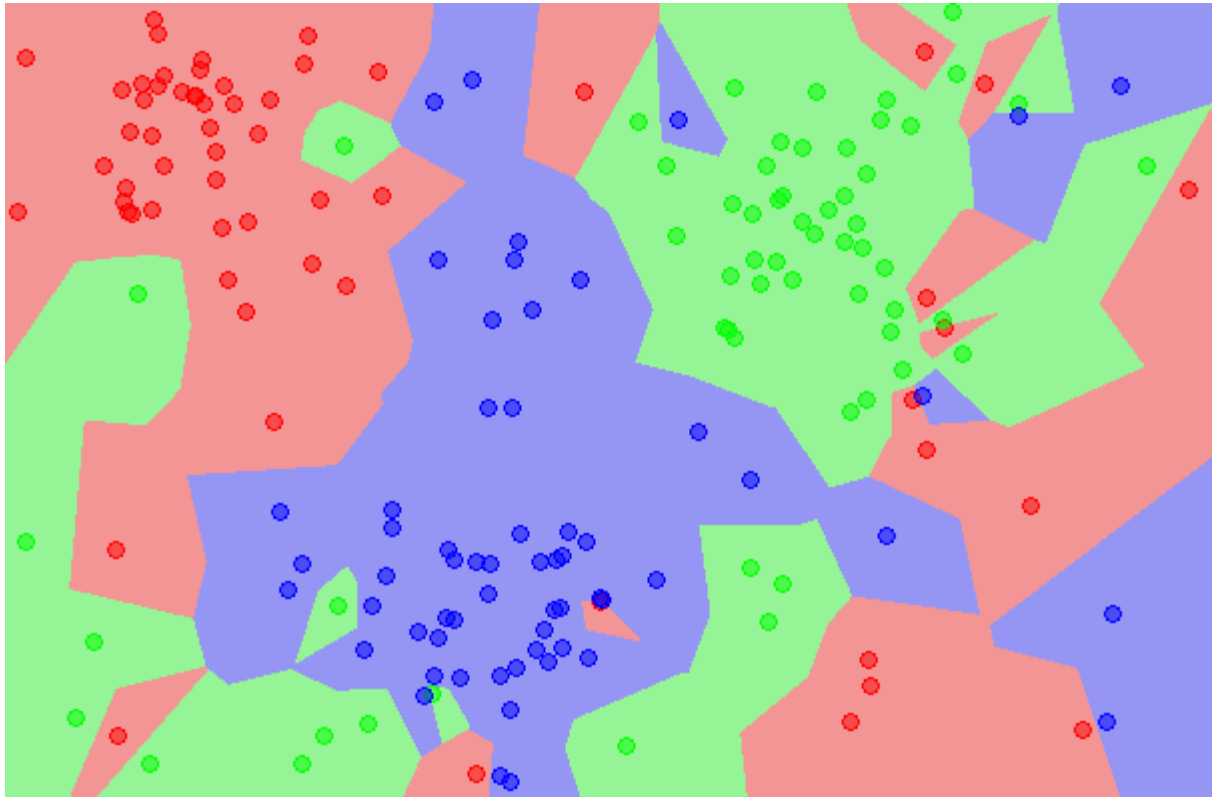
University of Ljubljana, Faculty of Computer and Information Science

Bias, variance and predictive models



Prof Dr Marko Robnik-Šikonja

Intelligent Systems, Edition 2024

Contents

- Bias and variance of prediction models
- Bayes optimal classifier
- Simple regression models:
  - linear models, nearest neighbor, regression trees, regression rules
- Simple classification models:
  - nearest neighbor, naïve Bayes, decision trees, decision rules, logistic regression
- Biases in data

A generalization problem

- Our method has generally been designed to make error small on the training data, e.g., with linear regression, we choose the line such that MSE is minimized.

- What we really care about is how well the method generalizes i.e. how well it works on new data. We call this new data "**Test data**".

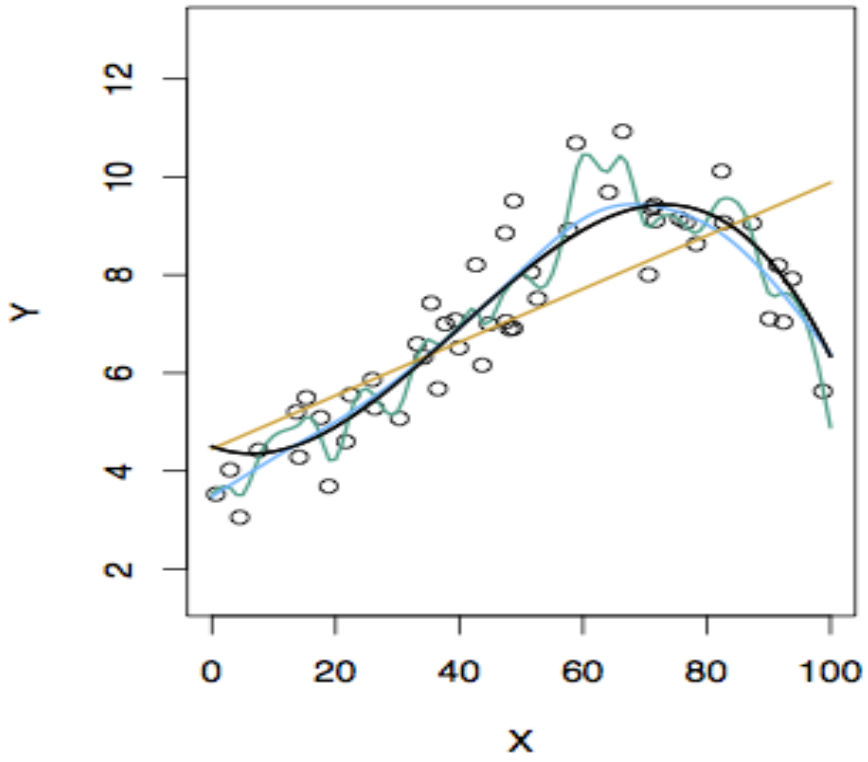
- There is no guarantee that the method with the smallest training error will have the smallest test (i.e. new data) error.

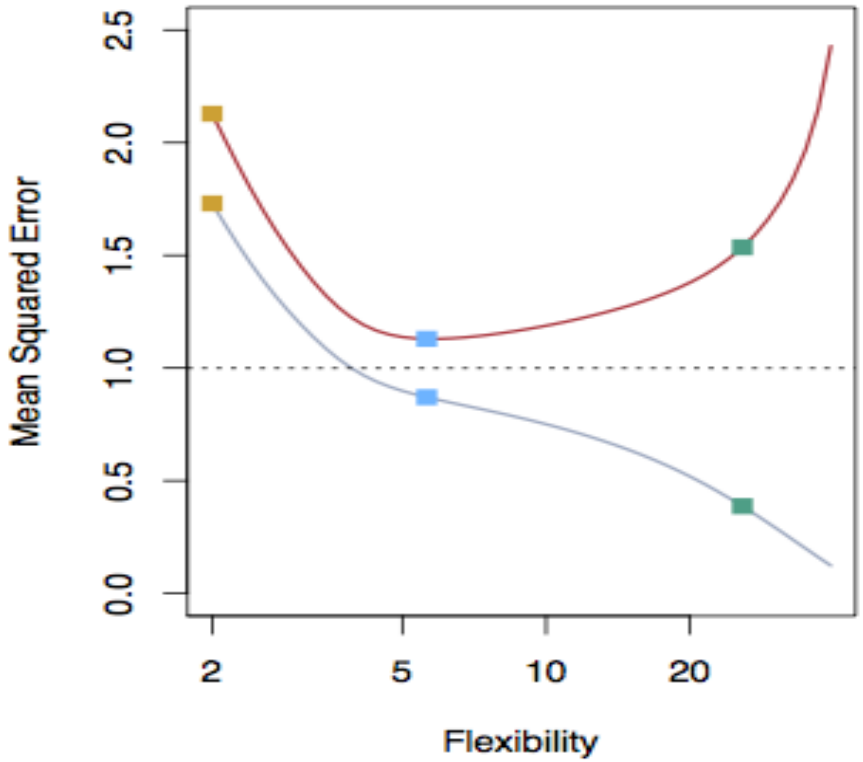
Training vs. test error

- In general the more flexible a method is the lower its training MSE will be, i.e. it will “fit” or explain the training data very well.
  - More flexible methods (such as splines) can generate a wider range of possible shapes to estimate *f* as compared to less flexible and more restrictive methods (such as linear regression). The less flexible the method, the easier to interpret the model. Thus, there is a trade-off between flexibility and model interpretability.

- However, the test MSE may in fact be higher for a more flexible method than for a simple approach like linear regression.

Different levels of flexibility: example 1





LEFT

Black: Truth

Orange: Linear Estimate

Blue: smoothing spline

Green: smoothing spline (more flexible)

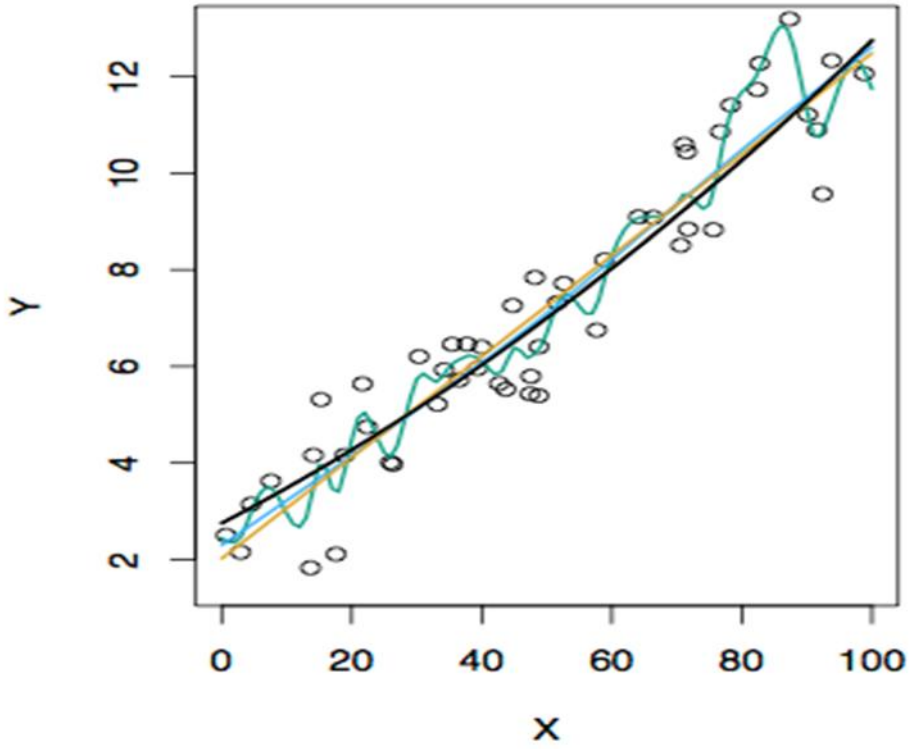
RIGHT

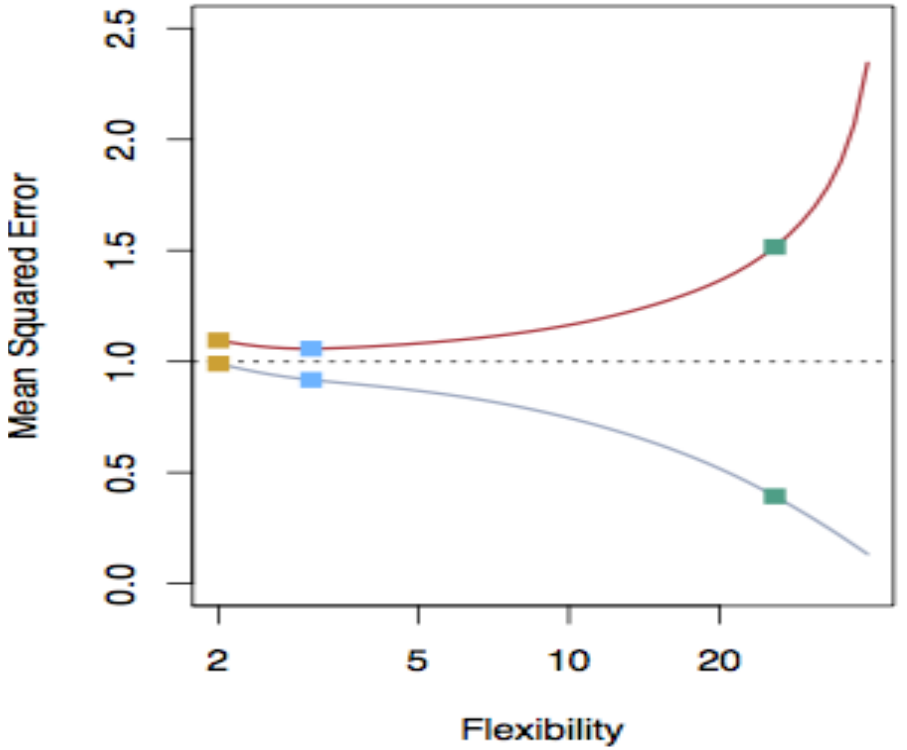
RED: Test MS

Grey: Training MSE

Dashed: Minimum possible test MSE (irreducible error)

