Statistical Learning Overview

DSA 6000: Data Science and Analytics, Fall 2019

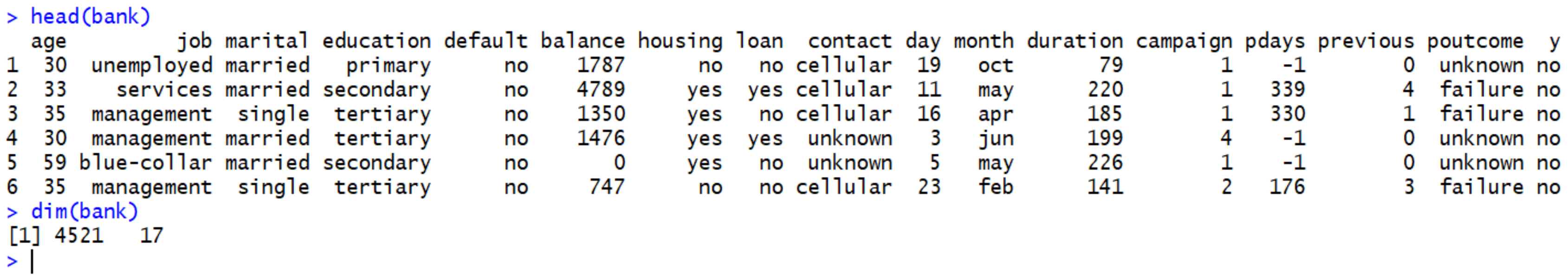
Wayne State University

Supervised learning problem

- Outcome measurement Y, also called dependent variable, response, target.
- Vector of p predictor measurements X, also called inputs, regressors, covariates, features, independent variables.
- In the regression problem, Y is quantitative, e.g., price, revenue, blood pressure, etc.
- In the classification problem, Y takes values in a finite, unordered set, e.g., {survived, died}, {0, 1, 2, …, 9}, {yes, no}
- For an arbitrary observation i, the response is represented by y_i and the input data forms a vector $X_i = (x_{i1}, x_{i2}, \dots, x_{ip})$.
- Training data: $(X_1, y_1), \dots, (X_n, y_n)$.

Example training data

- https://archive.ics.uci.edu/ml/datasets/Bank%2BMarketing



```
> head(bank)
  age         job marital education default balance housing loan  contact day month duration campaign pdays previous poutcome  y
1  30  unemployed married   primary      no    1787      no   no cellular  19   oct       79        1    -1        0  unknown no
2  33    services married secondary      no    4789     yes  yes cellular  11   may      220        1   339        4  failure no
3  35  management  single  tertiary      no    1350     yes   no cellular  16   apr      185        1   330        1  failure no
4  30  management married  tertiary      no    1476     yes  yes  unknown   3   jun      199        4    -1        0  unknown no
5  59 blue-collar married secondary      no       0     yes   no  unknown   5   may      226        1    -1        0  unknown no
6  35  management  single  tertiary      no     747      no   no cellular  23   feb      141        2   176        3  failure no
> dim(bank)
[1] 4521   17
> |
```

- Is Y quantitative or qualitative?
- What are the values for p and n in this training dataset?
- Of what data type is each attribute X_j?

Relationship between variables

- In a physical world under perfect condition: let d_0 be the displacement of a particle from the origin at time $t = 0$ and v be the velocity, then the displacement at time t is $d_t = d_0 + vt$.
- This is a **deterministic** linear relationship between d_t and t, and it predicts d_t perfectly.
- However, in many situations, the relationship between variables is not deterministic.
  - Electric consumption of a house (Y) and the size of the house (X)
  - Fuel usage of an automobile (Y) and the vehicle weight (X)
  - Salary (Y) and education level (X)
- Value of Y cannot be predicted perfectly from knowledge of the corresponding X.
- In statistical learning, we deal with **nondeterministic** relationships.

