K-NN and Performance Metrics

Machine Learning Models

Reminder: The goal of ML is to make **good predictions**, NOT to understand the **exact relationships**.

We will discuss situations in which different models are good or bad. This is because we want to make sure that we are **covering all possible cases** of the data.

Example: Linear regression is good for linear relationships. We won't always know if we have a linear relationship (due to dimensionality), but we can make sure we include methods that are good for linear and nonlinear relationships.

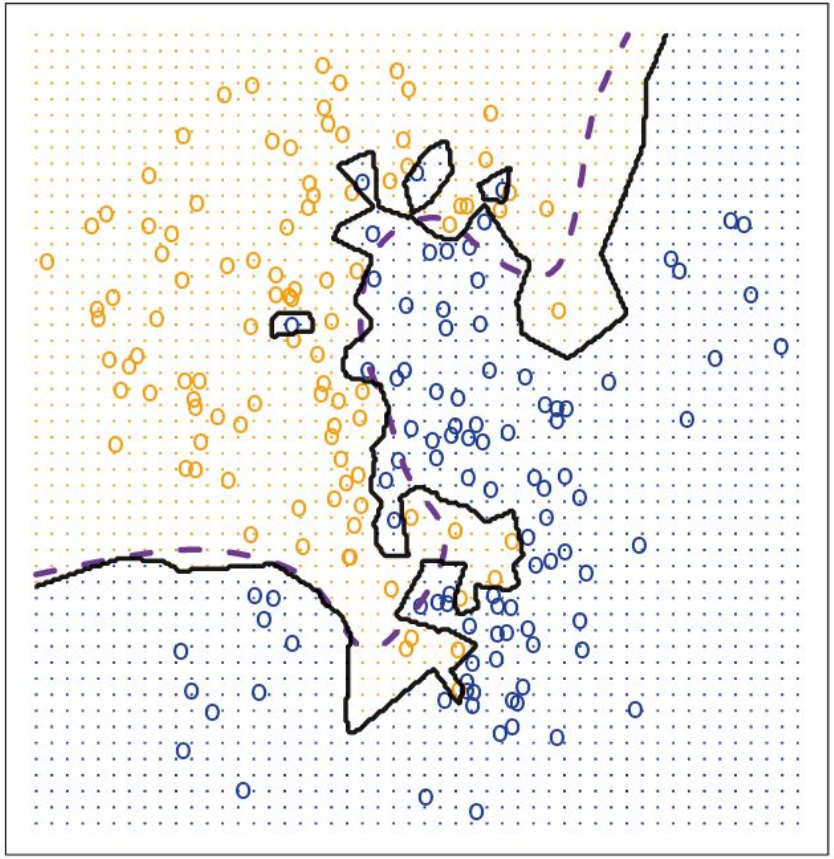
K-Nearest Neighbors

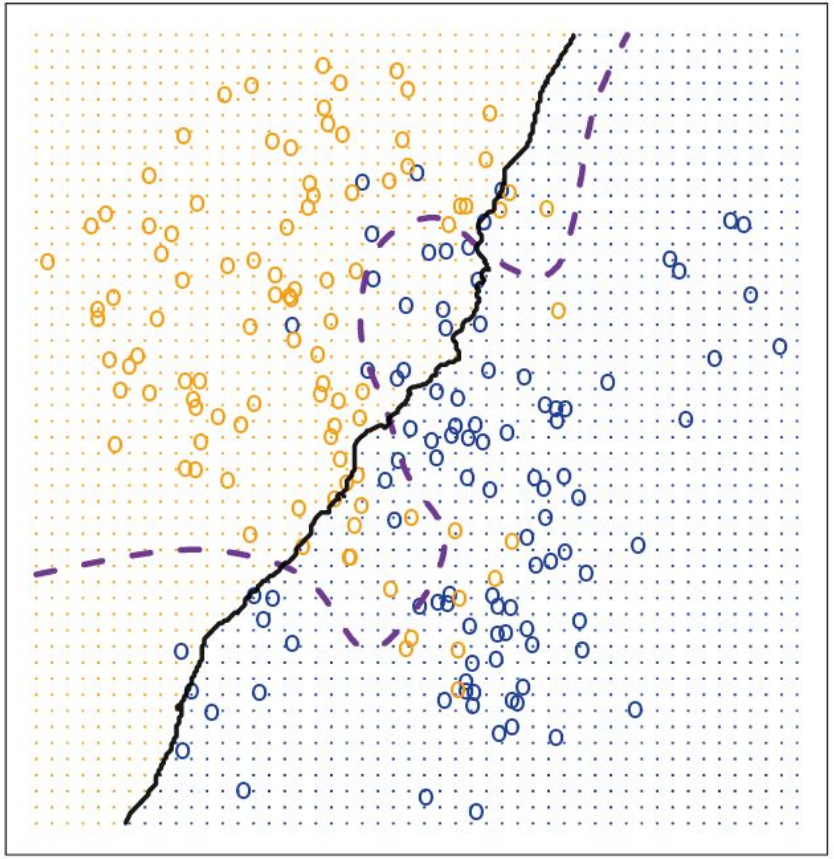
Many times, odd values of k are used to avoid ties. This isn't strictly necessary though.

- Remember, threshold does not need to be 0.5.

K is a tuning parameter. Try many different values and use the one that has the best performance.

- Might depend on sparsity of data. Look at graphs of performance and use different values of k as needed.





Features in K-NN (and models in general)

Note that the features for K-NN do not need to be spatial. It is just easiest to visualize them in that way. Many times, it is **high dimensional.**

K-NN is **dependent on scaling of features** because it's based on distance.

K-NN performs **poorly with sparse data**.

- If there aren't many near neighbors, it won't make good predictions.

K-NN Algorithm

Features must be **numerical** in some way.

- Categorical variables must be converted into numerical values.
- Note: Must scale these along with the numerical variables!

Weighted K-NN: Weight closer points higher. For example, use 1/distance as the weight.