Different levels of flexibility: example 2





LEFT

Black: Truth

Orange: Linear Estimate

Blue: smoothing spline

Green: smoothing spline (more flexible)

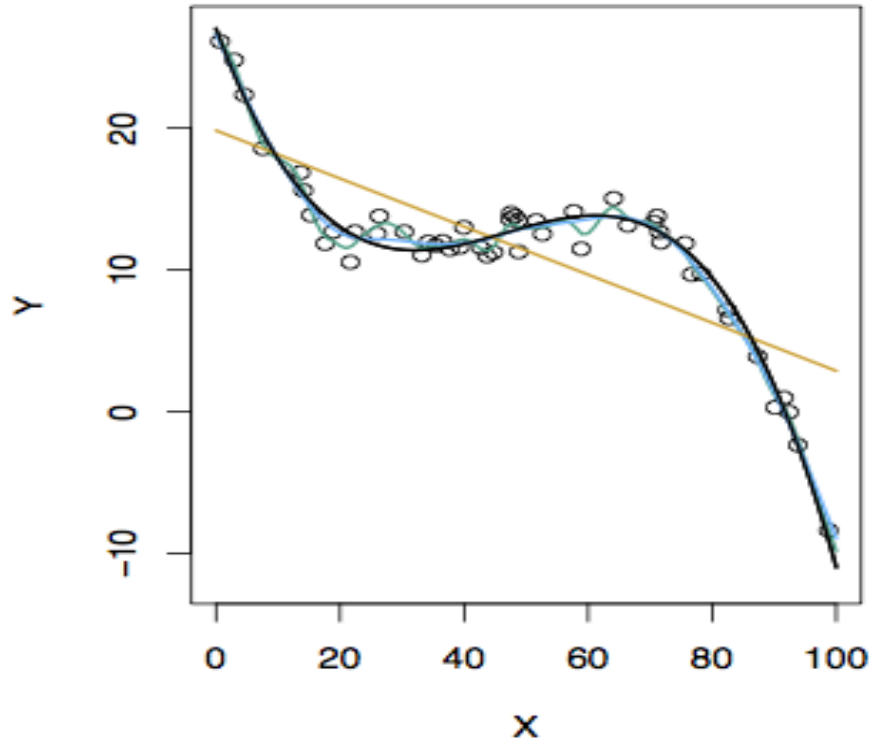
RIGHT

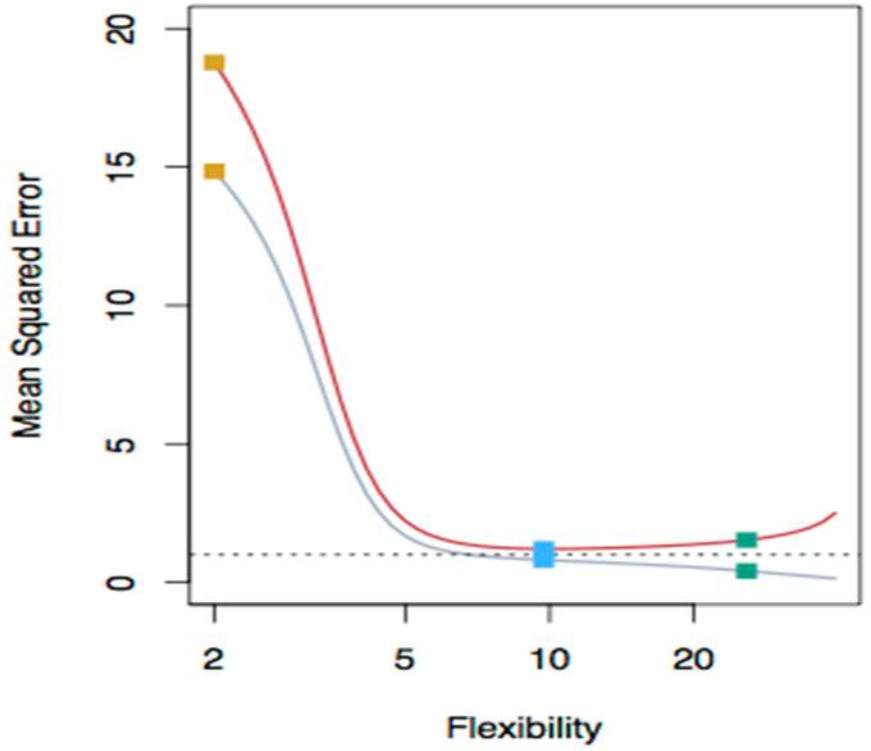
RED: Test MSE

Grey: Training MSE

Dashed: Minimum possible test MSE (irreducible error)

Different levels of flexibility: example 3





LEFT

Black: Truth

Orange: Linear Estimate

Blue: smoothing spline

Green: smoothing spline (more flexible)

RIGHT

RED: Test MSE

Grey: Training MSE

Dashed: Minimum possible test MSE (irreducible error)

Bias - variance trade-off

- The previous graphs of test versus training MSE's illustrates a very important trade-off that governs the choice of statistical learning methods.

- There are always two competing forces that govern the choice of learning method, i.e. bias and variance.

Bias of learning methods

- Bias in general: inclination or prejudice for or against one person or group, especially in a way considered to be unfair.

- Bias in ML refers to the error that is introduced by modeling a real life problem (that is usually extremely complicated) by a much simpler problem.
- A common definition of bias:

$$\text{Bias} = E[Y]-f(x)$$

- For example, linear regression assumes that there is a linear relationship between Y and X. It is unlikely that, in real life, the relationship is exactly linear so some bias will be present.

- The more flexible/complex a method is the less bias it will generally have.

Variance of learning methods

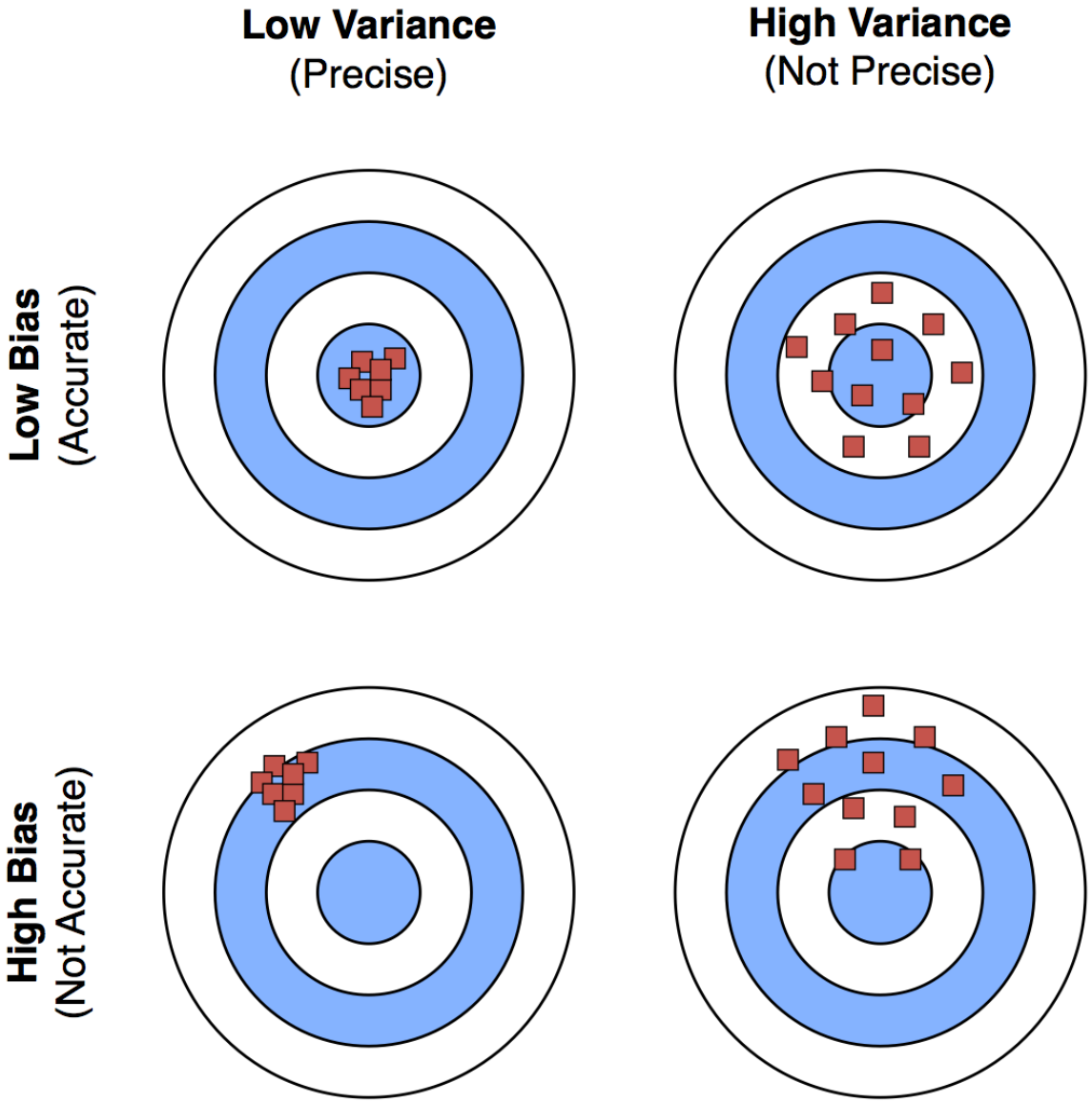
- Variance refers to how much your estimate for f would change if you had a different training data set.

- A common definition of variance:

  $\text{Var} = E[(Y - E[Y])^2]$

- Generally, the more flexible a method is, the more variance it has.

Bias-variance illustration



The trade-off?