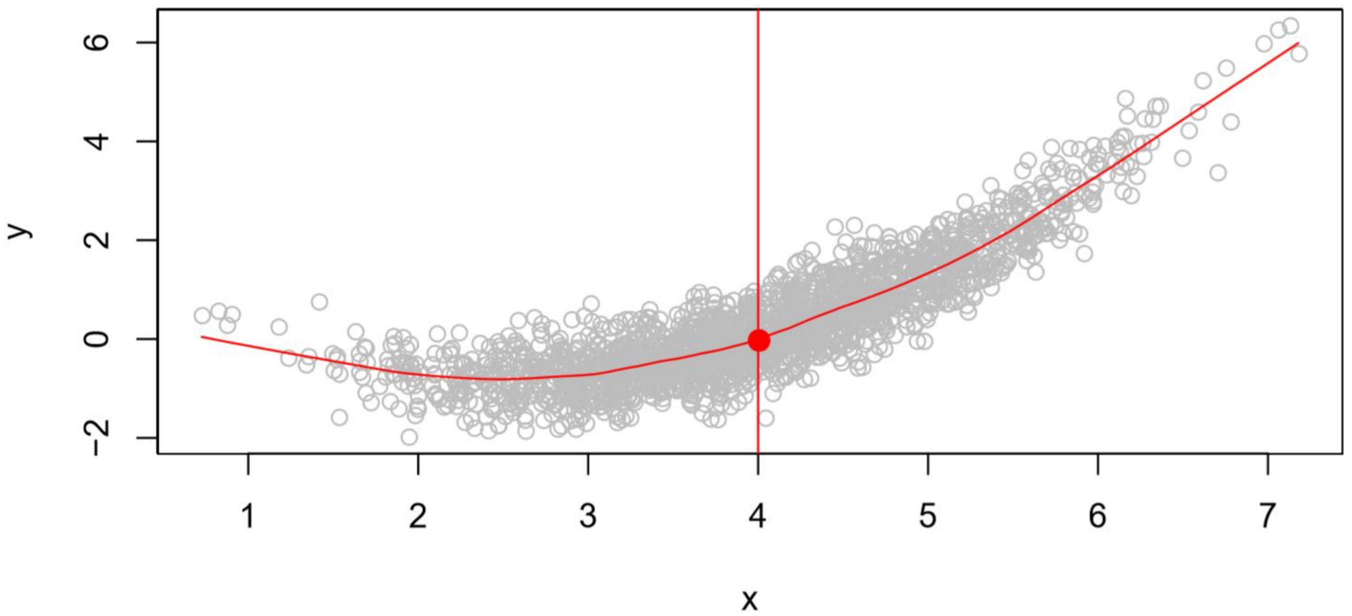
Model and assumption

- We assume that the response Y is a random variable, given by
  - $Y = f(X) + \epsilon$
  - f is a deterministic (i.e., fixed) but unknown function of the input X; it represents the *systematic information* that X provides about Y.
  - ϵ is a random variable with mean zero.
- Statistical learning refers to a set of approaches for ***estimating f*** from data.
  - *Estimation* is a statistical term. What is being estimated is always a *fixed but unknown* thing, usually called a *parameter*. We never estimate a random variable.
  - Unless in simulated cases, we never know the true value of the parameter (in this case, f), but we can quantify the accuracy of our estimation by statistical theory.

Reducible and irreducible errors

- Suppose we have obtained an estimate of f, call it $\hat{f}$.
- Given a new input vector X, we can make a prediction, denoted by $\hat{Y}$, for the value of Y by
    - $\hat{Y} = \hat{f}(X)$
- The accuracy of $\hat{Y}$ as a prediction of Y depends on two quantities
    - **Reducible error**, contributed by $\hat{f}$.
        - Since $\hat{f}$ is not the true f, there is necessarily some error in it.
        - This error can be made smaller (hence reducible) by improving $\hat{f}$.
    - **Irreducible error**, contributed by ϵ.
        - ϵ for sure exists (hence irreducible) by our model assumption, i.e., $Y = f(X) + \epsilon$
        - Even if we knew the true f and made a prediction by $\hat{Y} = f(X)$, there would still be an random error ϵ between Y and $\hat{Y}$

Data perspective



- Our model assumption $Y = f(X) + \epsilon$ with $E(\epsilon) = 0$ leads to $f(X) = E(Y|X)$
  - $f(X)$ is called the regression function, which is fixed and unknown.

- Given any value for X, say $X = 4$, Y is a random variable (along the vertical line). $f(X)$ is the expected value of Y corresponding to the value of X, ,the red dot (its precise vertical location is actually unknown, since the data points observed/plotted are only a sample).

Why estimate f?

- Two main reasons: **prediction** and **inference**
- Prediction: the input vector X is easy to obtain, the output Y is not
  - X: the characteristics of a patient’s blood sample; Y: risk of severe adverse reaction to a particular drug
  - In predictive applications, f can be a black box. We care about the prediction accuracy, not f itself.
- Inference: seek to understand how Y changes as a function of X
- Depending on the main purpose (prediction or inference), different methods for estimating f may be appropriate
  - Linear models are highly interpretable, good for inference
  - SVM are quite accurate in many cases, but less interpretable

Prediction Accuracy and Model Interpretability

- A simple model is usually easier to interpret, e.g., a linear model.
- A simple model may yield low accuracy (even on the training set) for a complex problem, due to underfitting.
- A complex model is not necessarily more accurate than a simpler model (due to overfitting). We are talking about accuracy on the test data, not training data.
- We often prefer a simpler model involving fewer parameters over a black-box model involving many parameters.
- *Occam's razor principle*: in explaining a thing no more assumptions should be made than are necessary.
- *"Everything should be made as simple as possible, but not simpler."*
  -- Albert Einstein (unconfirmed)

