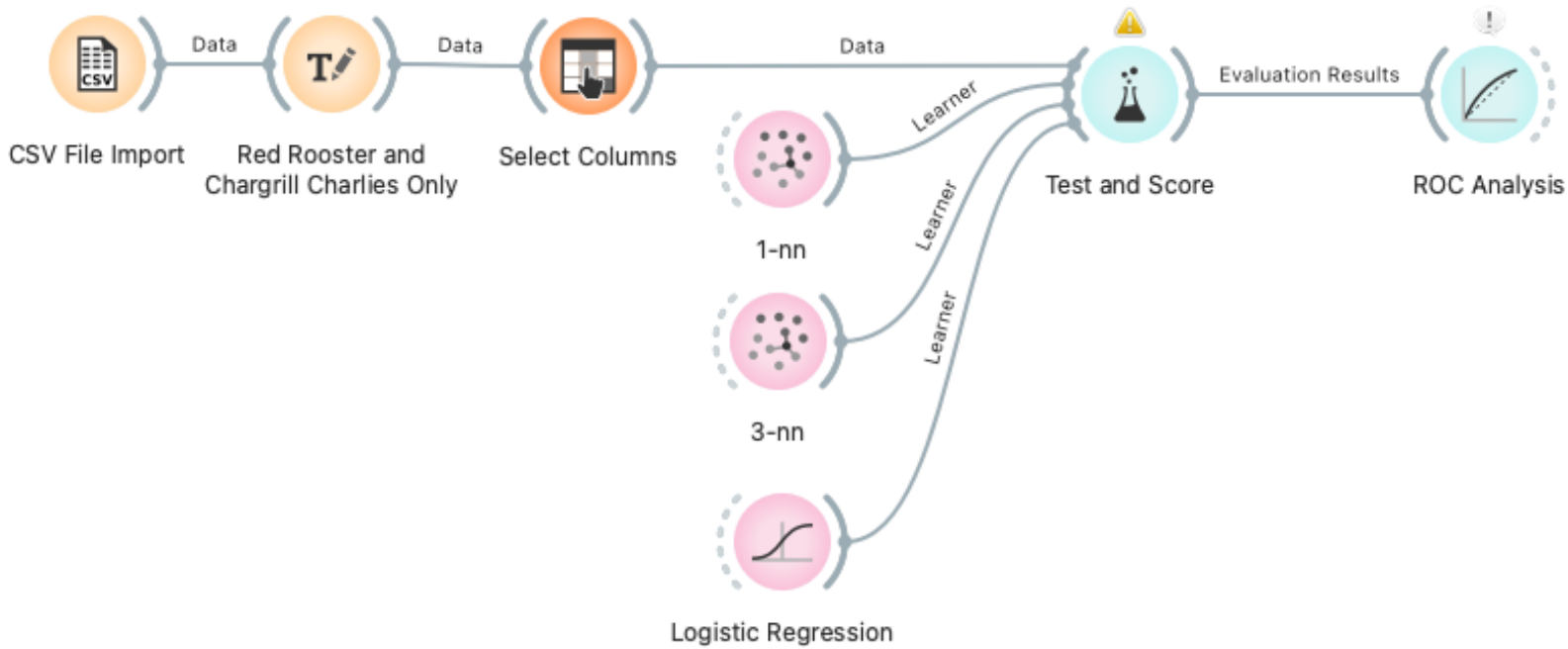
Week 6 — Trees, Rules (CN2), Ensembles, AUC & Explainability in Orange

Greg Baker

1st September 2025

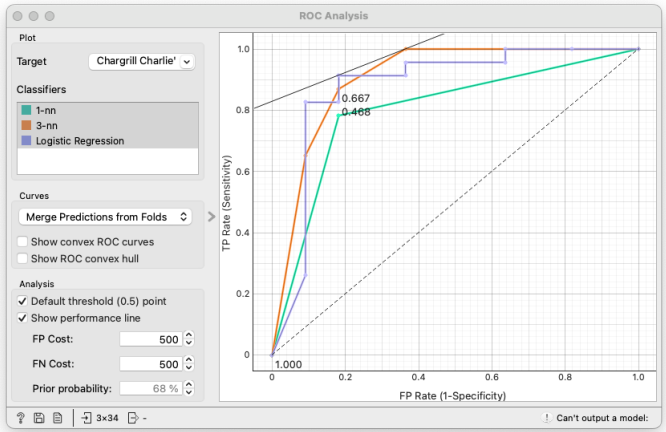
ROC / AUC

Reviewing last week, I didn't do a great job explaining the ROC ("receiver operating characteristic") curve.
Aka "area under curve"



AUC, ROC, and thresholds (the quick, correct story)

- $\mathrm{TPR} = \dfrac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$,
  $\mathrm{FPR} = \dfrac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$.
- ROC curve: plot TPR vs FPR as you sweep the decision threshold.
- AUC is the probability a random positive scores higher than a random negative.
- **Key point:** AUC ignores calibration; it measures *ranking* quality.
- Threshold choice is application-specific (tradeoffs), not baked into AUC.