kNN Distance Measures

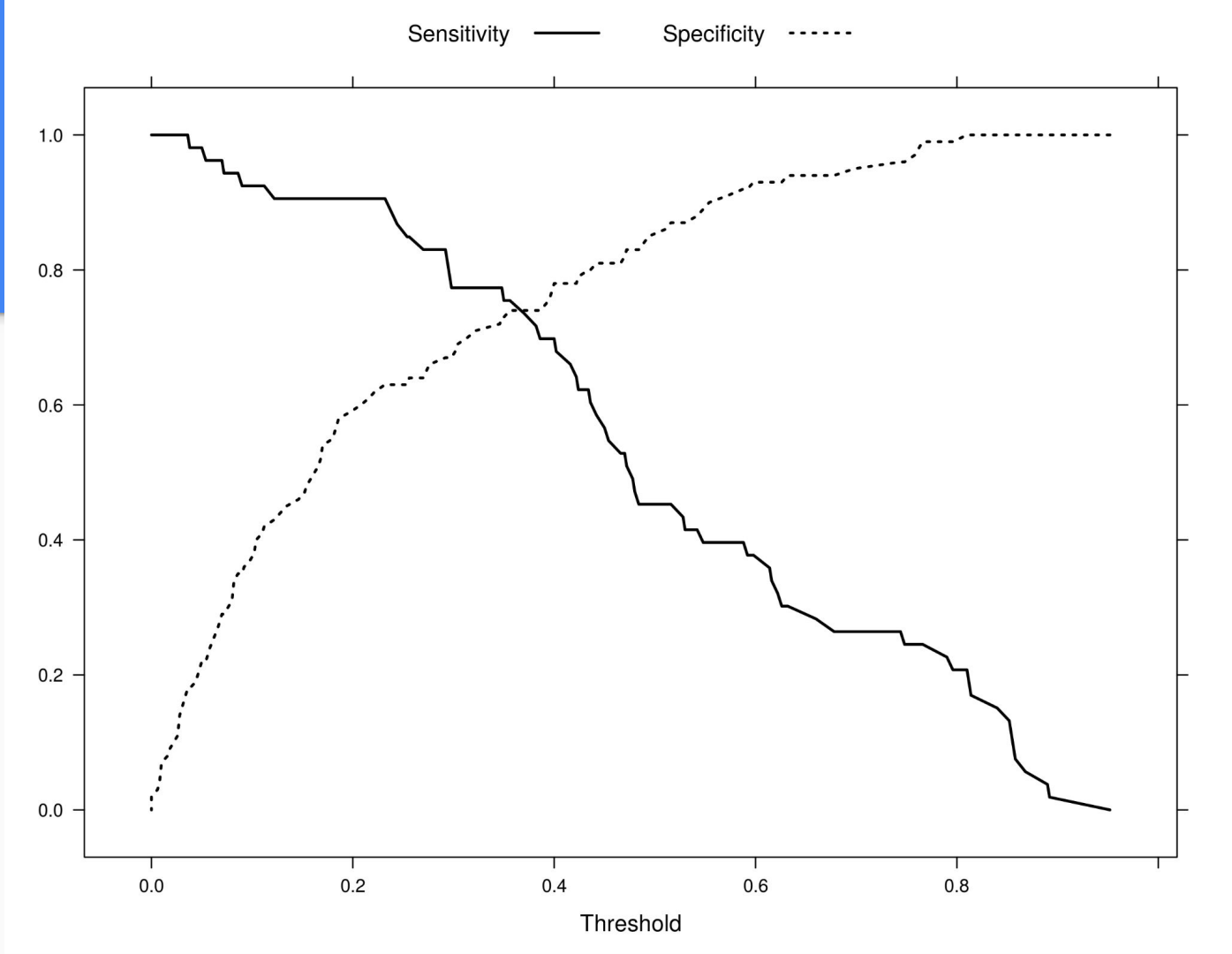
Different distance measures might lead to different results because of how variables might be scaled

Can also tune choice of distance measures using CV.

Precision vs. Recall

Note: The intersection does not matter because there is **no meaning behind precision and recall being the same**.

Precision and Recall have different denominators and thus have different **baselines** to compare against.


Sensitivity
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
Threshold

Predictions as Scores

For categorical variables, you usually get a value that is “probability-like” as your prediction

- Logistic regression: predicted probability
- K-NN: Proportion of positive cases
- Weighted K-NN: Weighted proportion of positive cases

Think of these are **scores**, not probabilities. The higher the score, the more likely it is to be a positive case.

Using Scores

Predicted scores have no intrinsic meaning (again, don't think of them as probabilities!). Default cutoff of 50% doesn't necessarily mean that should be how you should make predictions.

Use different **threshold** values depending on the precision/recall tradeoff (or other measures) that you want.

Using top x% of data

Sometimes, we don’t have the ability to choose a threshold as we want.

- Example: **Budget constraint** for providing interventions.

“**Downside**”: Usually does not provide “best performance” (because it is a constraint)