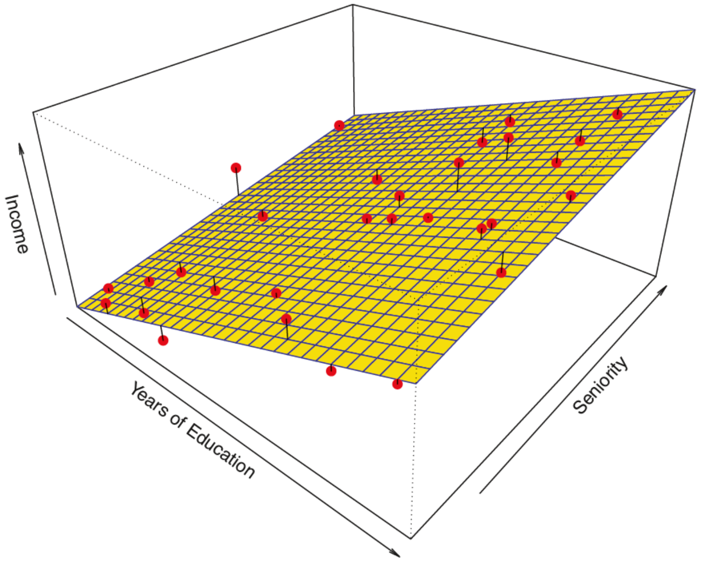
Optional further reading on accuracy v.s. interpretability

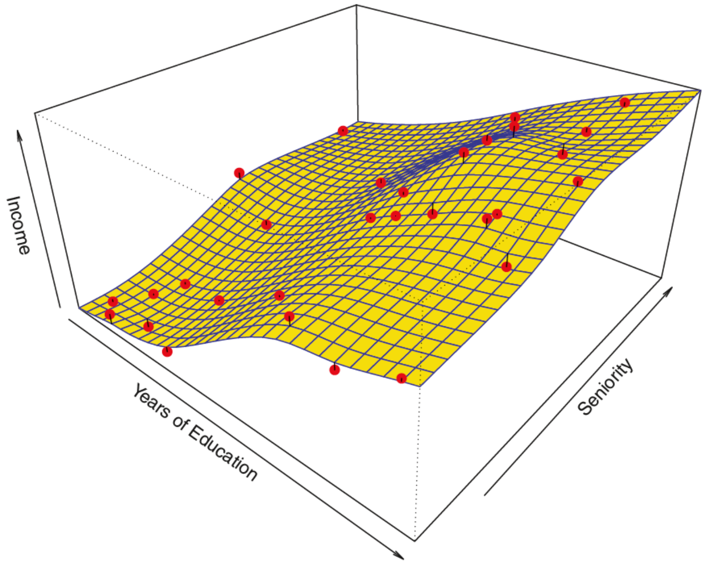
- *Predictive modeling: Striking a balance between accuracy and interpretability* https://www.oreilly.com/ideas/predictive-modeling-striking-a-balance-between-accuracy-and-interpretability
- *Balance: Accuracy vs. Interpretability in Regulated Environments* https://www.elderresearch.com/blog/predictive-model-accuracy-versus-interpretability
- And the references thereof.

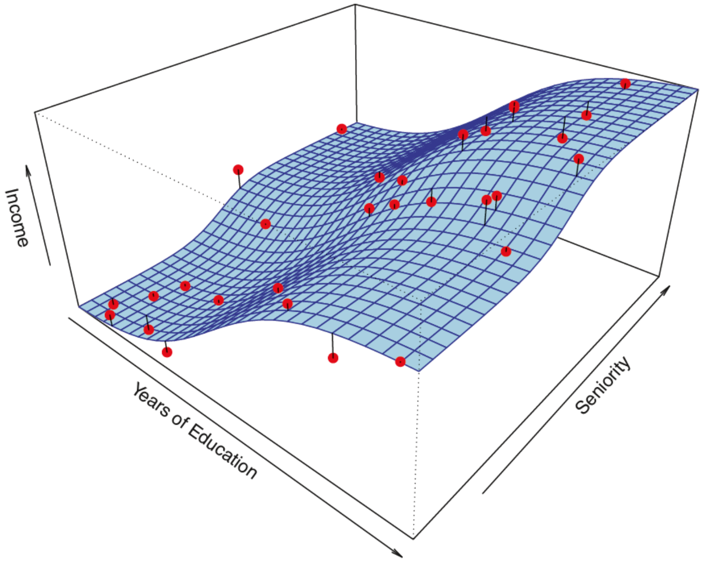
How to estimate f?

