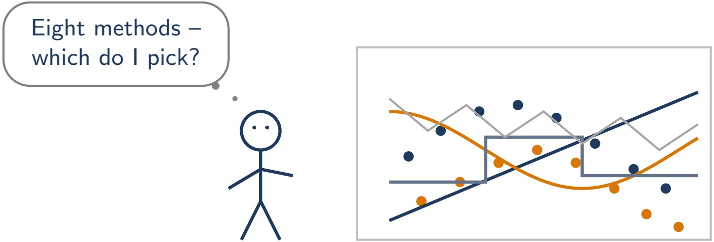
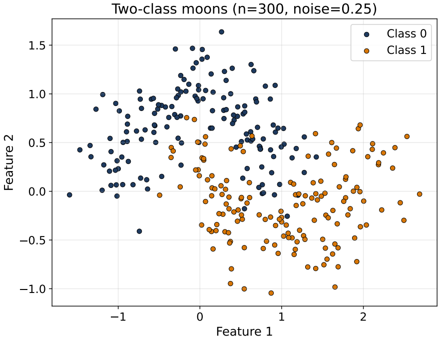


One dataset. Eight choices. Different boundaries.

Why so many classification methods?

- Banks classify loans into default/no-default.
- Card networks classify transactions into fraud/no-fraud.
- Brokers classify trade signals into act/skip.

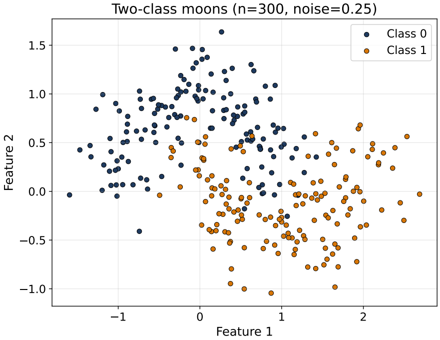
Dozens of methods exist. They all output the same kind of label. They differ in HOW they decide.



Before we start – can you name three classification methods?

What does our test dataset look like?

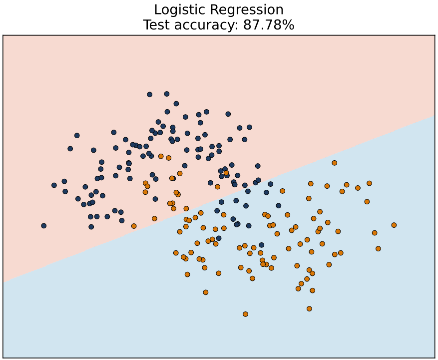
- 300 synthetic 2D points, two interlocking crescents (the “moons” dataset).
- 70/30 train/test split, seed=42.
- Every method below is trained on the SAME 210 training points and scored on the SAME 90 test points.
- **Class 0** = navy. **Class 1** = amber.



make_moons is scikit-learn’s canonical non-linear benchmark dataset.

How does Logistic Regression draw the line?

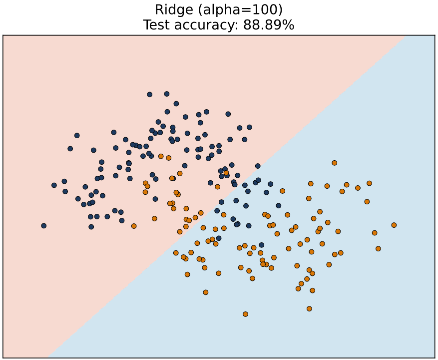
- Decision rule: predict class 1 if $\sigma(w^\top x + b) > 0.5$.
- The boundary is where $w^\top x + b = 0$ – a single straight line.
- On moons: $\sim$88% test acc. The line gets close but cannot curve.
- Trains in milliseconds. Easy to interpret coefficients.



Why can logistic regression never draw a curved boundary without feature engineering?

What does Ridge do when alpha is huge?

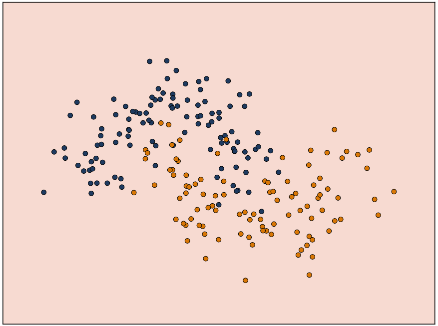
- RidgeClassifier with $\alpha = 100$ shrinks weights toward zero – but on moons it still finds a working straight-line boundary ($\sim$89%).
- Same family as Logistic Regression. Same straight-line shape. Slightly different slope from the L2 penalty.
- To see Ridge actually fail, you would need much larger α (10^4 or more) – moons is close to linearly separable when scaled.
- The lesson: regularization changes the boundary's tilt, not its shape family.



Why does Ridge with huge α still work here, while Lasso with $C = 0.01$ collapses?

What does L1-penalized logistic (Lasso) look like?

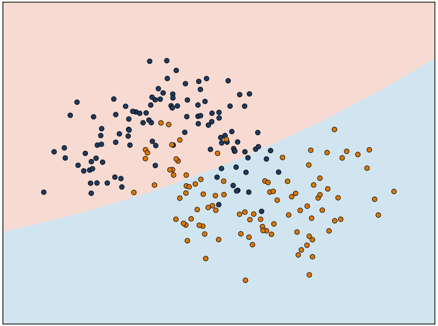
- L1 penalty with $C = 0.01$ pushes weights all the way to zero.
- Boundary collapses toward majority class – $\sim$42% acc on moons (worse than majority baseline).
- Strong L1 is a feature-selection tool, not a curve-drawing tool.
- Use it when you have 500 features and suspect only a handful matter.