Using Multiple Performance Measures

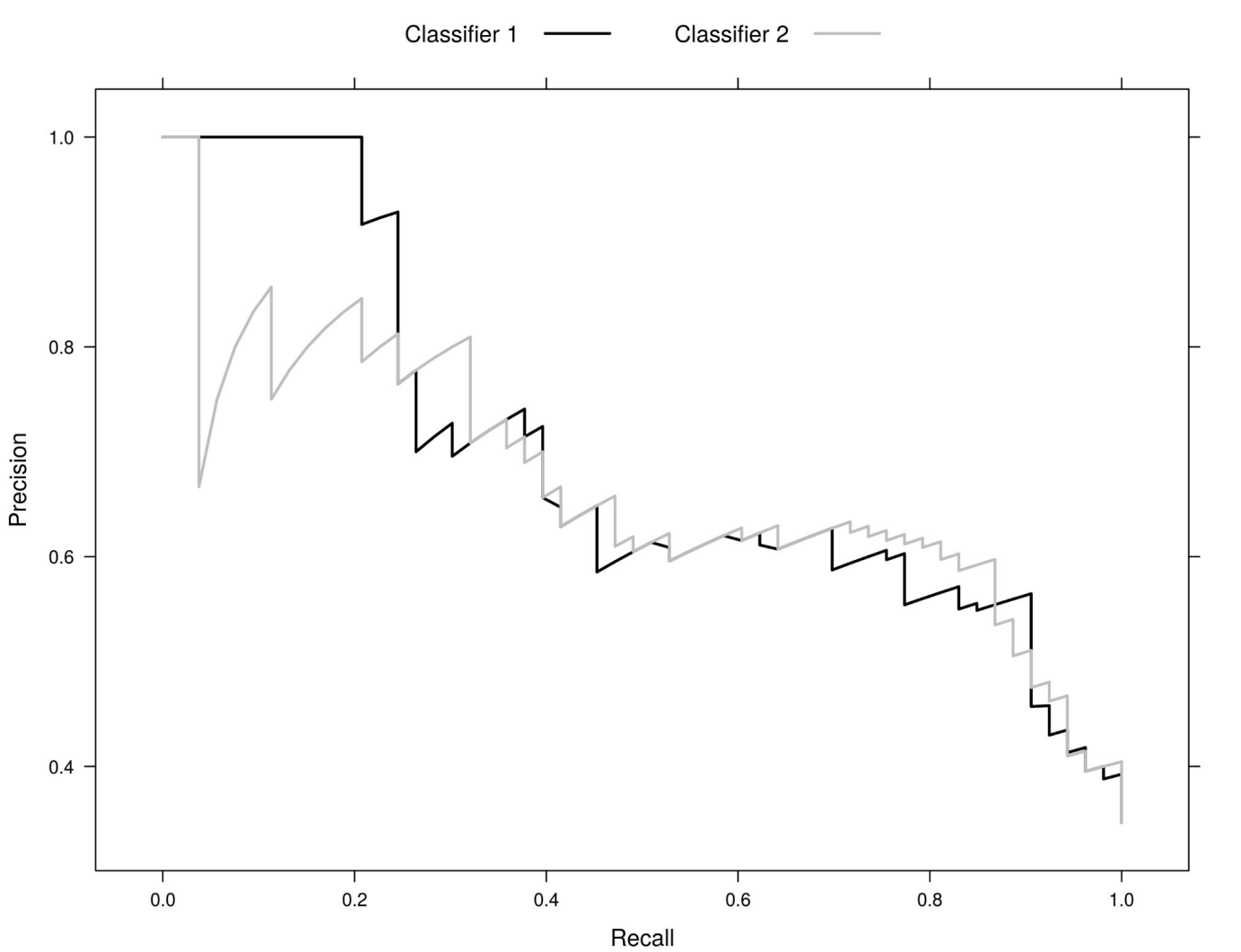
Example: We want to prevent as many cases of lead poisoning as possible, but want to make sure we are being efficient with the intervention because that money can be used for other programs more efficiently.