- It is the main task of statistical learning.
- **Parametric** methods: assume a functional form of f, then estimate the parameters
  - E.g., Linear model: $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$
    - Reduces the task of estimating an entirely arbitrary p-dimensional function to estimating only $p+1$ coefficients
  - The assumption may be too strong for the real world.
- **Non-parametric** methods: make no assumption about the functional form of f, instead seek an estimate of f that gets as close to the data points as possible, with some constraints (e.g., on smoothness, complexity, etc.)
  - E.g., Splines, Decision Trees

Comparisons

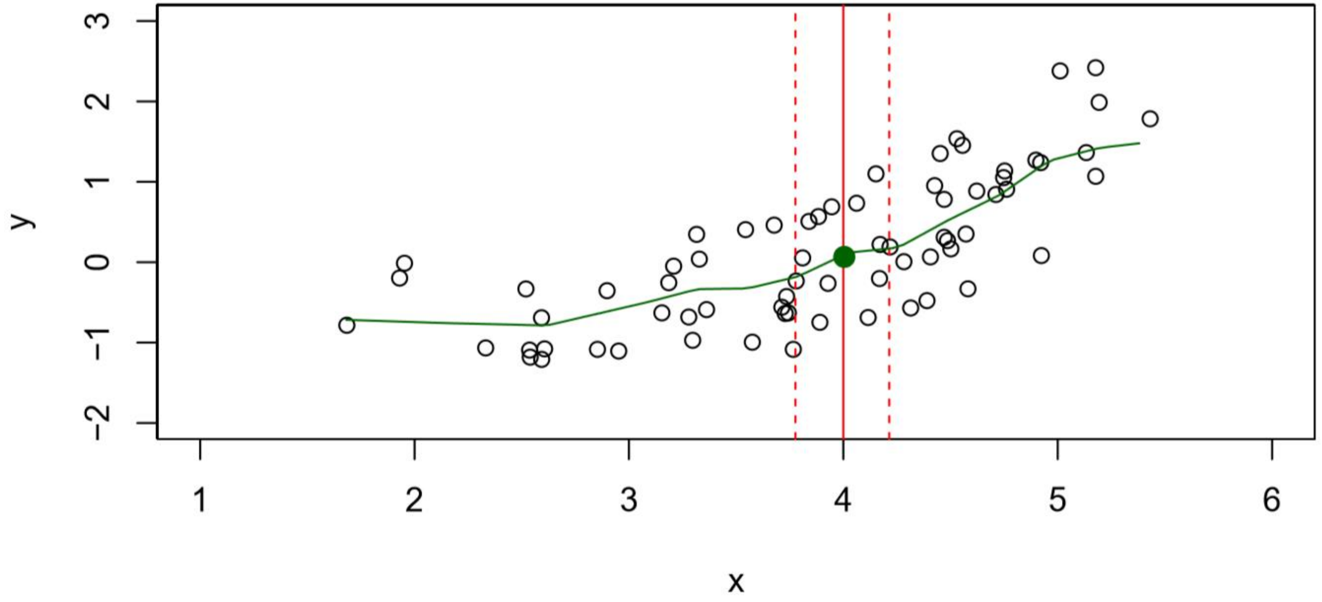






- Upper left: Linear fit
- Upper right: Smoothing spline
- Left: true f, known since the data is simulated

How to estimate f?



- Typically we have few if any data points with $X = 4$ exactly, so we cannot compute $E(Y|X = 4)$.
- One way is via nearest neighbor averaging:
  - $\hat{f} = \text{Avg}(Y|X \in N(x))$ where $N(x)$ is some neighborhood of x.
  - For high dimensional data, suffers from curse of dimensionality: nearest neighbors tend to be far away.