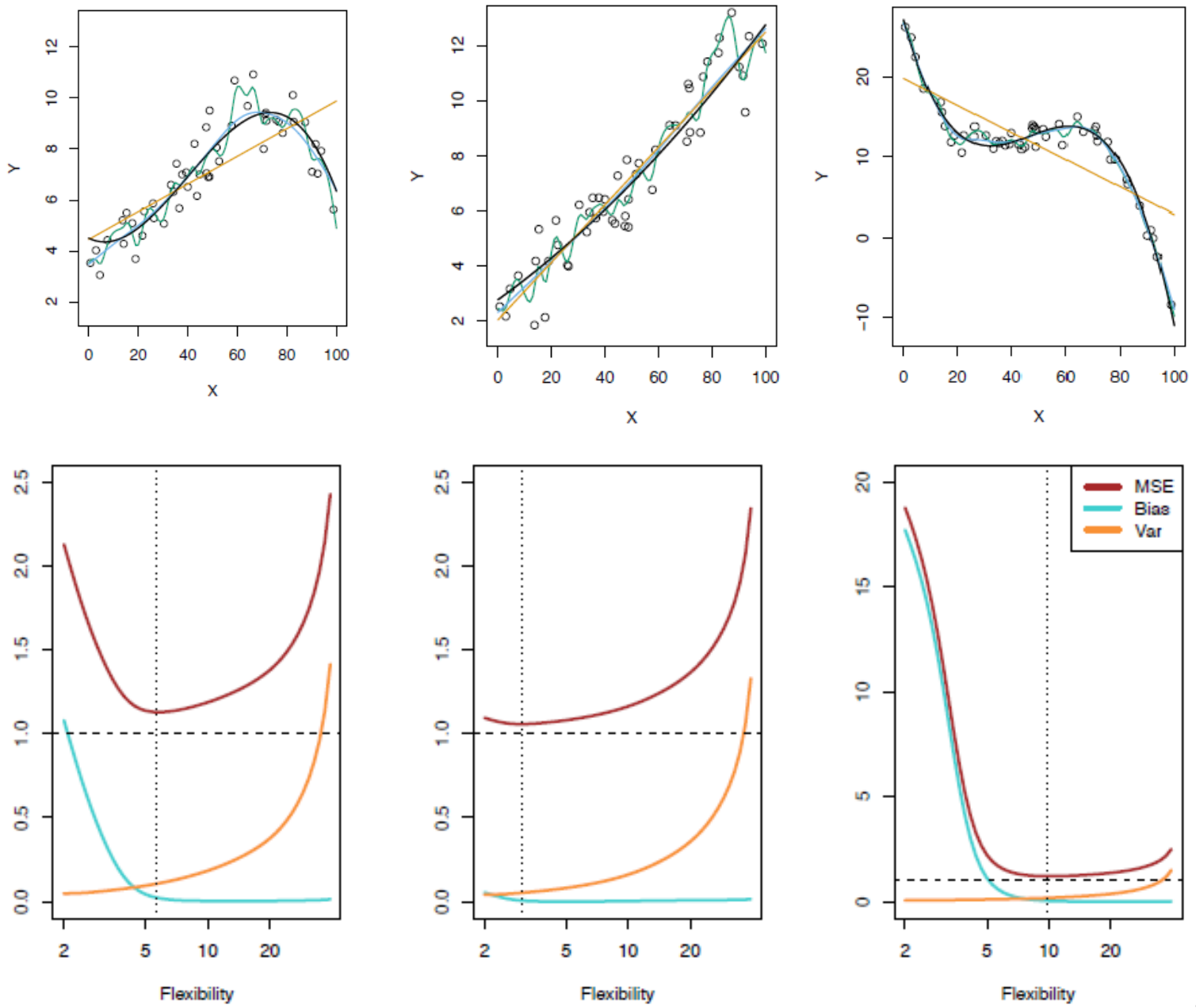
- It can be shown that for any given, $X=x_0$, the expected test MSE for a new Y at x_0 will be equal to

$$Expected\,Test\,MSE = E\left(Y - f(x_0)\right)^2 = Bias^2 + Var + \underbrace{s^2}_{\text{Irreducible Error}}$$

where Bias = E[Y]-f(x) and Var = $E[(Y - E[Y])^2]$

- What this means is that as a method gets more complex
  - the bias will decrease and
  - the variance will likely increase
  - but expected test MSE may go up or down!
- The trade-off is only present if we assume fixed error!
- For some models there may be no trade-off!

Test MSE, bias and variance



Bayes classifier

- In classification, the optimal classification for an instance (x_0, y_0) can be obtained by selecting the class *j* which maximizes the probability

$$P(Y = j | X = x_0)$$

- This classifier is called the Bayes (optimal) classifier
- It implies that learning is actually an estimation of the conditional data distribution

Bayes error rate

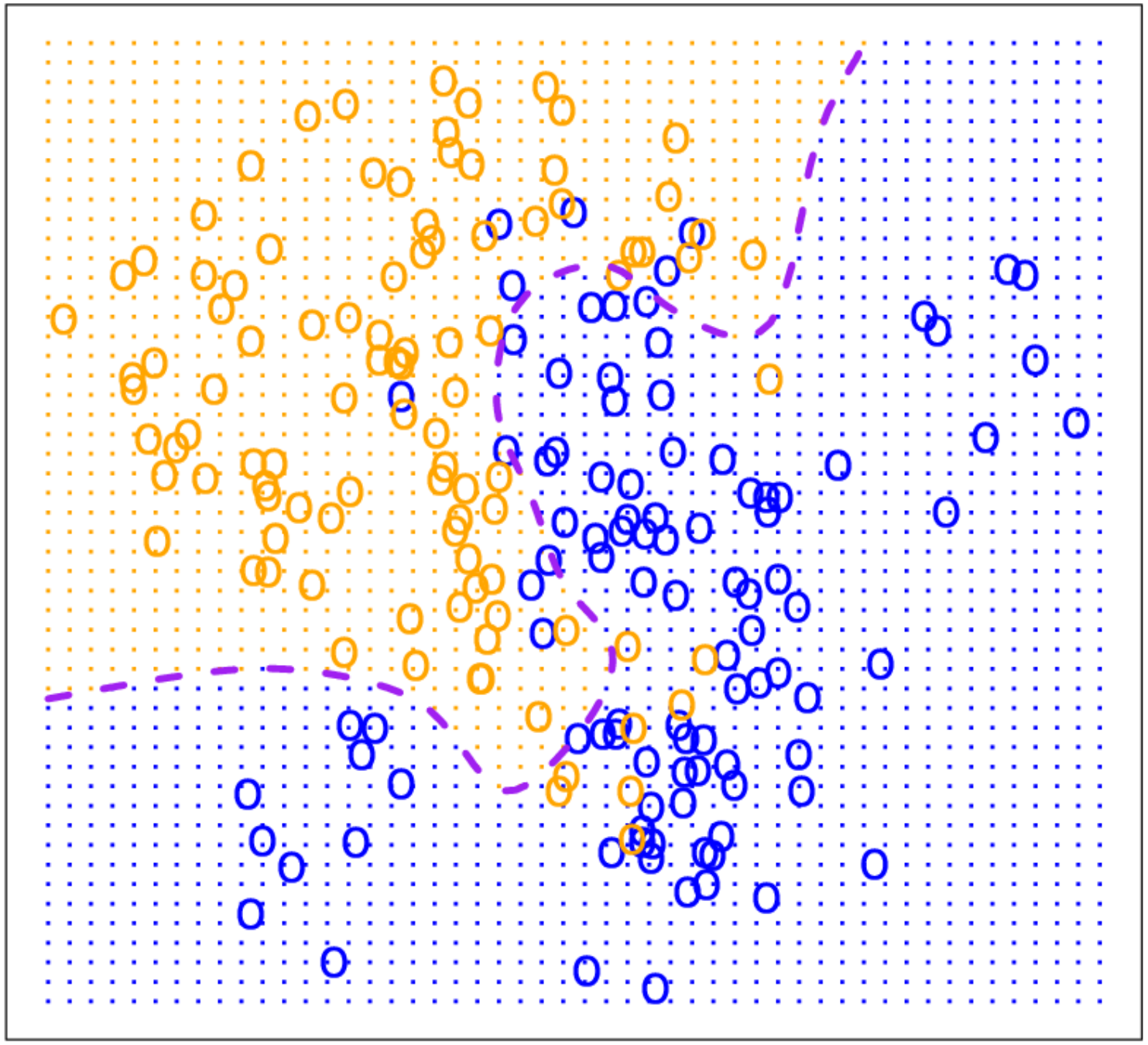
- The Bayes error rate refers to the lowest possible error rate that could be achieved if somehow we knew exactly what the “true” probability distribution of the data looked like.

- On test data, no classifier (or statistical learning method) can get lower error rates than the Bayes error rate.

- Of course, in real life problems, the Bayes error rate can’t be calculated exactly.

Bayes optimal classifier

- for new x_0 returns the maximally probable prediction value $P(Y=y \mid X=x_0)$
- this means to select the class j with $\arg\max_j P(Y=y_j \mid X=x_0)$
- Why is this probability not easy to estimate?
- The dotted line show the Bayes decision boundary, where $P(Y=y \mid X=x_0) = 0.5$



Bayes classifier approximations

- Two models can be viewed as directly approximating the Bayes classifier $P(Y = j | X = x_0)$
- Naive Bayesian classifier
- uses Bayesian formula to get inverse conditional probabilities
- assuming conditional independence between features

$$P(C|X_1X_2 \dots X_n) = \frac{P(C) \cdot P(X_1X_2 \dots X_n|C)}{P(X_1X_2 \dots X_n)} \approx \frac{P(C) \cdot \prod_i P(X_i|C)}{\prod_i P(X_i)}$$

- Nearest neighbour classifier
  - directly estimates the conditional probability using instances near to x_0

K-Nearest Neighbors (KNN)

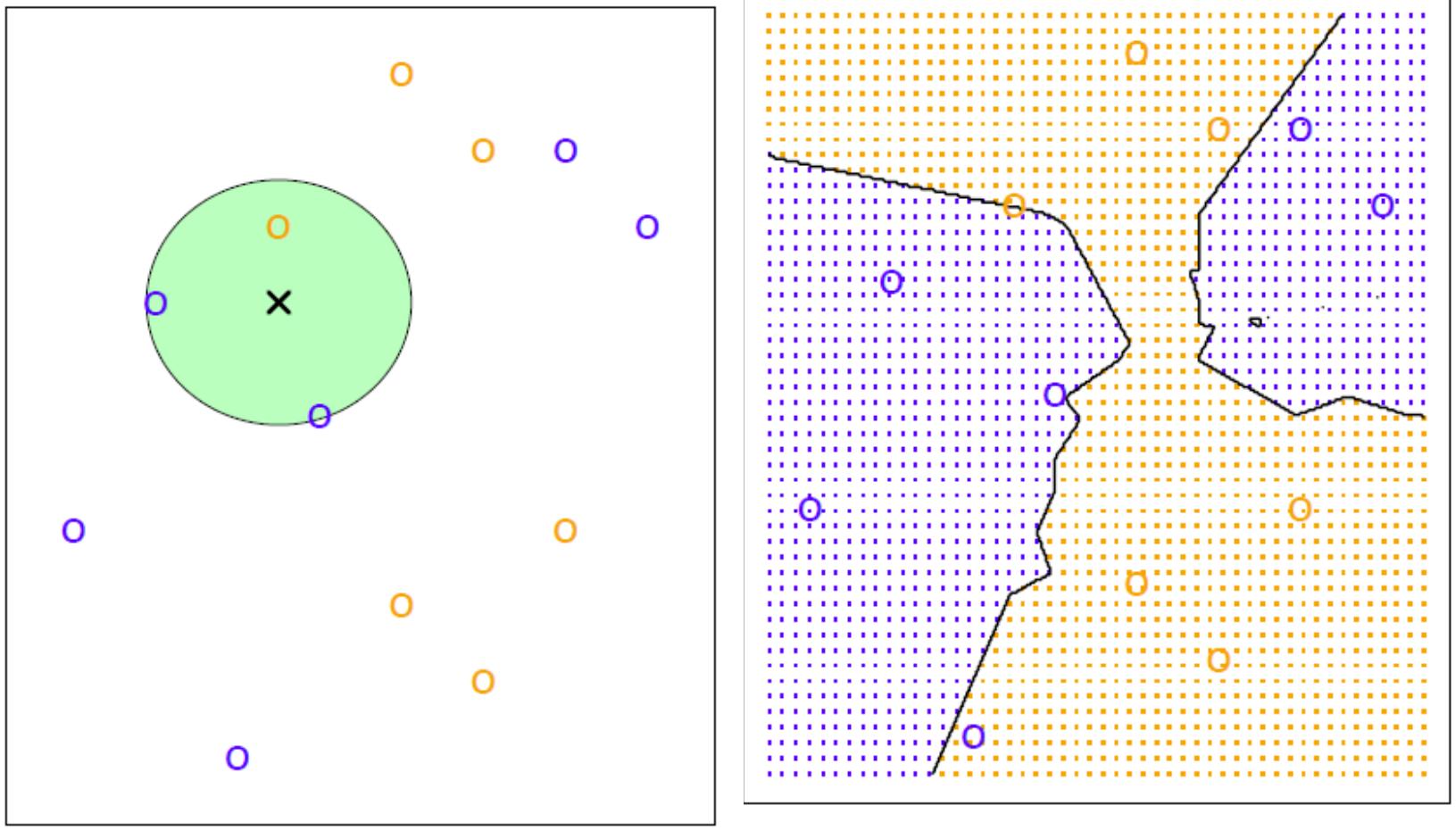
- k Nearest Neighbors is a flexible approach to estimate the Bayes classifier.