Where we are & where we're going

- Last week: geospatial classification ("Red Rooster line") with Logistic Regression vs k-NN; plus a taste of AUC.

- Today:
  1. Decision Trees & CN2 (rules).
  2. Proper evaluation: **AUC/ROC**, thresholds, calibration.
  3. Ensembles: **Random Forests** and **Stacking**.
  4. Quick hyperparameter tuning in Orange: k for k-NN, #trees for RF.

Two reasons that data science projects happen

Inference We want to make automatic predictions, as accurately as possible. (You end up writing programs, often in Python)

Explainability We want to understand *why*. What features are discriminative and what effect do they have? (Nobody cares how you did it)

The very first question on any project: which goal are you aiming for?

Sometimes you don't have a choice

- **GDPR** General Data Protection Regulation, a comprehensive data privacy regulation in the European Union. AI decisions with legal or significant impact must be explainable, ensuring transparency and accountability.
  - **Extra-territorality** Applies to organizations processing EU citizens' data, *even if your company is in Australia*
- **Similar laws** Legal frameworks in other countries also emphasize AI explainability with extra-territorality:
  - **China** Draft Data Security Law and Personal Information Protection Law include provisions to protect citizens from unfair or opaque AI decisions.

Examples

- Software that scans resumes that filters out unqualified candidates.
- A Turn-it-in competitor that fails students for cheating
- Car insurance claim assessment that analyses the client's version of the accident

All of these have a *legal effect* (cause a contract to go ahead / be rejected): therefore they all have an explainability requirement.

The dearth of good sources on this

- I can't find any good readings on GDPR and PIPL.
- Lots of random web pages, some of which are somewhat trustworthy.
- Some very dull journal articles

What you need to know: *GDPR and PIPL require explainability if the model is used for something important that might affect someone's life.*

Sometimes you do have a choice

There are advantages to explainable models.

- Easier to debug
- Can be a unique selling point or product differentiator
- Being ethical is a good idea

Cynthia Rudin: Explainable is better than extra accuracy

- **Cynthia Rudin:** Professor of Computer Science, Electrical and Computer Engineering, and Statistical Science at Duke University.
- **Research focus:** Interpretable machine learning, AI fairness, and robustness.
- **Explainable AI:** Developing models that provide clear explanations for their decisions, fostering trust and transparency.
- **Accountability:** Enables humans to understand and verify model decisions, ensuring responsible AI deployment.
- **Regulatory compliance:** Helps organizations meet legal requirements for explainability and transparency.
- **Improved decision-making:** Facilitates better collaboration between humans and AI, enhancing overall performance.

Is Professor Rudin correct?

One of the most important questions in AI today.

Co-12 Explanation Quality Properties for Evaluating Explainable AI

Correctness
Match between model and explanation.

Completeness
How much of the model is explained?

Consistency
Robustness to small changes in model and implementation.

$g(x) = g(x)$

Continuity
Robustness to small changes input.

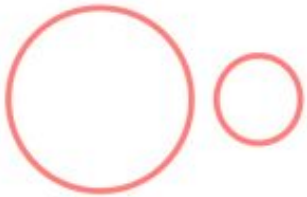
$g(x) = g(x')$

Contrastivity
Discriminative to other events or targets?

$g(x|Cat) != g(x|Dog)$

Covariate Complexity
Complexity of features in the explanation

Compactness
Size of the explanation



Composition
Presentation format

Confidence
Probability information available?

$p = ?$

Context
Useful for users?

Coherence
Match with domain knowledge.

$g(x) =$ 

Controllability
Can user influence explanation?

$g(x)$ 

M. Nauta et al. "From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI." ACM Computing Surveys (2023).

Explanation / Model / User

Recap from last week

- The Red Rooster line: we found that neighbour methods worked better than logistic regression at predicting the chain (given the latitude and longitude)

Recap from last week

- The Red Rooster line: we found that neighbour methods worked better than logistic regression at predicting the chain (given the latitude and longitude)
- A line would have given us an *explainable* model.
- You could give the formula for latitude and longitude, or you could draw the decision boundary on a map (perhaps find some landmarks it goes through)

Recap from last week

- The Red Rooster line: we found that neighbour methods worked better than logistic regression at predicting the chain (given the latitude and longitude)
- A line would have given us an *explainable* model.
- You could give the formula for latitude and longitude, or you could draw the decision boundary on a map (perhaps find some landmarks it goes through)
- Neighbour methods were better at predicting, but we don't know *why* Red Rooster and Chargrill Charlie's stores are near other stores from the same chain

The secret to interesting data science projects

Bring together different datasets

Realistic variables that might influence which chain a fast food store will be

Realistic variables that might influence which chain a fast food store will be

- Suburb wealth (Chargrill Charlie's is a premium brand?)
- Cars per household (Red Rooster is the only one that usually has a drive-through)
- Daytime worker density (Nando's and Oporto's are often inside a mall)
- Arabic spoken at home (El Jannah's target Middle-Eastern diaspora users)

Acquire some data

> *The typical ML project involves about 15 seconds thinking about the model, and 1.5 years trying to get the data into it.* – Charles H. Martin, PhD

Where to get data from?