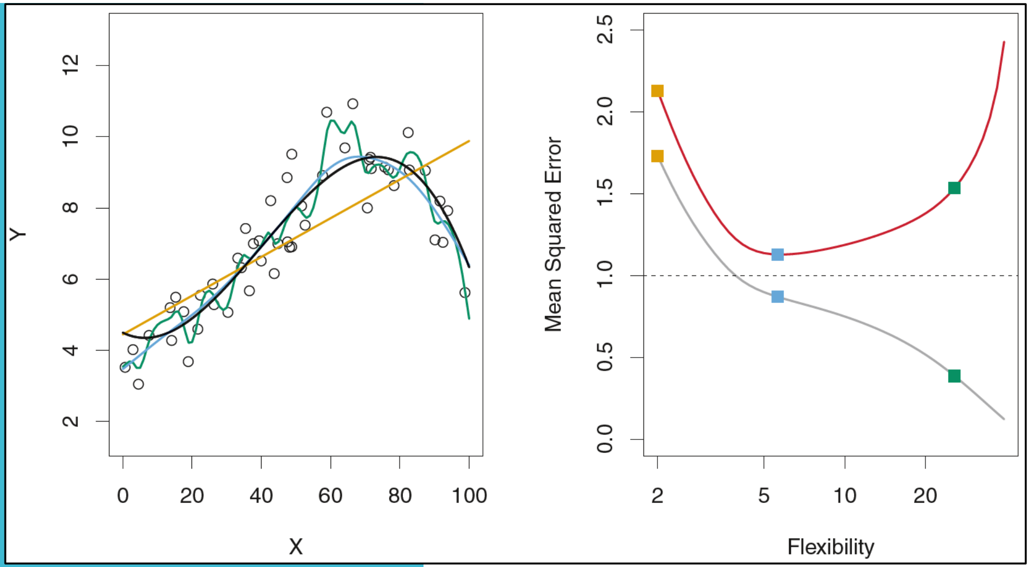
Accessing Model Accuracy

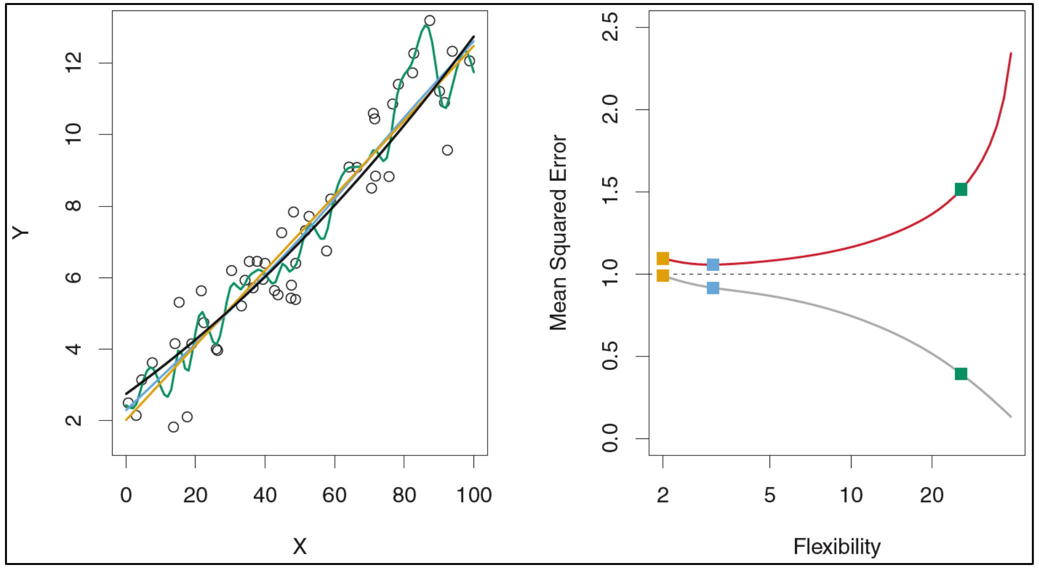
- Mean Squared Error (MSE) should be your friend in this course.
- Suppose we fit a model $\hat{f}(X)$ using a training data set Tr = $\{X_i, y_i\}_{i=1}^{n}$
- The training MSE is: $\mathrm{MSE_{Tr}} = (\frac{1}{n}) \sum_{i=1}^{n} \left(y_i - \hat{f}(X_i)\right)^2$
- Actually, the prevailing model fitting method, i.e., Ordinary Least Squares (OLS) method, explicitly minimize the MSE under the model assumption (i.e., the functional form)
- Training MSE does not reflect how well the model generalizes on new data
- On the same training data, however, it can reflect how well different models (e.g., polynomial model with different orders) fit the data.

MSE on the test set

- To see how the model generalizes to new data, we should evaluate it on the test data set that is unseen in the model building stage.
- Let Te = $\{X_i, y_i\}_{i=1}^{M}$ be the test set, then the test MSE is:
  - $\mathrm{MSE}_{\mathrm{Te}} = (\frac{1}{M}) \sum_{i=1}^{M} \left(y_i - \hat{f}(X_i)\right)^2$
- Test MSE reflects how the model generalizes on new cases.