- For any given *x* we find the *k* closest neighbors to *x* in the training data, and examine their corresponding *y*.

- If the majority of the *y*'s are orange, we predict the orange label otherwise the blue label.

- The smaller that *k* is the more flexible the method will be.

KNN example with k = 3



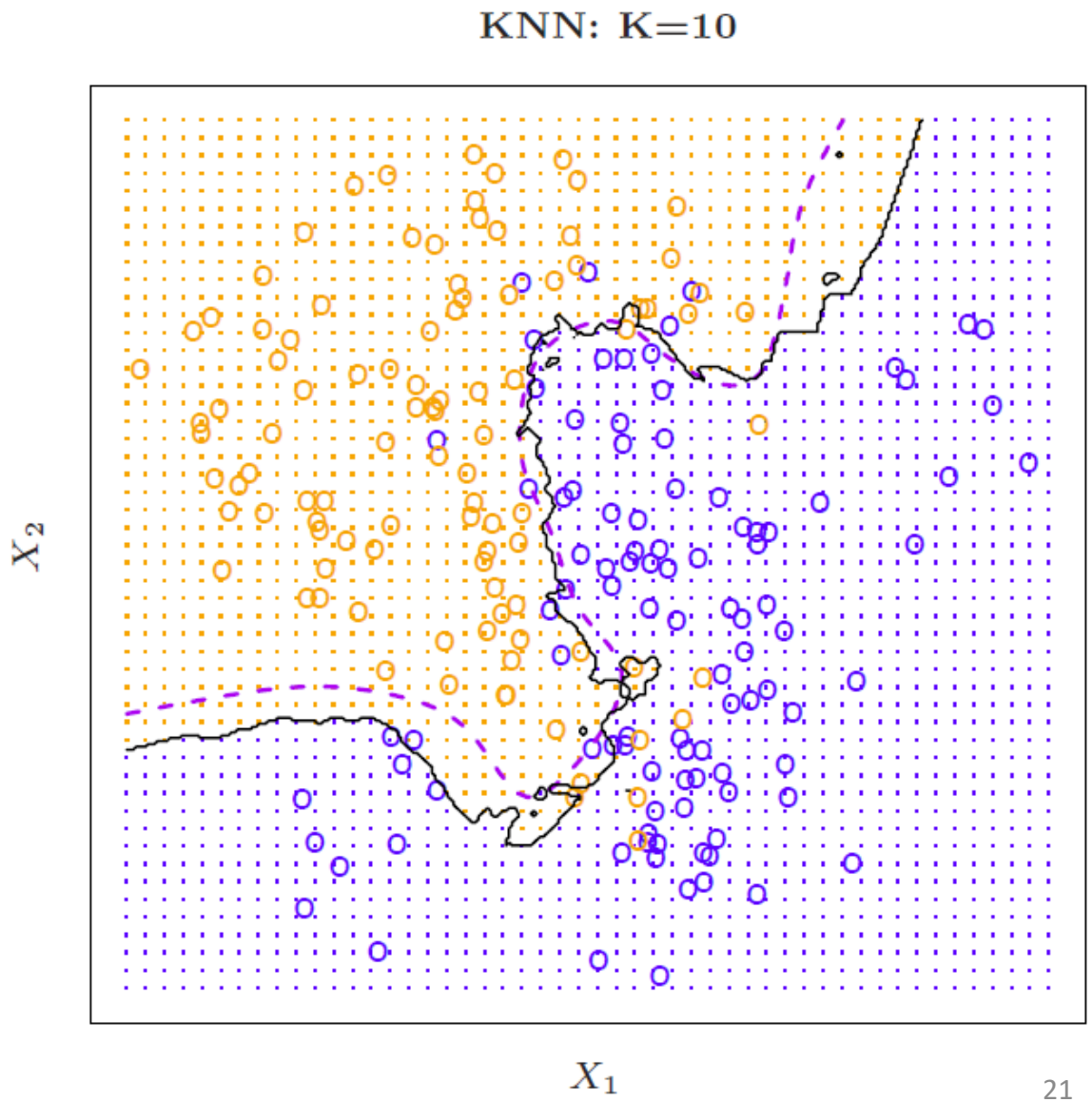
K-NN classifier

- Given a positive integer *K* and a test observation x_0, the KNN classifier first identifies the *K* points in the training data that are closest to x_0, represented by $\mathcal{N}_0$.
- It then estimates the conditional probability for class *j* as the fraction of points in $\mathcal{N}_0$ whose response values equal *j*:

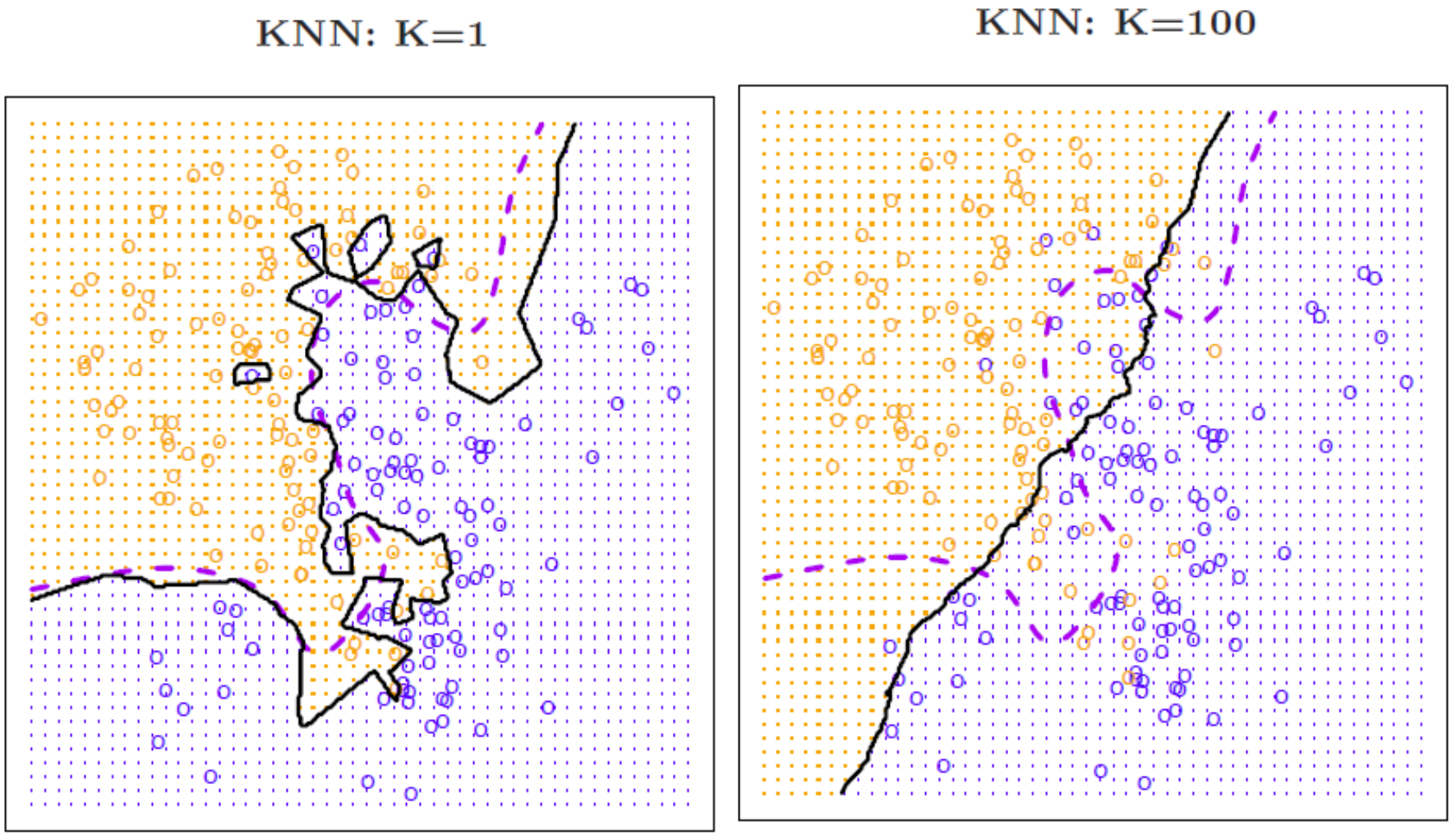
$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j).$$

- applies Bayes rule and classifies the test observation x_0 to the class with the largest probability.