Internal systems Logs, databases, and so on

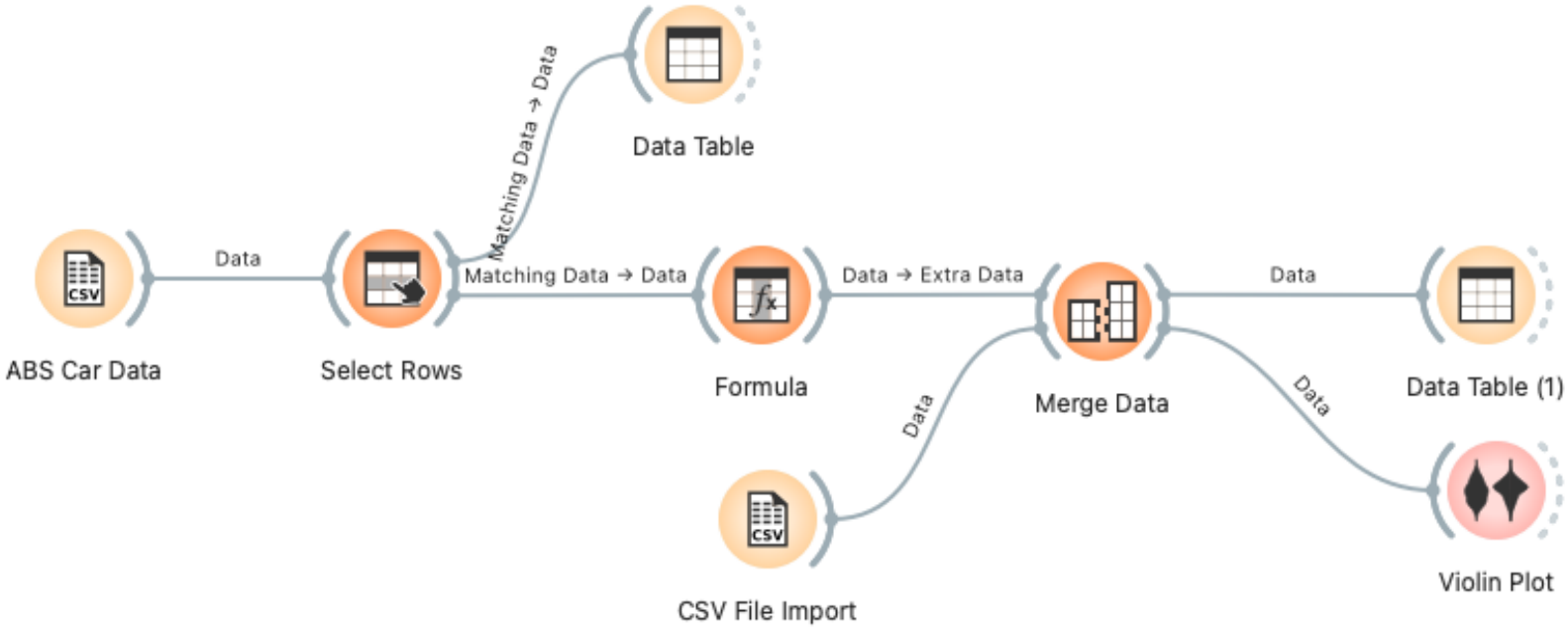
Commercial suppliers You can buy data sets

Open government data e.g. https://data.gov.au/

Or elsewhere... Conduct a survey, run a bot, ...?

https://digital.atlas.gov.au/

Worked example with the number of vehicles



- Formula to calculate average cars
- Merge on `ABS_SA2` with `Statistical Areas Level 2 2021 name`

What would we expect an explanation to look like?

If the average number of cars per household is greater than 3 and the land price is less than $1500 per square metre, expect it to be Red Rooster.

If the land value is more than $6000 expect Chargrill Charlie's

...

How would a computer find this?