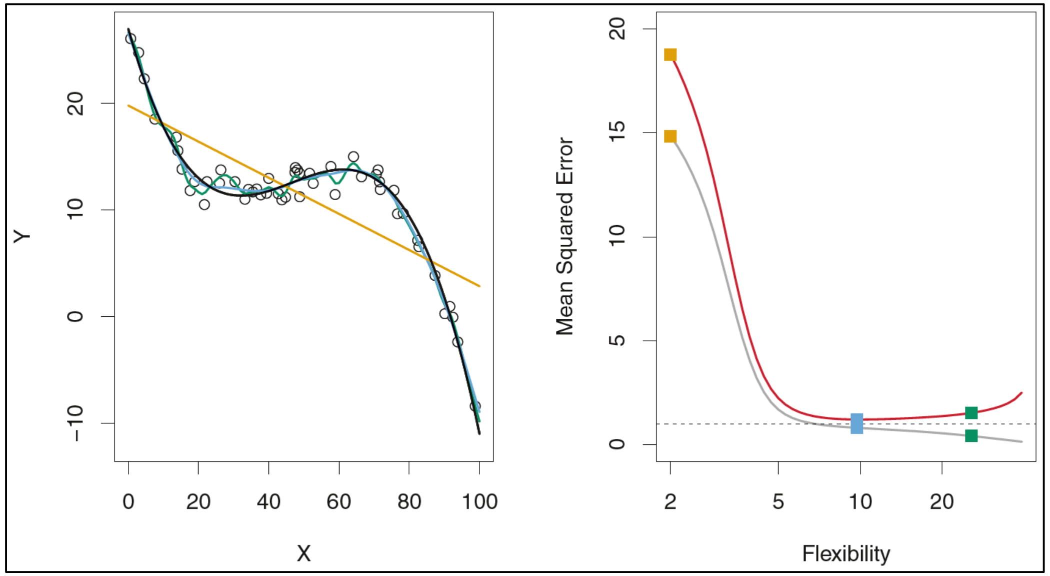
Training and test MSE





<u>On each set of plots:</u>

- Black curve is truth.
- Red curve on the right is test MSE, gray curve on the right is training MSE.
- Orange, blue and green are three models of different flexibility.



Bias and Variance

- **Bias**: the error introduced by making strong assumptions about reality; the error due to over-simplification
  - The simpler the model, the stronger the assumption is imposed by it. A simpler model has more bias.
  - Models with many parameters have more degrees of freedom, are more flexible, impose weaker or fewer assumptions, hence have less bias.
- **Variance**: the amount by which the model changes if trained by different samples of data; how stable or insensitive the model is against noise in the data
  - A flexible model is able to carve out intricate irregularities in data which may simply be noise; if the model is trained on a different training sample (from the same population), it tends to change a lot to carve out the noise. Hence a more flexible model has more variance.

Bias-Variance Tradeoff

Suppose we have fit a model $\hat{f}(x)$ to some training data Tr, and let (x_0, y_0) be a test observation drawn from the population. If the true model is $Y = f(X) + \epsilon$ (with $f(x) = E(Y|X = x)$), then

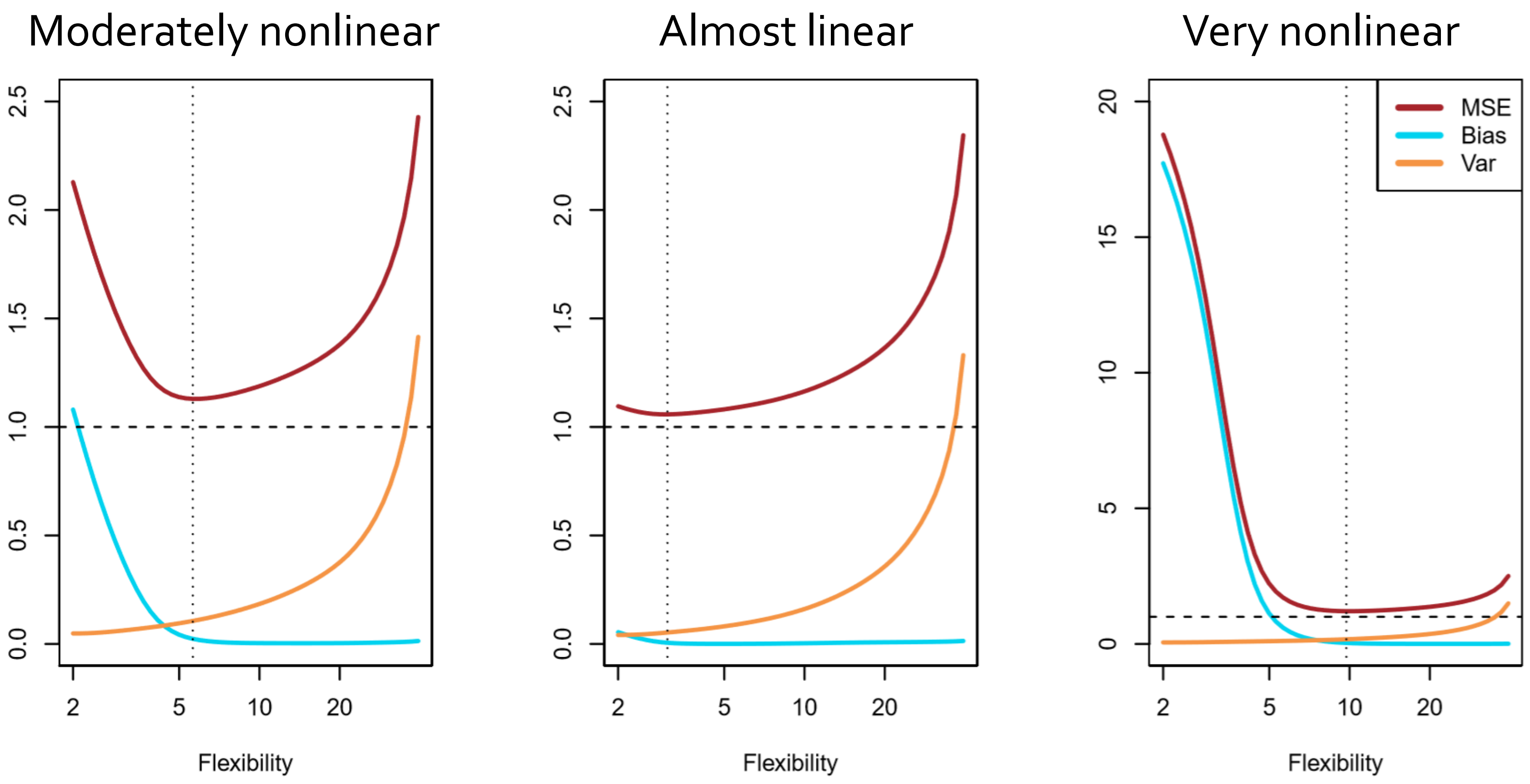
$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