Simulated data:
K = 10

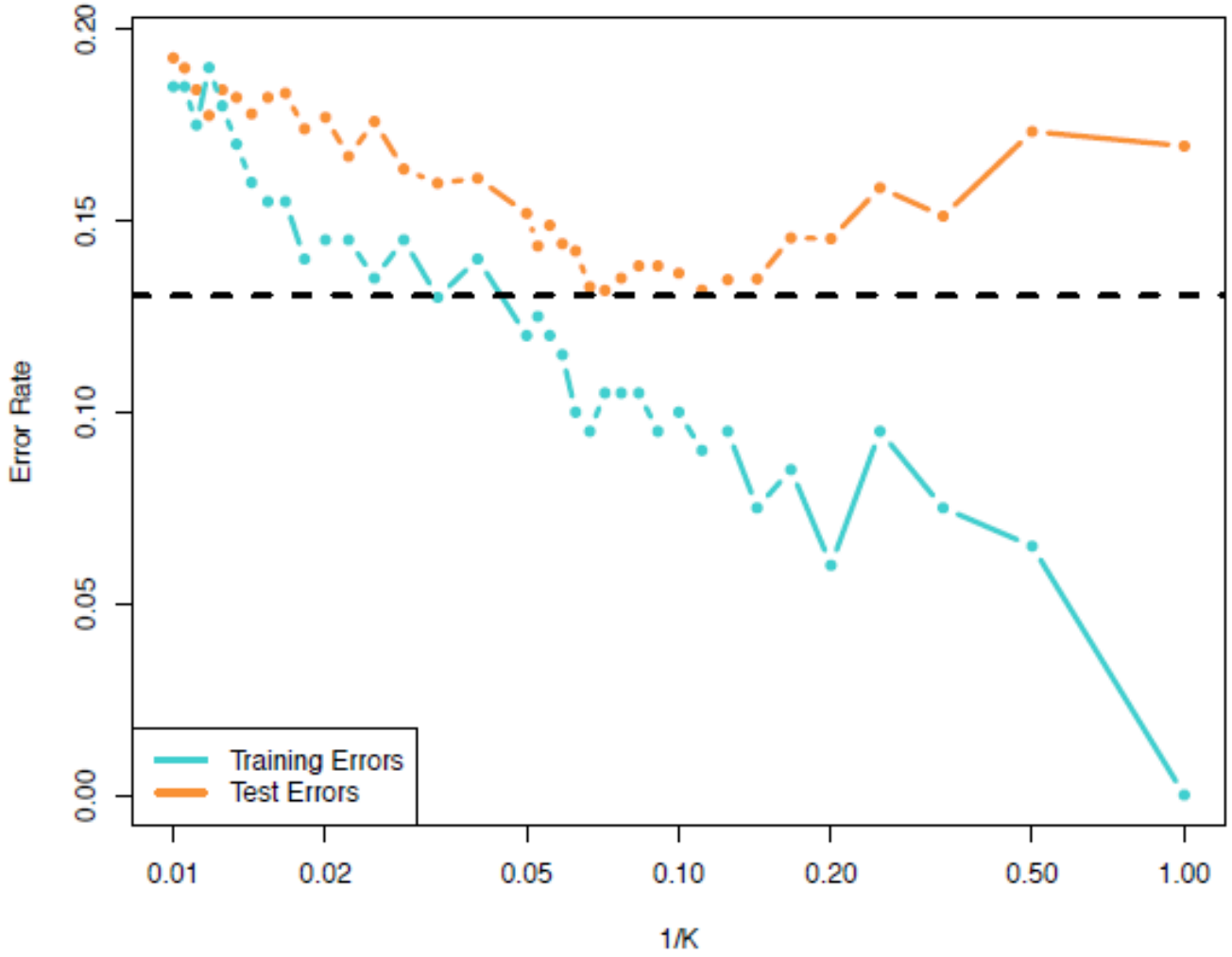


K = 1 and K = 100



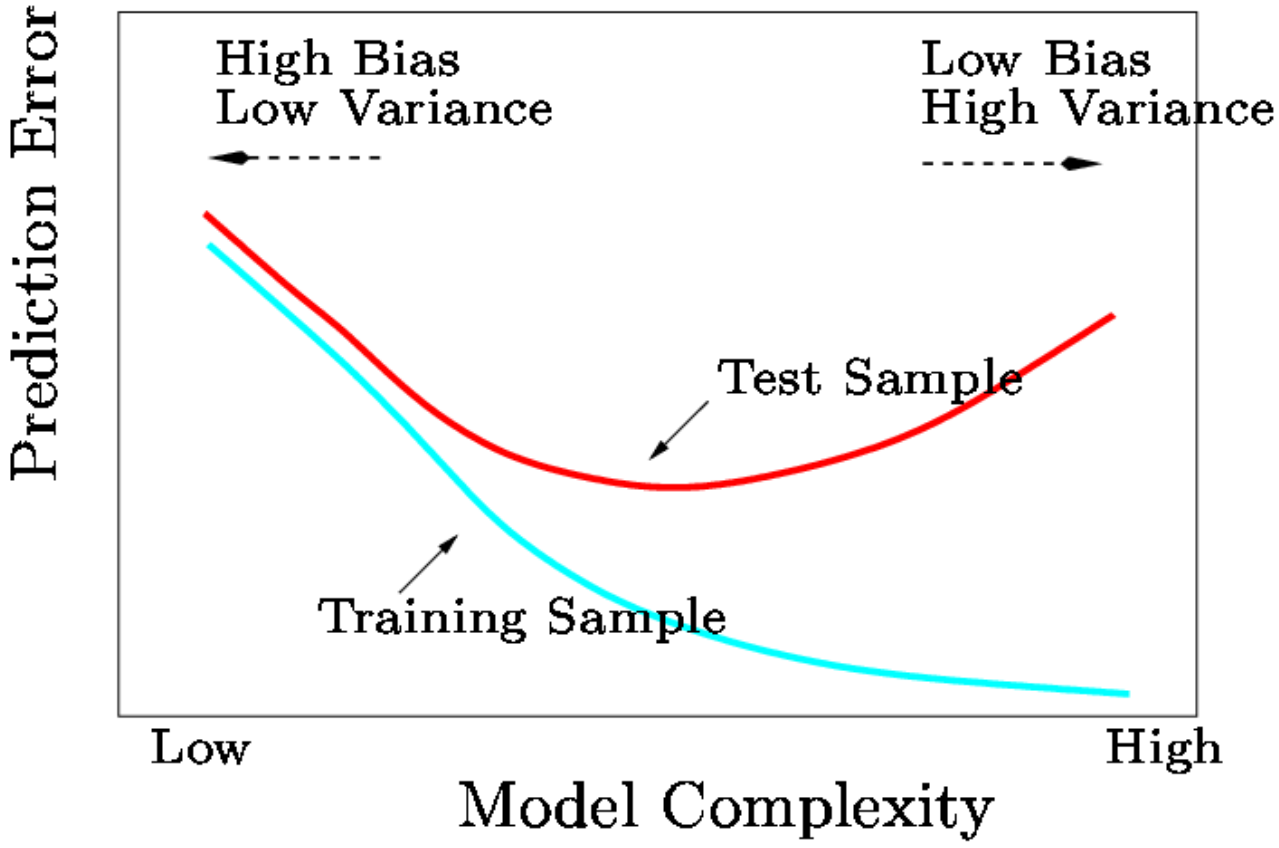
Training vs. test error rates on the simulated data

- Notice that training error rates keep going down as k decreases or equivalently as the flexibility increases.
- However, the test error rate at first decreases but then starts to increase again.



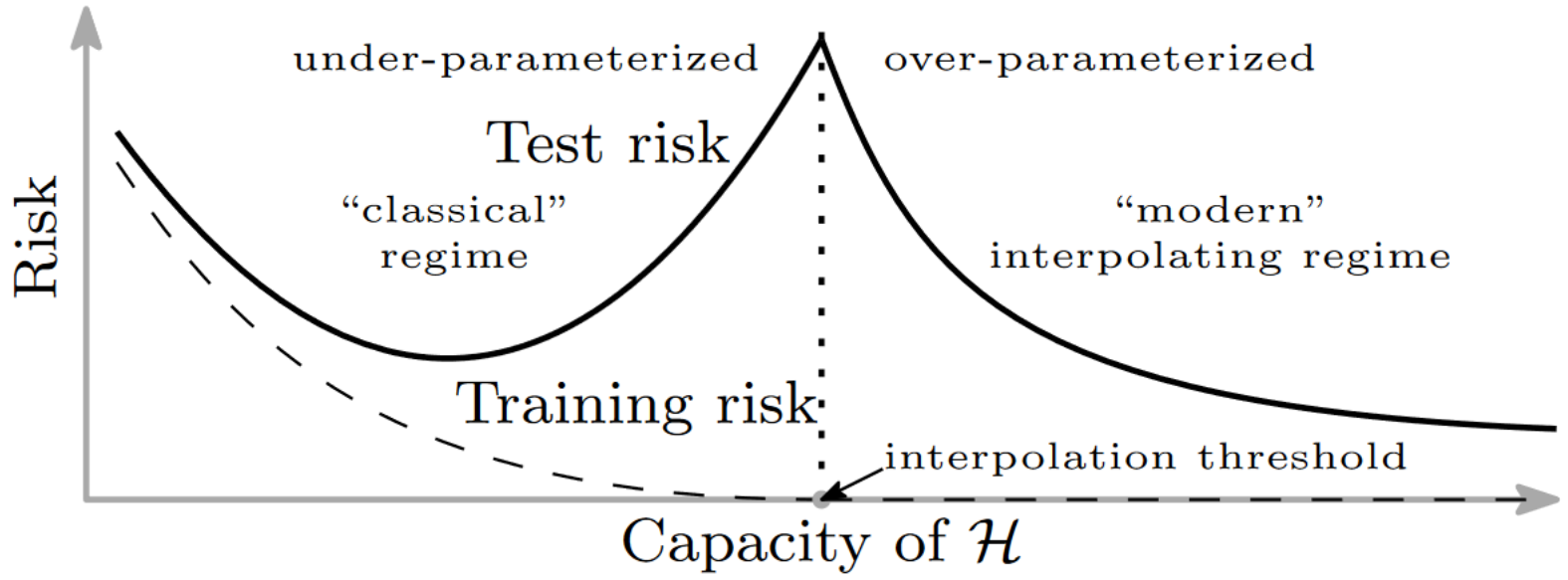
A fundamental picture

- In general training errors will always decline.
- However, test errors will decline at first (as reductions in bias dominate) but will then start to increase again (as increases in variance dominate).
- This is a conventional wisdom, but it is not true for all methods and all training regimes.



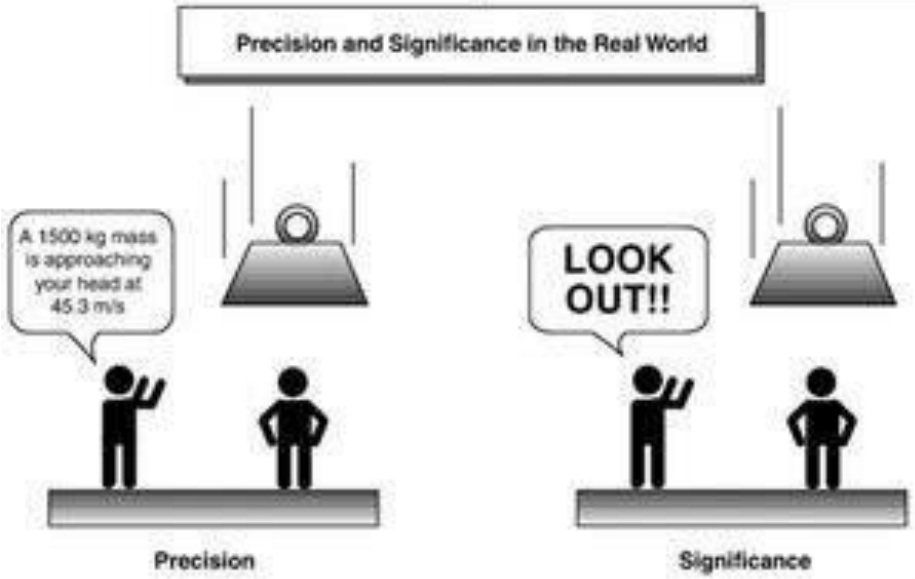
The double descent curve

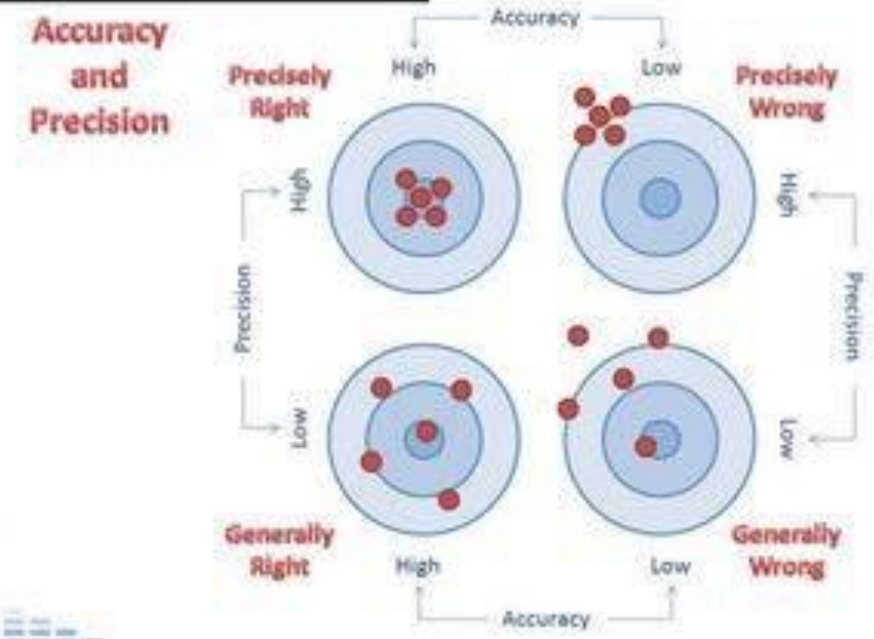
- While for some models, like kNN, there seem to be a trade-off between bias and variance, this is not a universal phenomenon
- E.g., overparametrization in neural networks produce double descent curve (similar evidence for random forests)

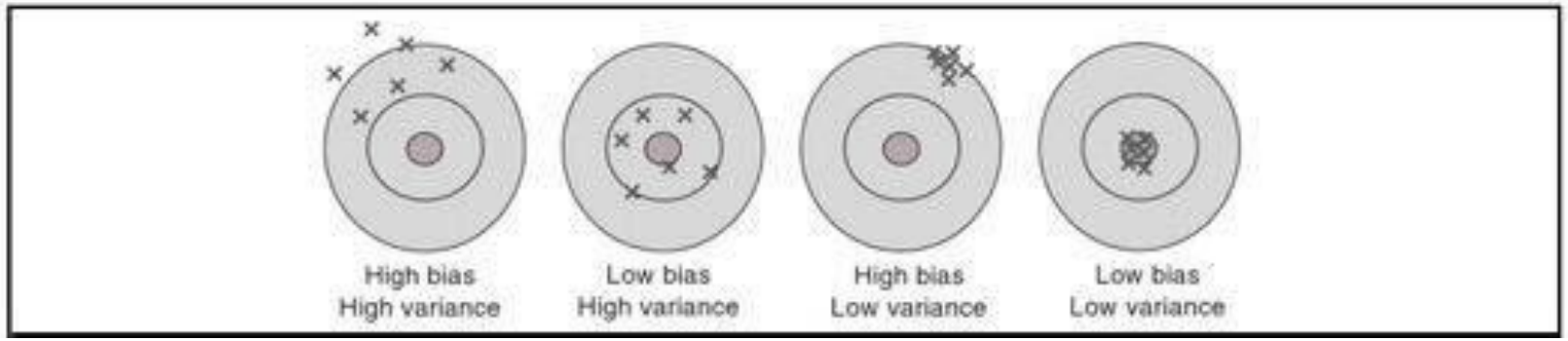


Belkin, M., Hsu, D., Ma, S. and Mandal, S., 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, *116*(32), pp.15849-15854.