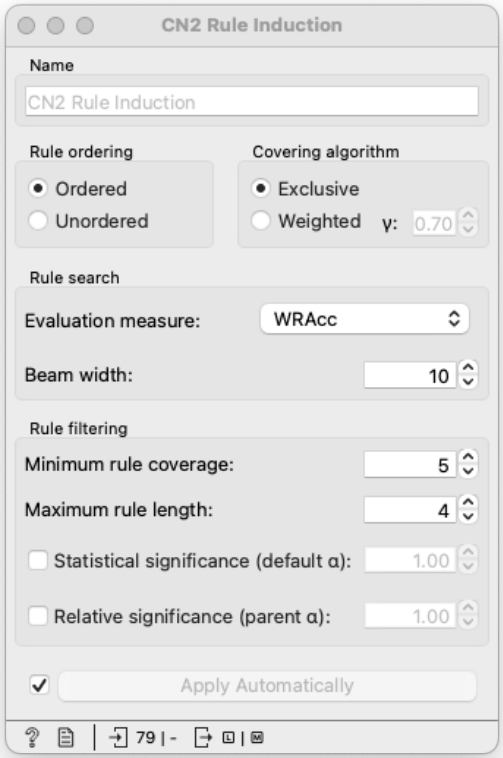
1. Pick a feature.
2. Find all the mid-point values between features
3. Ask if the targets above and below that value on the feature are in different proportions. If they are, that's interesting.
4. (CN2) Keep track of a few interesting features at a time. For the interesting features, try repeating steps 1–3 on those split datasets.
5. (CN2) Commit to a rule, and then take that data out of consideration

CN2 Rule Induction: compact, human-readable rules

- Learns a list of *if–then* rules that cover the positives. (Usually ordered list)
- Quality metric often: **WRAcc** (weighted relative accuracy):

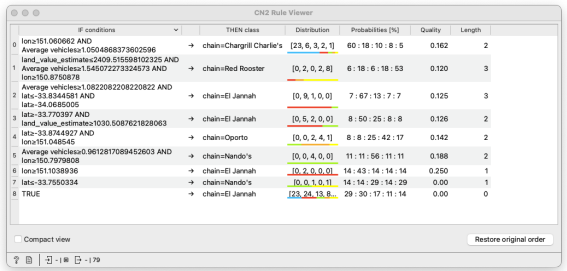
$$\mathrm{WRAcc} = \frac{\mathrm{TP}}{n} - \frac{\mathrm{cov}}{n} \cdot \frac{P}{n},$$

where cov is rule coverage.



Demo: CN2 in Orange (geospatial task)

1. CN2 Rule Induction from `File`.
2. CN2 Rule Viewer: read 2–3 top rules. Are they geographic commonsense?
3. Send `CN2` into `Test & Score`; compare AUCvs Tree/Logistic/kNN.
4. Quick check: **overlap** and **default** rule at the end.



Decision trees

- Save lives[1]
- Make data scientists look awesome
- "If you follow this simple diagram, we make lots of money."

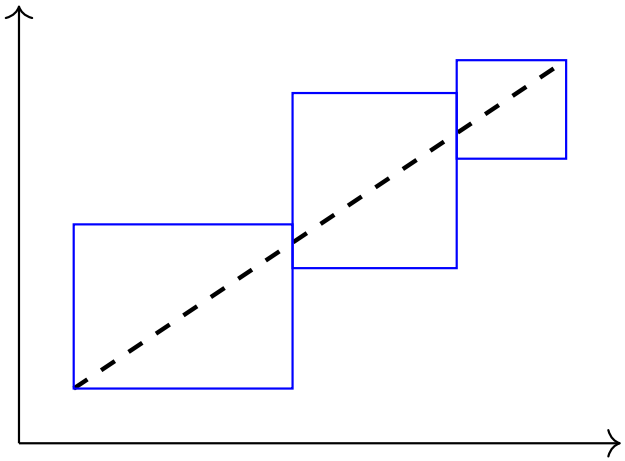
[1] http://www.ncbi.nlm.nih.gov/pubmed/25160603

Decision Trees: the idea

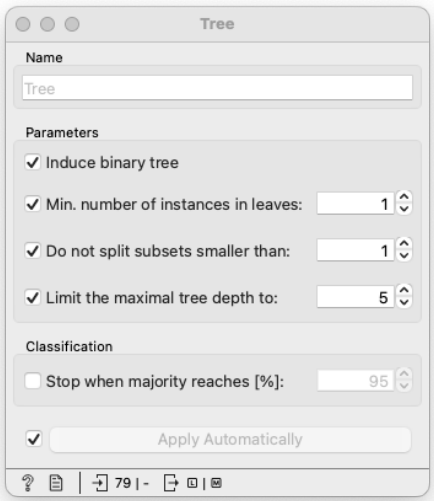
- Greedy, axis-aligned splits to reduce impurity.
- **Gini** for node t: $G(t) = \sum_k p_k(1 - p_k) = 1 - \sum_k p_k^2$.
  - The proportion of samples belonging to the class multiplied by the proportion that don't belong
  - $0 =$ "pure"; $0.5 =$ "maximum impurity"
- **Entropy** for node t: $H(t) = -\sum_k p_k \log_2 p_k$.
  - Calculate how 'surprising' it would be to randomly pick a sample from that class (using the logarithm), weight this surprise by how often that class actually appears
  - $0 =$ "no surprise at all, already know what the answer will be"; $1 =$ "no way of predicting anything at all"
- Split quality: $\Delta = I(\text{parent}) - \sum_i \frac{n_i}{n} I(\text{child}_i)$ with $I \in \{G, H\}$.
- Bias–variance: deep trees overfit; shallow trees underfit.