The expectation averages over the variability of y_0 as well as the variability in Tr. Note that $\mathrm{Bias}(\hat{f}(x_0))] = E[\hat{f}(x_0)] - f(x_0)$.

Typically as the *flexibility* of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the flexibility based on average test error amounts to a *bias-variance trade-off.*

Bias-Variance Tradeoff for the three examples

The ground truth relationship between X and Y:



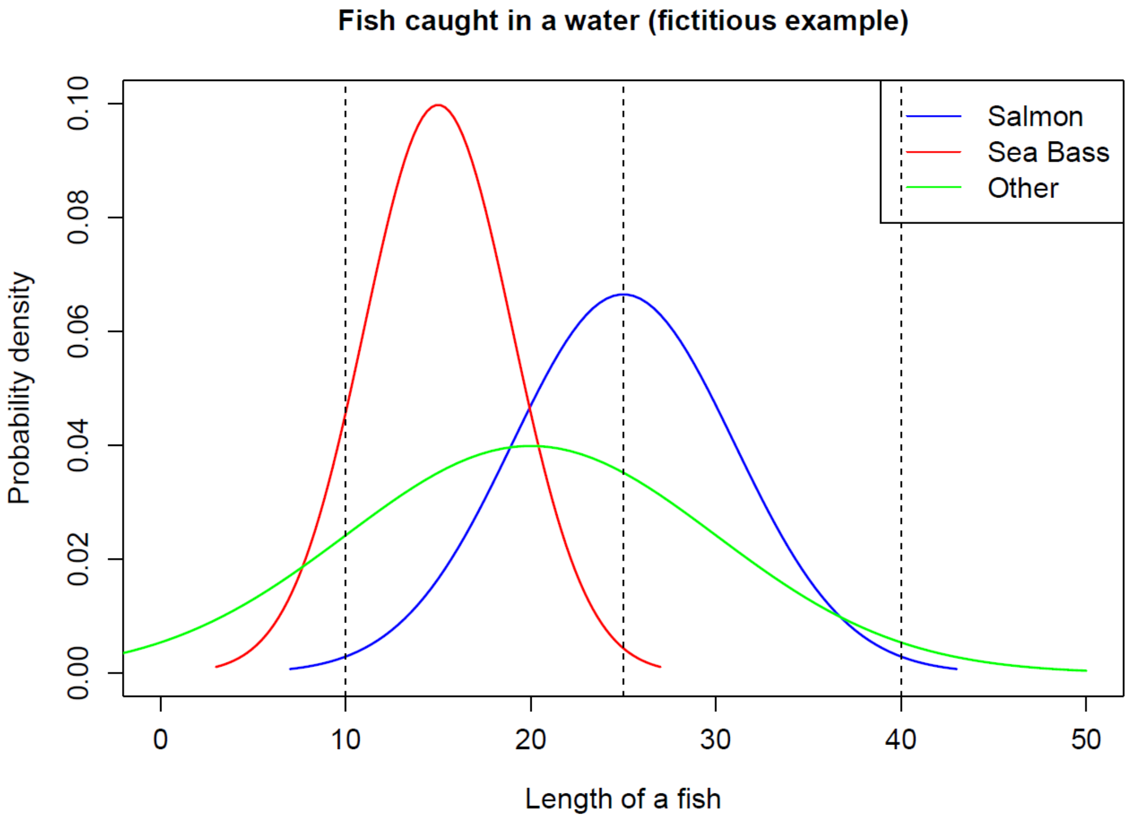
- Bias-Variance tradeoff plot can have different forms
- The core principles are the same

Classification

- To quantify the accuracy of $\hat{f}$ in a classification setting, we use the ***error rate***.
- Training error rate: $\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$
  - $\hat{y}_i$ is the predicted class label for the ith observation.
  - $I(y_i \neq \hat{y}_i)$ is an indicator variable, equal to 1 if $y_i \neq \hat{y}_i$, 0 otherwise
- Test error rate: Ave$(I(y_i \neq \hat{y}_i))$
- A good classifier is one for which the test error rate is small.