Precision vs. Significance
Accuracy vs. Precision
Bias vs. Variance





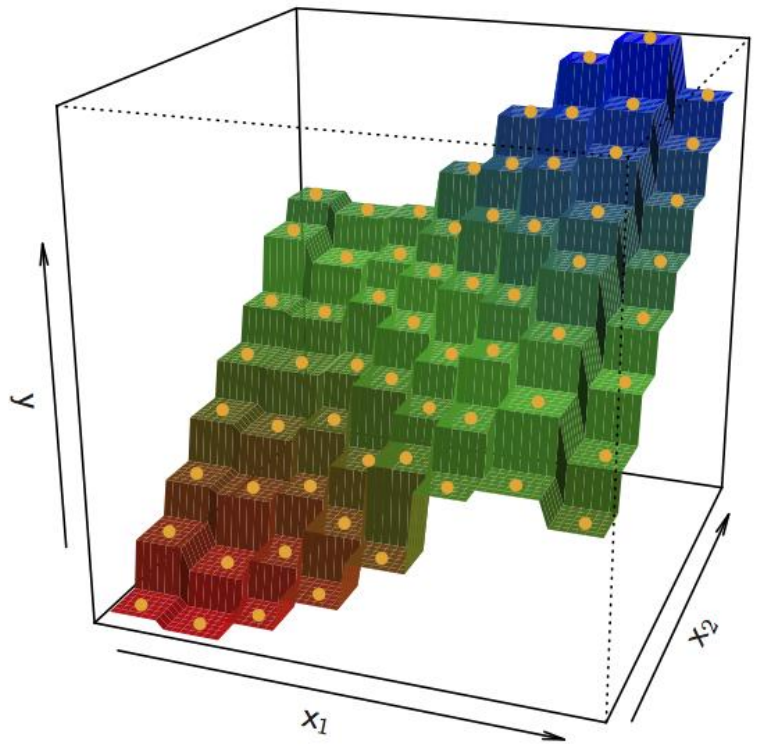


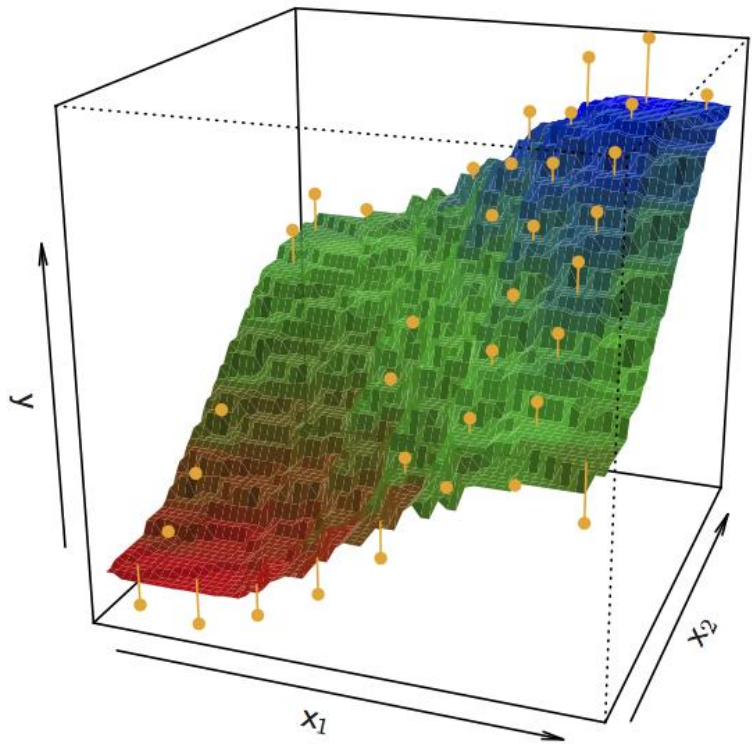
K-nearest neighbor for regression

- kNN regression is similar to the kNN classifier.
- given a set of instances (x_i, y_i)
- To predict *y* for a given value of *x*, consider *k* closest points to *x* in training data and take the average of the responses. i.e.

$$f(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

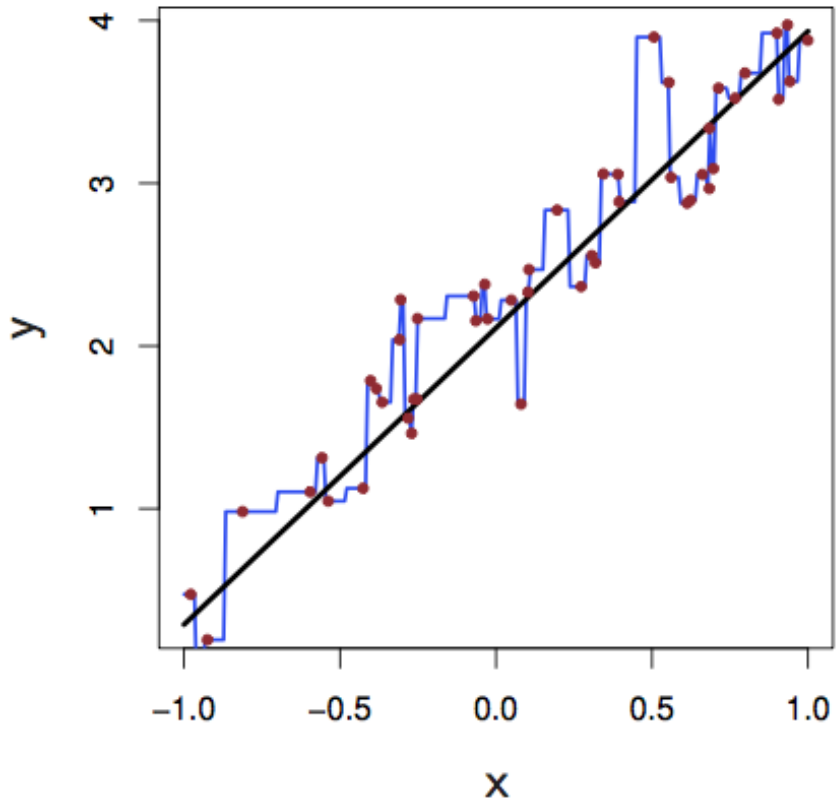
KNN Fits for k =1 and k = 9

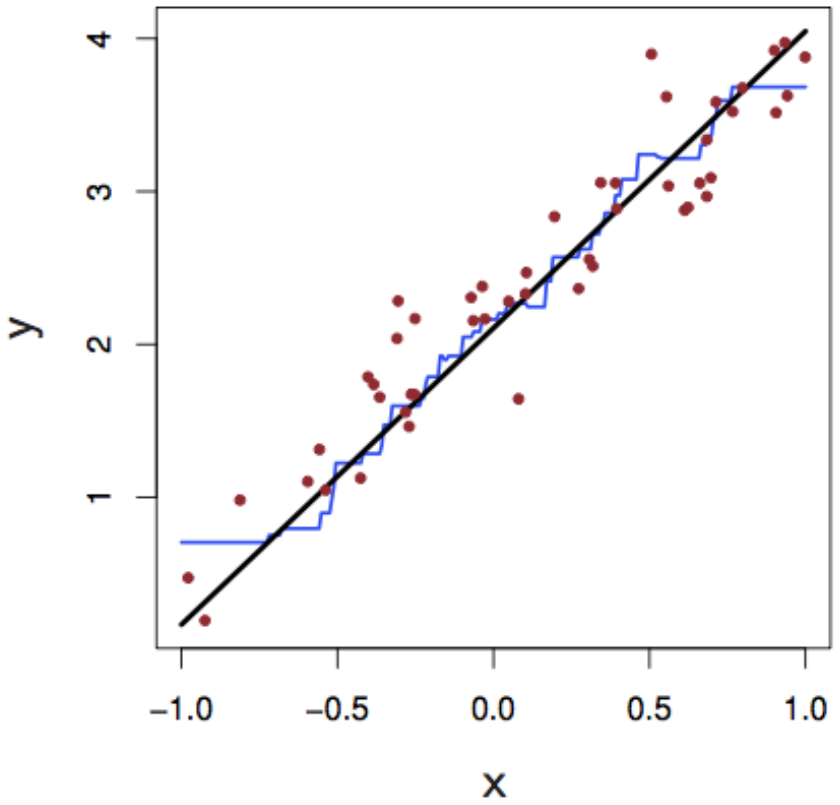




KNN fits in one dimension (k =1 and k = 9)

- black line: actual function,
- blue line: regressional kNN

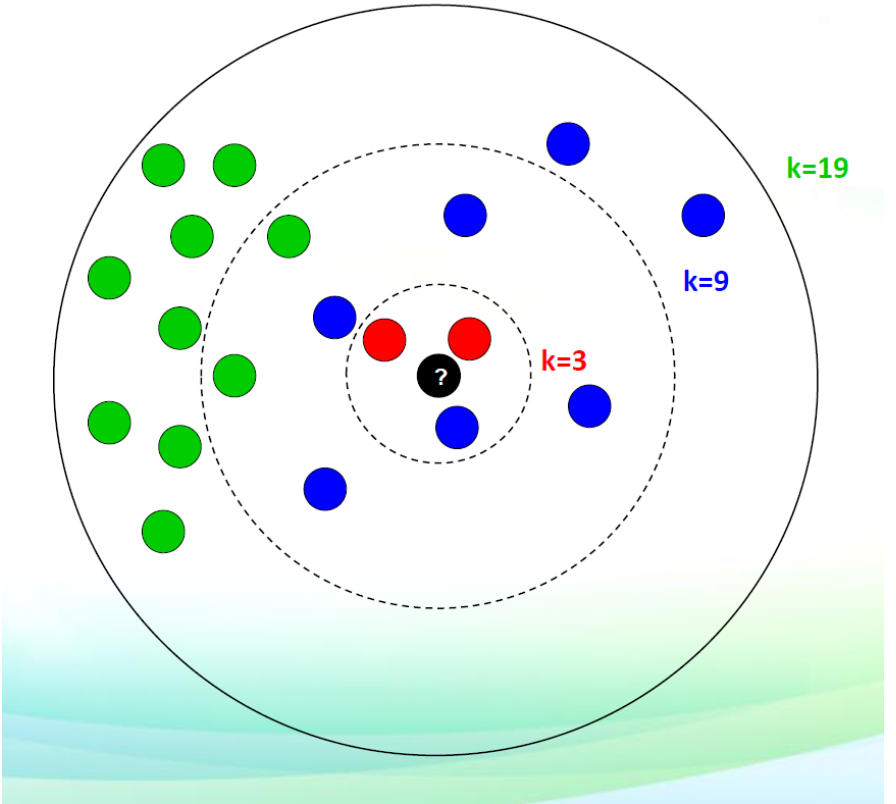




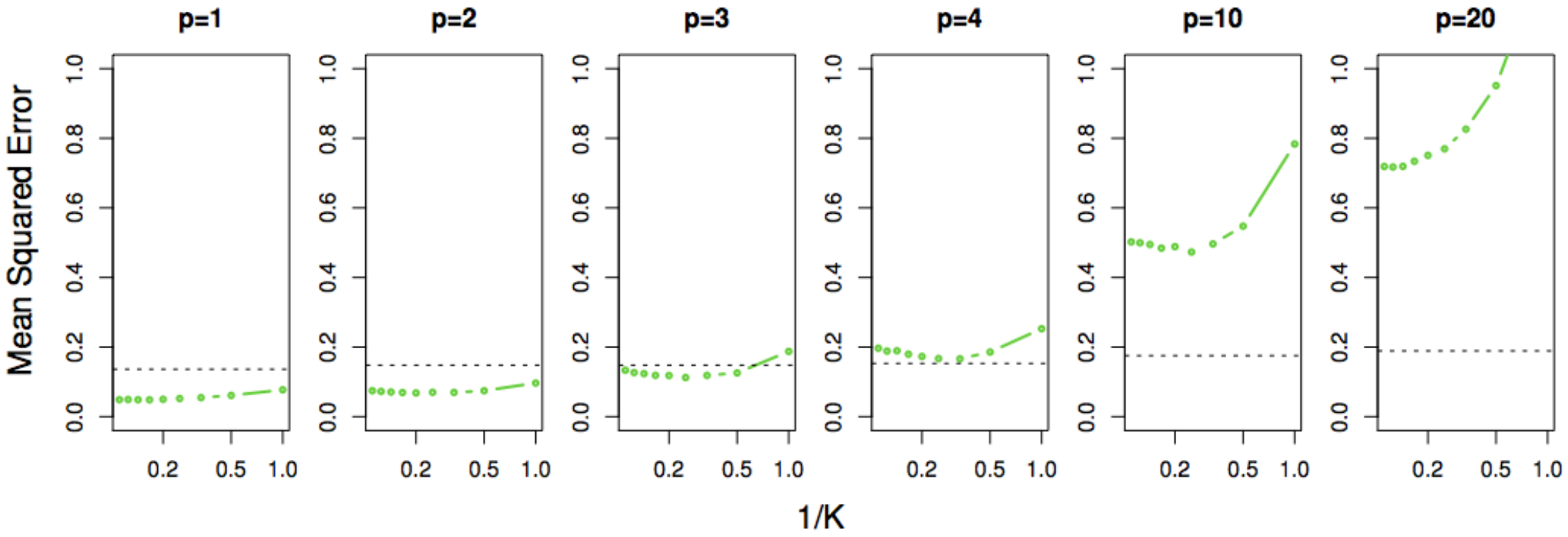
Choice of k in KNN

- If *k* is small, kNN is much more flexible than linear regression.
- Is that better?