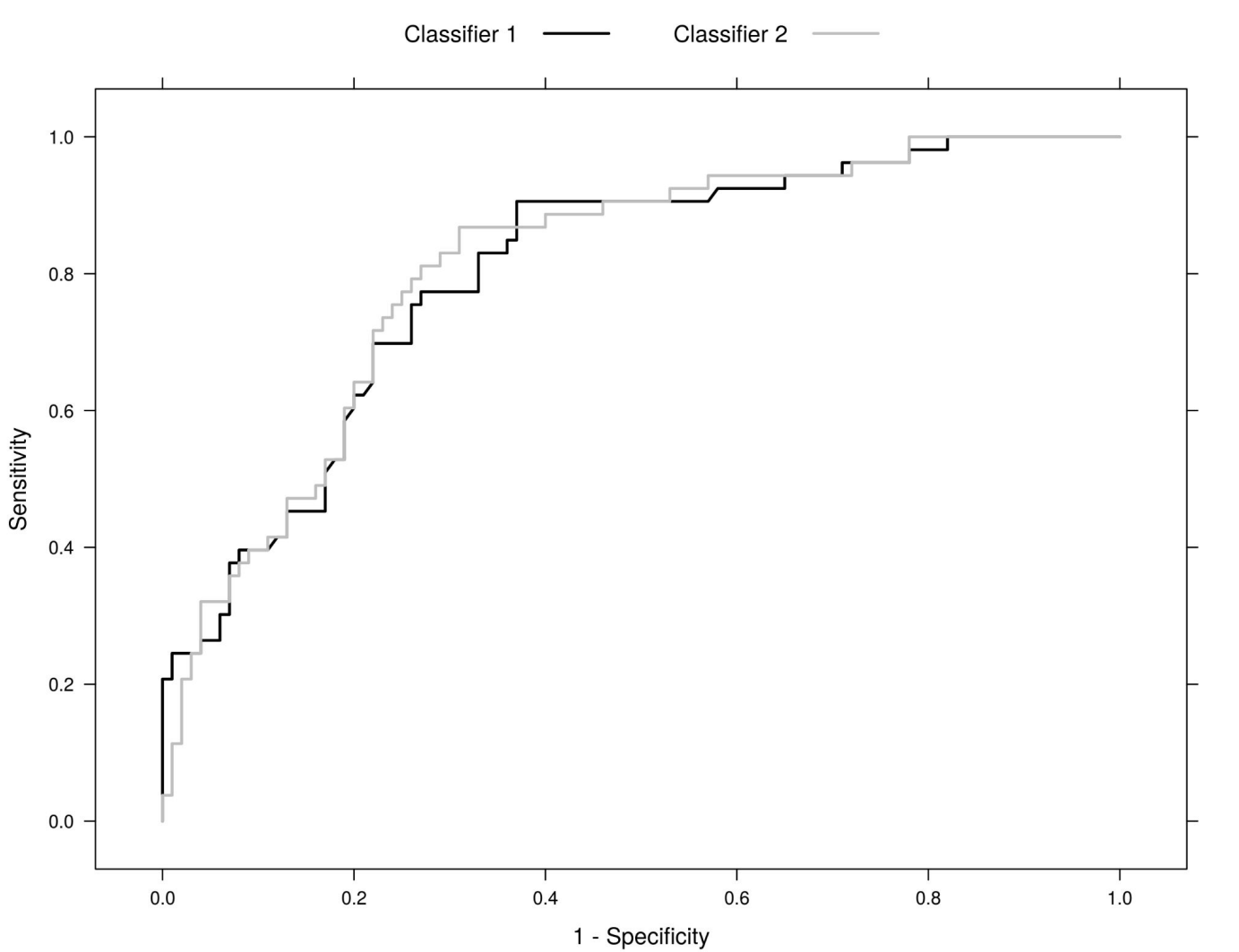
- **One possible way of determining best model**: Choose model that maximizes recall with a precision higher than 50%.

ROC and PR Curves

Typically used more as a way to compare models at a range of sensitivity/specificity or precision/recall values.

Can use AUC values to choose model, though you need to specify threshold when actually making predictions


Classifier 1
Classifier 2
Precision
Recall
1.0
0.8
0.6
0.4
0.0
0.2
0.4
0.6
0.8
1.0


Classifier 1
Classifier 2
Sensitivity
1 - Specificity
0.0
0.2
0.4
0.6
0.8
1.0