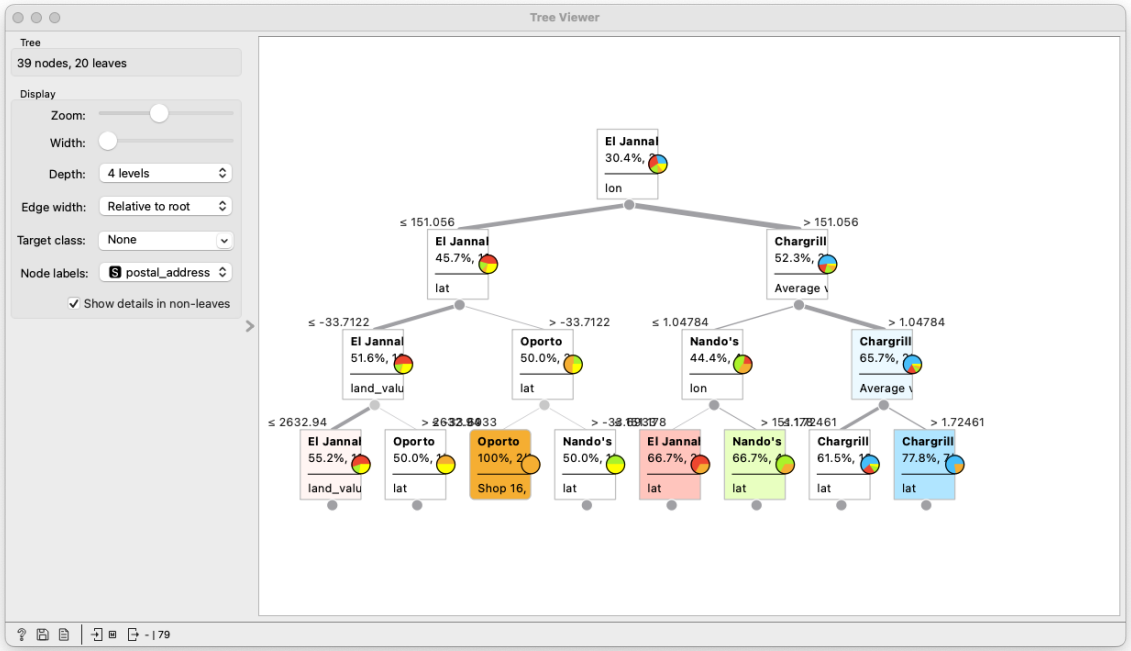
Decision Trees

- Trees divide the data space up into *rectangles*
- If the "line" is oblique $\Rightarrow$ trees approximate it with a staircase of rectangles.