- The results may be highly dependent on the choice of *k*.



kNN is not so good in high dimensional situations



- p is the number of dimensions

Speeding up KNN algorithm

- precondition: normalization of dimensions, e.g., to [0, 1]
- naive search for nearest neighbors: $O(n \cdot d \cdot t)$
  - n is number of instances
  - d is number of dimensions
  - t is number of nearest neighbors
- exact search for low dimensional spaces
  - k-d trees (d is around 10)
  - quad-trees (d=2), octrees (d=3)
  - R-tree (rectangular tree, also R+, R*, ...), d=2 or 3
- approximate search
  - RKD-tree (random k-d tree)
  - locally sensitive hashing (LSH),
  - hierarchical k-means

Rule learning

- Using IF-THEN Rules for Classification
- Represent the knowledge in the form of IF-THEN rules

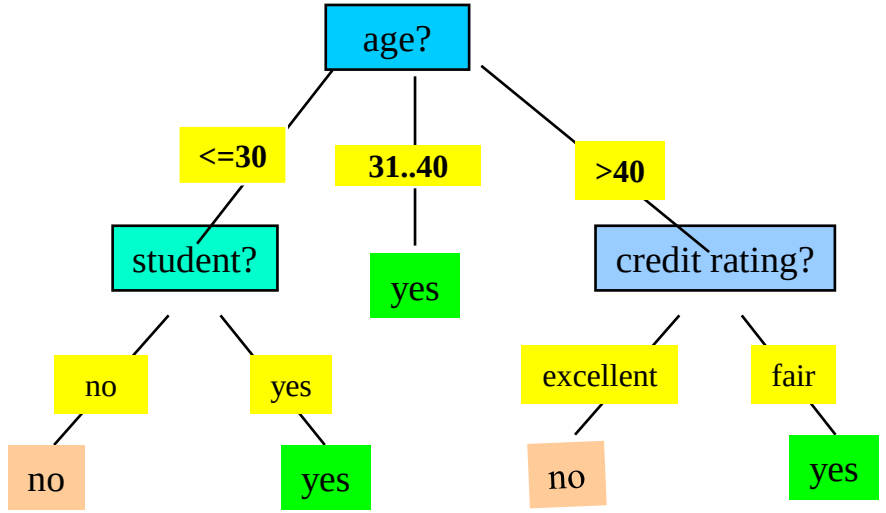
  R: IF *age* = youth AND *student* = yes THEN *buys_computer* = yes
  - Rule antecedent/precondition vs. rule consequent
- Assessment of a rule: *coverage* and *accuracy*
  - n_{covers} = # of instances covered by R
  - $n_{correct}$ = # of instances correctly classified by R

  coverage(R) = n_{covers} /|D| /* D: training data set */

  accuracy(R) = $n_{correct}$ / n_{covers}
- If more than one rule is triggered, we need **conflict resolution**
  - Size ordering: assign the highest priority to the triggering rules that has the "toughest" requirement (i.e., with the *most attribute tests*)
  - Class-based ordering: decreasing order of *prevalence or misclassification cost per class*
  - Rule-based ordering (**decision list**): rules are organized into one long priority list, according to some measure of rule quality or by experts

1st approach: Rule extraction from a decision tree

- Rules are *easier to understand* than large trees
- One rule is created *for each path* from the root to a leaf
- Each attribute-value pair along a path forms a conjunction: the leaf holds the class prediction
- Rules are mutually exclusive and exhaustive



- Example: Rule extraction from the above *buys_computer* decision-tree

IF *age* = young AND *student* = *no* THEN *buys_computer* = *no*

IF *age* = young AND *student* = *yes* THEN *buys_computer* = *yes*

IF *age* = mid-age THEN *buys_computer* = *yes*

IF *age* = old AND *credit_rating* = *excellent* THEN *buys_computer* = *no*

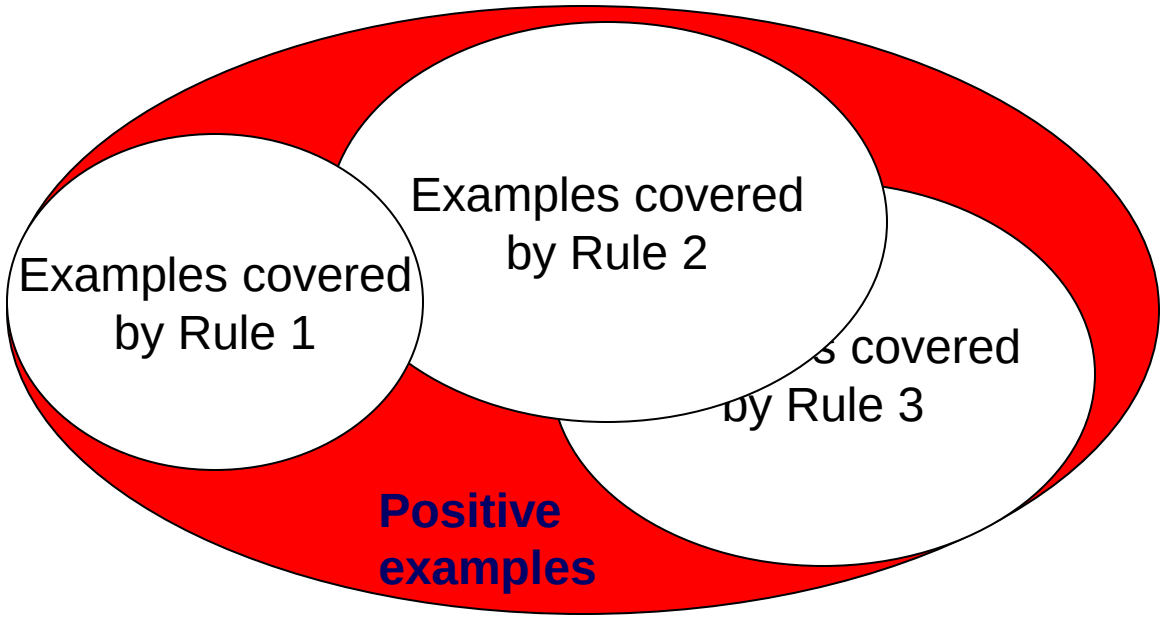
IF *age* = old AND *credit_rating* = *fair* THEN *buys_computer* = *yes*

2nd approach: rule induction, sequential covering method

- Sequential covering algorithm: extracts rules directly from training data
- Typical sequential covering algorithms: FOIL, AQ, CN2, RIPPER
- Rules are learned *sequentially:* a rule for a given class C_i will cover many instances of C_i but none (or few) of the instances of other classes
- Steps:
  - Rules are learned one at a time
  - Each time a rule is learned, the instances covered by the rules are removed
  - Repeat the process on the remaining instances until *termination condition*, e.g., when no more training examples or when the quality of a rule returned is below a user-specified threshold
- Compare with decision-tree induction which produce a set of rules *simultaneously*

Sequential covering algorithm

while (enough target instances left)
generate a rule
remove positive target instances satisfying this rule



Rule generation

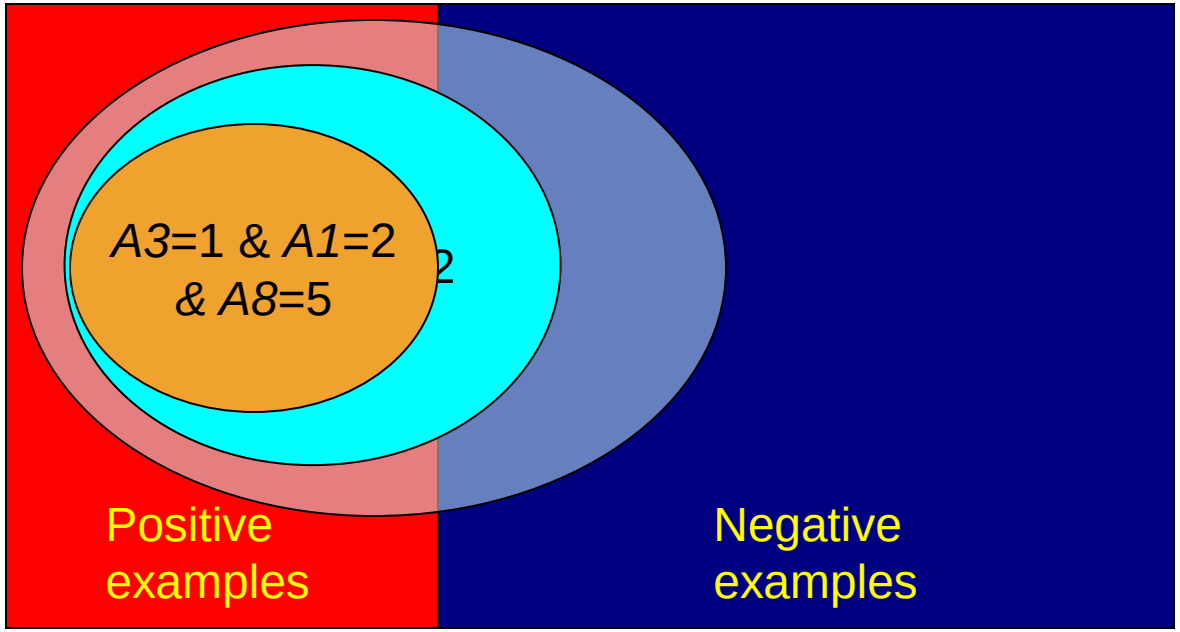
- To generate a rule

  while(true)

    find the best predicate *p*

    if ruleQuality(*p*) > threshold **then** add *p* to current rule

    else break



How to learn one rule?

- Start with the *most general rule* possible: condition = empty
- *Adding new attributes* by adopting a greedy depth-first strategy
  - Picks the one that most improves the rule quality
- Rule-quality measures: consider both coverage and accuracy
- Foil-gain (in FOIL & RIPPER): assesses information gain by extending condition

  Gain(R_0, R_1) := t * ($\log_2(p_1/(p_1+n_1)) - \log_2(p_0/(p_0+n_0))$)
  - R_0 denotes a rule before adding a new condition
  - R_1 is an extension of R_0 after adding the new condition
  - An instance is covered by a rule if it fulfills all of its preconditions.
  - p_0 denotes the number of positive instances, covered by R_0,
  - p_1 the number of positive instances, covered by R_1
  - n_0 and n_1 are the number of negative instances, covered by the according rule.
  - t is the number of positive instances, covered by R_0 as well as by R_1
- Rule pruning based on an independent set of validation instances
- favors rules that have high accuracy and cover many positive instances
- Pos/neg are # of positive/negative tuples covered by Rule
- If *FOIL_Prune* is higher for the pruned version of Rule, prune Rule

$$FOIL\_Prune(Rule) = \frac{pos - neg}{pos + neg}$$

Ethical consideration of bias in ML models

- bias in models:
  - characteristic of models,
  - affects error,
  - unlikely to be ethically problematic
  - when it can be problematic?
- bias in statistics
- bias in algorithms
- bias in data:
  - data unrepresentative of true population,
  - might be ethically problematic

Biases in the data

- Machine learning models are not inherently objective. Engineers train models by feeding them a data set of training examples, and human involvement in the provision and curation of this data can make a model's predictions susceptible to bias.
- When building models, it's important to be aware of common human biases that can manifest in your data, so you can take proactive steps to mitigate their effects.
- The biases listed provide just a small selection of biases that are often uncovered in machine learning data sets; this list *is not intended to be exhaustive*. Wikipedia's catalog of cognitive biases enumerates over 100 different types of human bias that can affect our judgment. When auditing your data, you should be on the lookout for any and all potential sources of bias that might skew your model's predictions.