Set C=100 instead of 0.01. What boundary would you get?

How does Naive Bayes carve space?

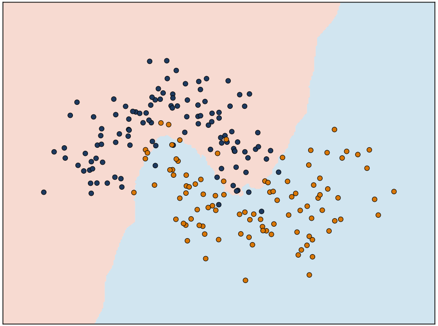
- Fits a Gaussian per-feature per-class. Assumes features are independent given the class.
- Decision rule: $\arg\max_c P(c) \prod_i P(x_i \mid c)$.
- Boundary is quadratic (Bayes-optimal under the assumed Gaussian).
- $\sim$89% acc on moons. Often surprisingly strong on text classification.



Naive Bayes is “naive” because it ignores feature correlations.

What does KNN look like with k=15?

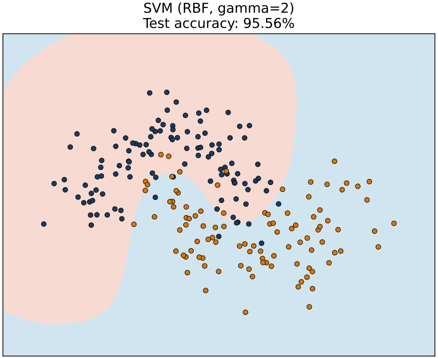
- Predict the majority class among the 15 nearest training points.
- Boundary follows local point density – bumpy where classes intermix.
- $\sim$94% acc on moons. No model is "trained" – the data IS the model.
- Slow at predict time on big data: every query searches all training points.



$K = 1$ **vs** $K = 50$ – **which overfits, which underfits?**

How does an RBF-kernel SVM fit the curves?

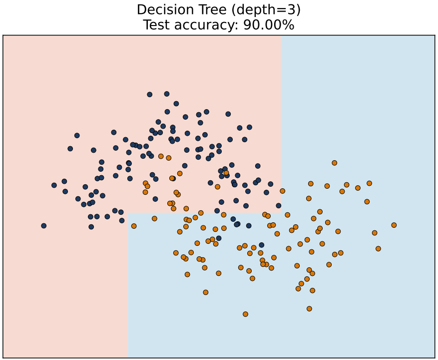
- RBF kernel maps points to an infinite-dimensional space.
- SVM finds the maximum-margin separator there.
- In original 2D, the boundary is smooth and curved. $\sim$96% acc on moons – the highest of the eight.
- Hyperparameter γ controls boundary tightness around each point ($\gamma = 2$ is moderate).



High γ = tight boundary around each point. Low γ = near-linear.

How does a depth-3 decision tree split space?

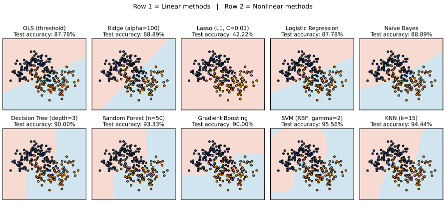
- At each node: split on one feature at a threshold that maximizes information gain.
- `max_depth=3` $\Rightarrow$ at most $2^3 = 8$ leaves.
- Boundary is axis-aligned rectangles – horizontal and vertical only.
- $\sim$90% acc on moons. Easy to draw, easy to explain to a regulator.



What shape would `max_depth=1` produce? What about `max_depth=20`?

Can we see all eight at once?

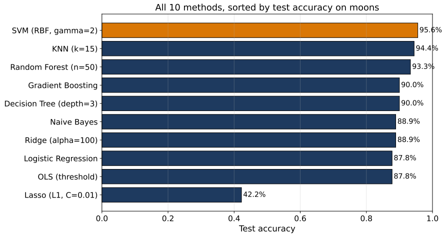
- Row 1 = linear methods. Most struggle on moons (curved boundary required).
- Row 2 = nonlinear methods. Most succeed – accuracy spread across all ten methods is ~53 percentage points (42% to 96%).
- Same dataset, same seed, same train/test split. The only thing that differs is the algorithm.
- **Reading boundary shape tells you the family before you read accuracy.**



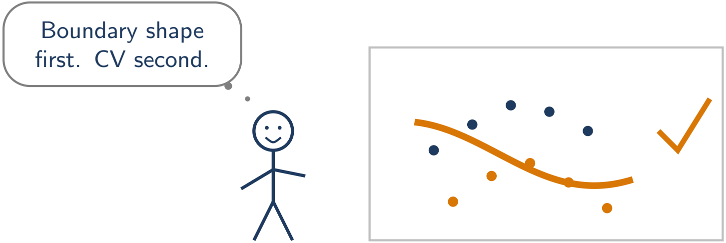
This single chart justifies every later lesson on cross-validation and hyperparameter tuning.

Which method should I reach for first?

- **Interpretable + linear data:** Logistic Regression or Lasso.
- **Tabular + mixed shapes:** Random Forest or Gradient Boosting.
- **Small data, any shape:** SVM with RBF kernel.
- **Streaming + few features:** KNN.
- **Always:** cross-validate. The single train/test number can mislead.



Name one method you would pick for a credit-scoring model and justify in one sentence.



Eight methods, one habit: read the shape before you trust the number.