Demo: Decision Tree in Orange

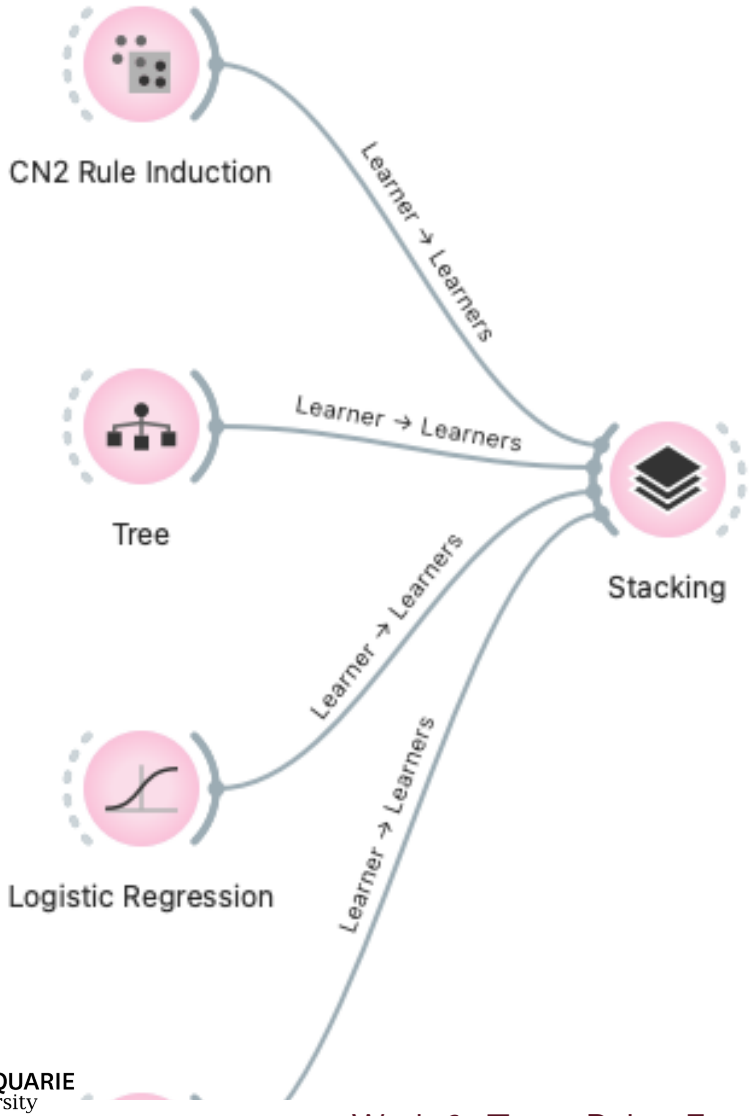




Stacking: two heads are smarter than one

- Train some models
- **Classifiers:** have them vote on the answer; **Regressors:** take the average of the model's output
- Helps when learners capture complementary patterns (e.g., linear + local + non-linear).
- Genius move (not possible with Orange): give more votes to some models based on confidence or validation accuracy