Reporting bias

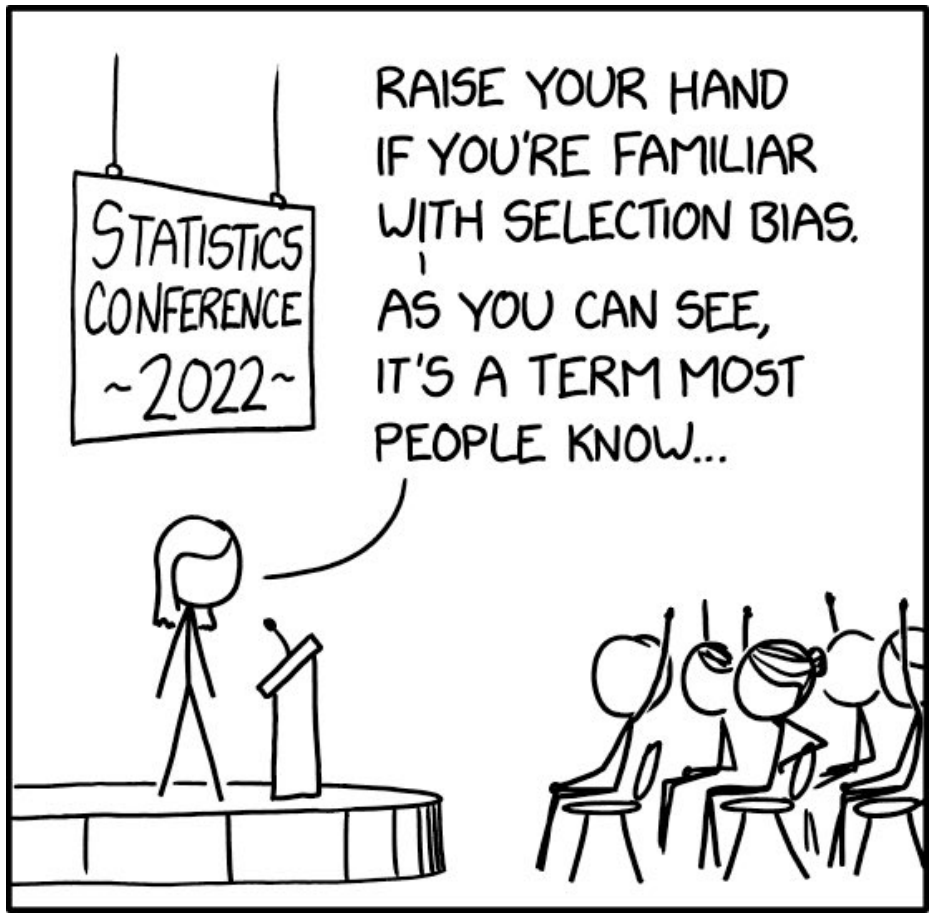
- **Reporting bias** occurs when the frequency of events, properties, and/or outcomes captured in a data set does not accurately reflect their real-world frequency. This bias can arise because people tend to focus on documenting circumstances that are unusual or especially memorable, assuming that the ordinary can "go without saying."
  - **EXAMPLE**: A sentiment-analysis model is trained to predict whether book reviews are positive or negative based on a corpus of user submissions to a popular website. The majority of reviews in the training data set reflect extreme opinions (reviewers who either loved or hated a book), because people were less likely to submit a review of a book if they did not respond to it strongly. As a result, the model is less able to correctly predict sentiment of reviews that use more subtle language to describe a book.

Automation bias

- **Automation bias** is a tendency to favor results generated by automated systems over those generated by non-automated systems, irrespective of the error rates of each (we also have the opposite bias)
  - **EXAMPLE**: Software engineers working for a sprocket manufacturer were eager to deploy the new "groundbreaking" model they trained to identify tooth defects, until the factory supervisor pointed out that the model's precision and recall rates were both 15% lower than those of human inspectors.

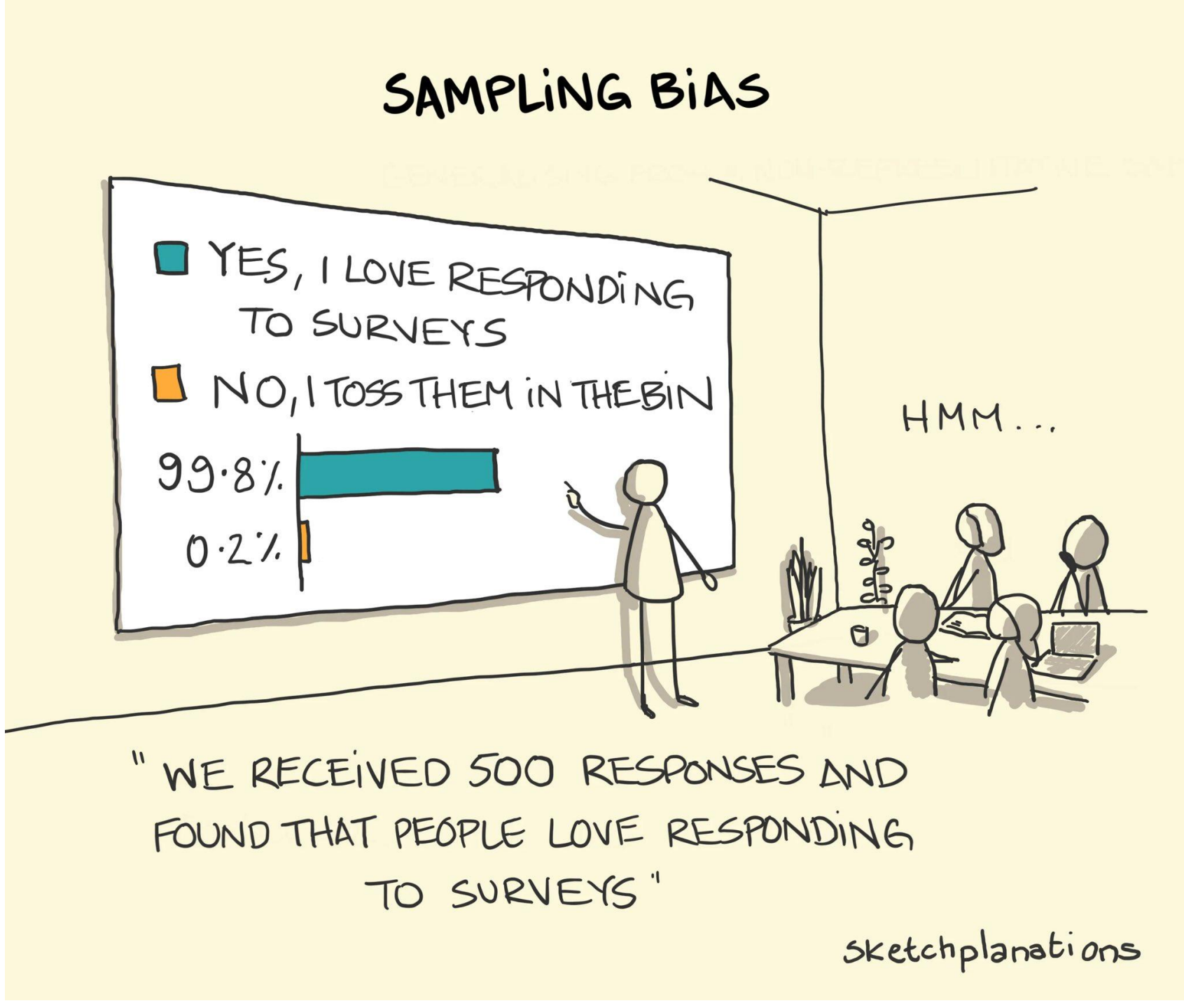
Selection bias

- **Selection bias** occurs if a data set's examples are chosen in a way that is not reflective of their real-world distribution.



Selection bias variants

- Selection bias can take many different forms:
  - **Coverage bias**: Data is not selected in a representative fashion.
    - **EXAMPLE**: A model is trained to predict future sales of a new product based on phone surveys conducted with a sample of consumers who bought the product. Consumers who instead opted to buy a competing product were not surveyed, and as a result, this group of people was not represented in the training data.
  - **Non-response bias** (or **participation bias**): Data ends up being unrepresentative due to participation gaps in the data-collection process.
    - **EXAMPLE**: A model is trained to predict future sales of a new product based on phone surveys conducted with a sample of consumers who bought the product and with a sample of consumers who bought a competing product. Consumers who bought the competing product were 80% more likely to refuse to complete the survey, and their data was underrepresented in the sample.
  - **Sampling bias**: Proper randomization is not used during data collection.
    - **EXAMPLE**: A model is trained to predict future sales of a new product based on phone surveys conducted with a sample of consumers who bought the product and with a sample of consumers who bought a competing product. Instead of randomly targeting consumers, the surveyor chose the first 200 consumers that responded to an email, who might have been more enthusiastic about the product than average purchasers.


SAMPLING BIAS
YES, I LOVE RESPONDING TO SURVEYS
NO, I TOSS THEM IN THE BIN
99.8%
0.2%
HMM...
" WE RECEIVED 500 RESPONSES AND FOUND THAT PEOPLE LOVE RESPONDING TO SURVEYS "
sketchplanations

Group attribution bias

- **Group attribution bias** is a tendency to generalize what is true of individuals to an entire group to which they belong. Two key manifestations of this bias are:
  - **In-group bias**: A preference for members of a group to which *you also belong*, or for characteristics that you also share.
    - **EXAMPLE**: Two engineers training a resume-screening model for software developers are predisposed to believe that applicants who attended the same computer-science academy as they both did are more qualified for the role.
  - **Out-group homogeneity bias**: A tendency to stereotype individual members of a group to which *you do not belong*, or to see their characteristics as more uniform.
    - **EXAMPLE**: Two engineers training a resume-screening model for software developers are predisposed to believe that all applicants who did not attend a computer-science academy do not have sufficient expertise for the role.

Implicit bias

- **Implicit bias** occurs when assumptions are made based on one's own mental models and personal experiences that do not necessary apply more generally. We are often not aware of these biases and some may be contrary to our conscious beliefs.
  - **EXAMPLE**: An engineer training a gesture-recognition model uses a head shake as a feature to indicate a person is communicating the word "no." However, in some regions of the world, a head shake actually signifies "yes." A common form of implicit bias is **confirmation bias**, where model builders unconsciously process data in ways that affirm preexisting beliefs and hypotheses. In some cases, a model builder may actually keep training a model until it produces a result that aligns with their original hypothesis; this is called **experimenter's bias**.
  - **EXAMPLE**: An engineer is building a model that predicts aggressiveness in dogs based on a variety of features (height, weight, breed, environment). The engineer had an unpleasant encounter with a hyperactive toy poodle as a child, and ever since has associated the breed with aggression. When the trained model predicted most toy poodles to be relatively docile, the engineer retrained the model several more times until it produced a result showing smaller poodles to be more violent.

Implicit Bias is...

Attitudes, Stereotypes, & Beliefs

that can affect how we treat others

based on categorizations such as...

Race

Ability

Gender

Culture

Language

Implicit bias runs contrary to our stated beliefs. We can say that we believe in equity (and truly believe it). But then unintentionally behave in ways that are biased and discriminatory.

Adapted from @DrJRMarcelin