Stacking in Orange

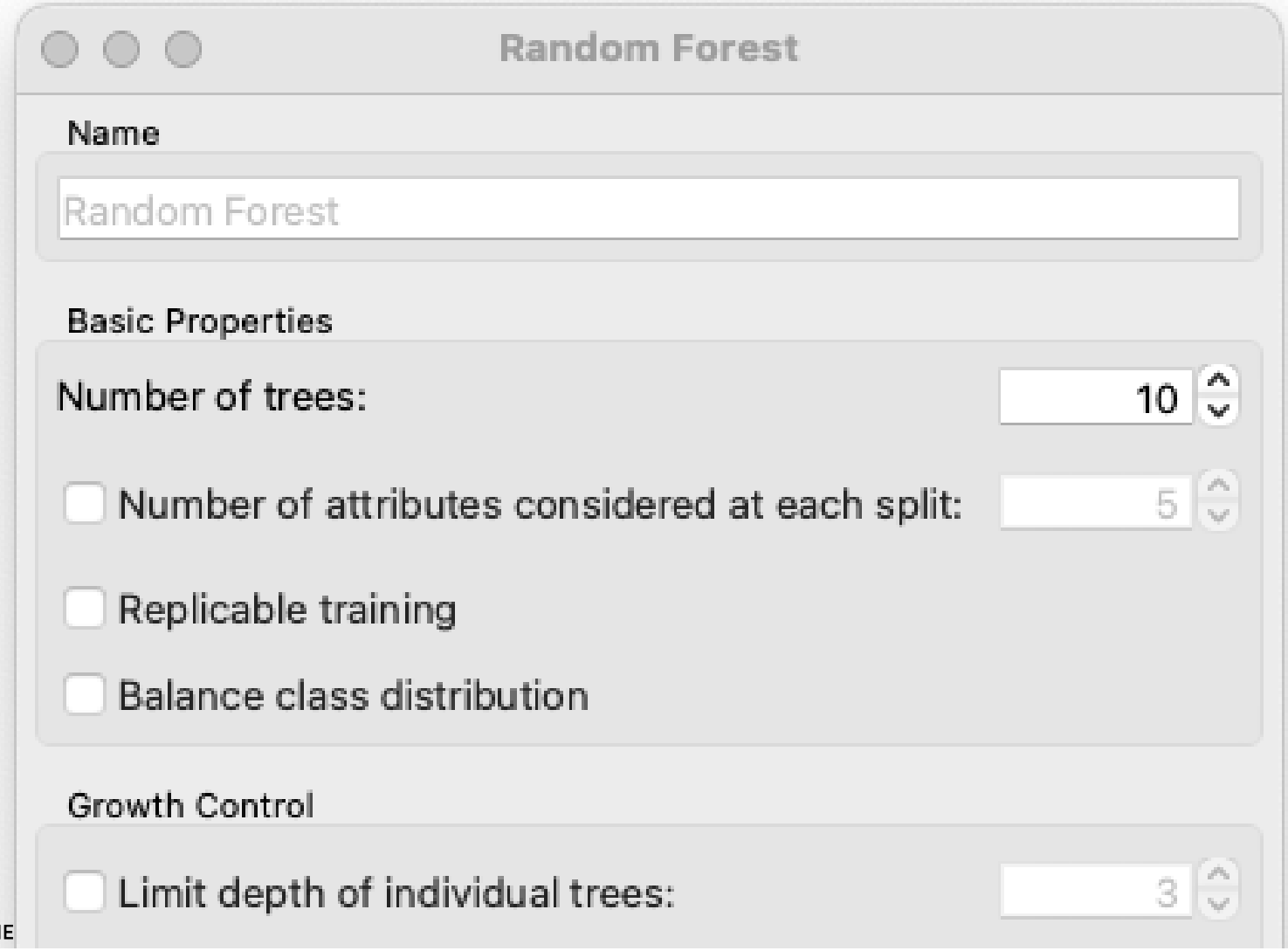


Random Forests

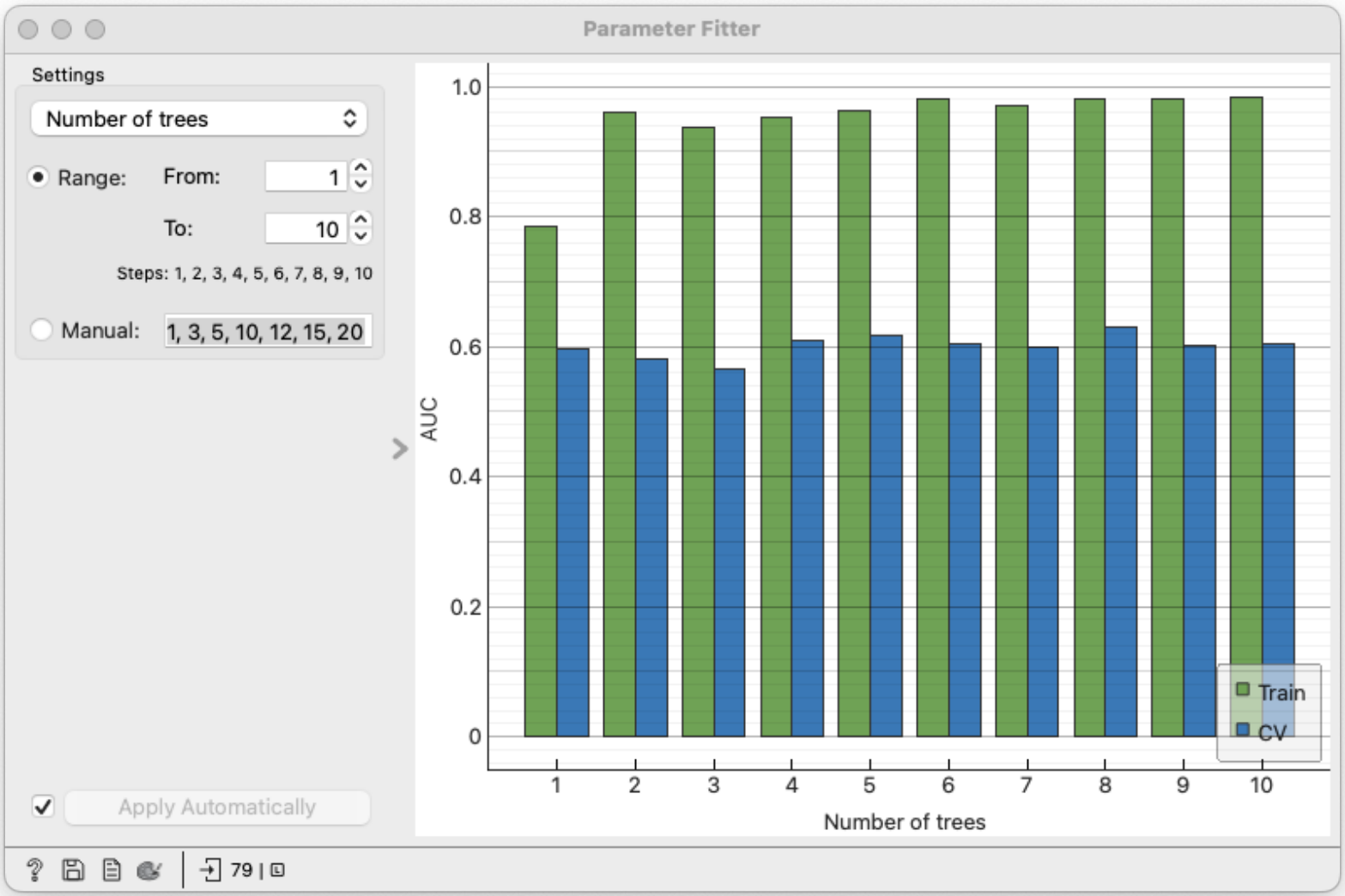
- We can make sure that decision trees capture complementary patterns: restrict which features they can split on
- We can make dozens of different decision trees by randomly selecting which features they are allowed to use each time they want to make a split
- Then we aggregate the decision trees (stack them, or some other variations)
- Lowers variance, keeps bias similar to trees; robust on tabular data.
- We keep some explainability: we can see which features where the most used across the trees

If you are dealing with tabular data, random forests will usually perform very well, and often be the best

Random Forest in Orange



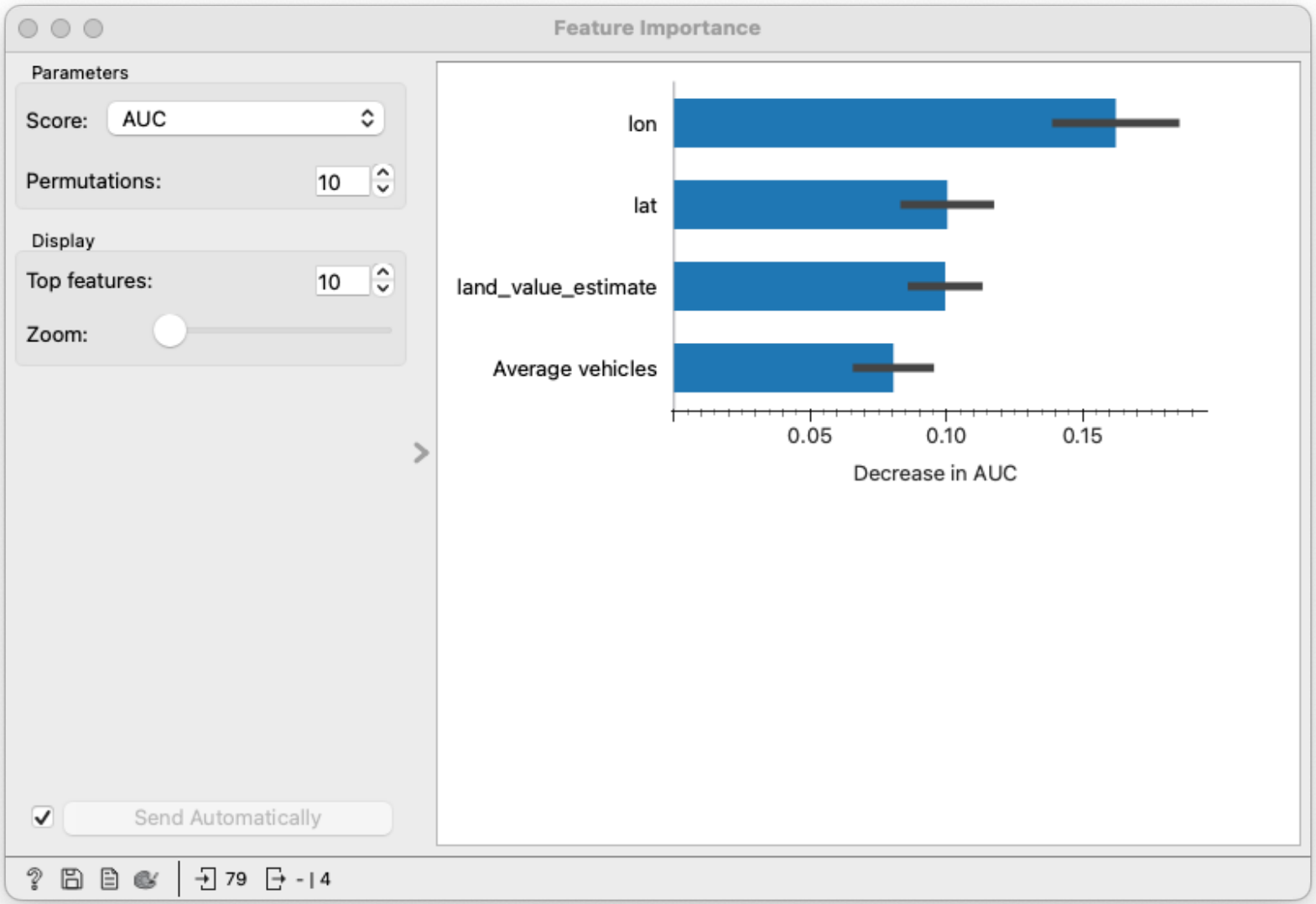
Random Forest Parameter Fitting in Orange



Hyperparameter tuning in Orange (what it can & can't do)

- **Parameter Fitter** tunes *one* integer parameter per run; expects a *Learner* input.
- Doesn't work for many Orange widgets (when we program in Python we'll do lots of this)

Random Forest: feature importance



Decision trees and random forests

Can also be regressors.

Example: land values using decision trees

Common pitfalls (callouts)

- "High AUC means good probabilities": **False**. Use calibration plots or calibrate.
- "Stacking always wins": **Nope**. If learners are redundant, meta adds nothing.
- "Parameter Fitter will tune everything": **No**. One integer param per run, learner input only.
- "Trees are interpretable by default": only when shallow; deep trees are not human-scale.

Summary — Week 6

- **Evaluation:** ROC curves show ranking ability; AUC = probability positive outranks negative.
- **Explainability vs Accuracy:** sometimes you must (GDPR/PIPL); sometimes you should (trust, debugging, ethics).
- **CN2 Rules:** human-readable *if–then* rules; good for insight.
- **Decision Trees:** interpretable (when shallow); risk of over/underfitting.
- **Random Forests:** ensembles of trees, robust on tabular data, still partly explainable.
- **Stacking:** combine diverse learners for complementary strengths.