The Bayes Classifier

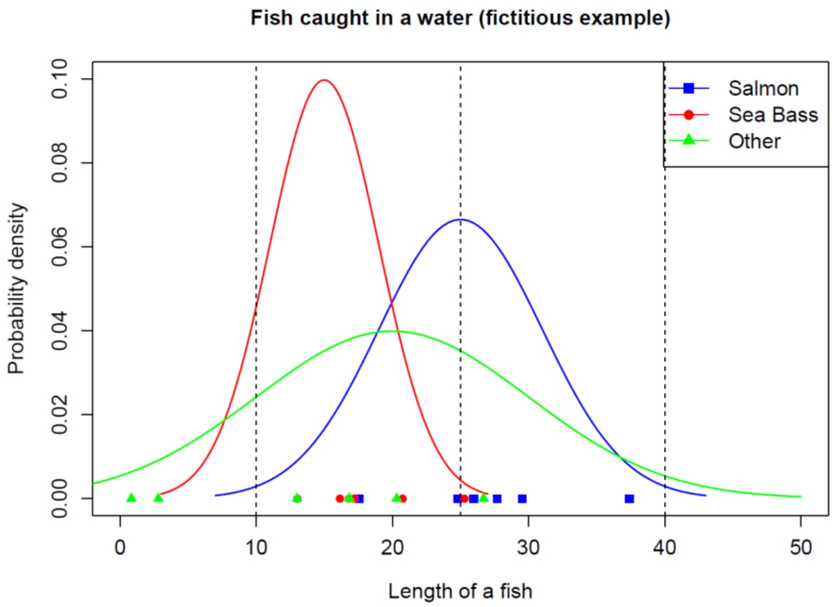
- **Bayes Decision Rule**: Assign the new observation to the most likely class, given its predictor value



Suppose you caught a 10-inch fish from the water. What is it?
P(Salmon | x = 10) ~= 0.003, P(Sea Bass | x = 10) ~= 0.045, P(Other | x = 10) ~= 0.024.
By Bayes decision rule, you classify the fish as a Sea Bass.

Bayes Classifier

- It can be proven that the Bayes classifier minimizes the expected test error rate.
- So the performance of the Bayes classifier provides the gold standard for all classifiers.
- However, in real situations the conditional probability distribution, i.e., the probability distribution of the predictor value for each class, is rarely known.
- No free lunch. We have to investigate other classifiers.

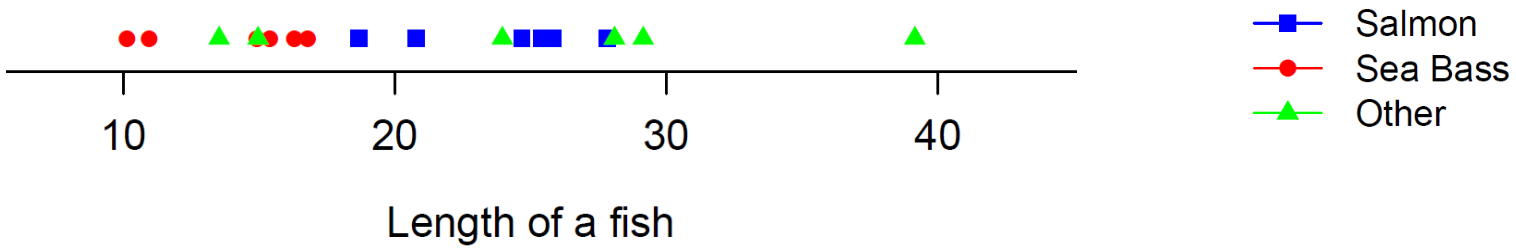


In practice:

You don't see these curves.

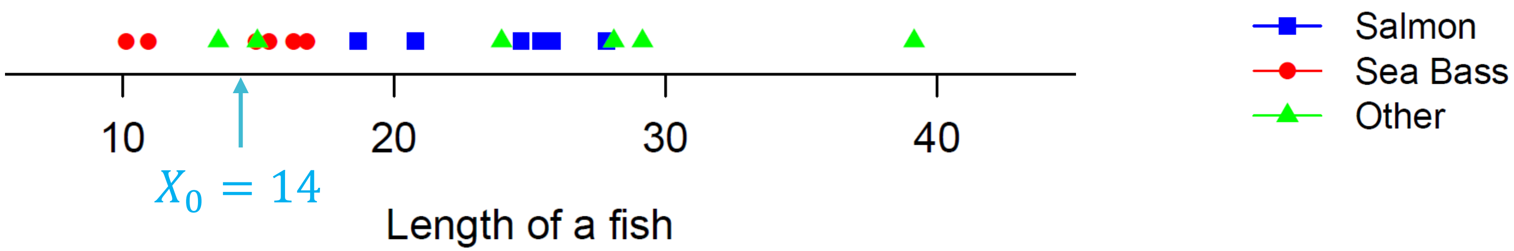
You only see some samples (fish caught and labeled in the past)

Practical problem and solution



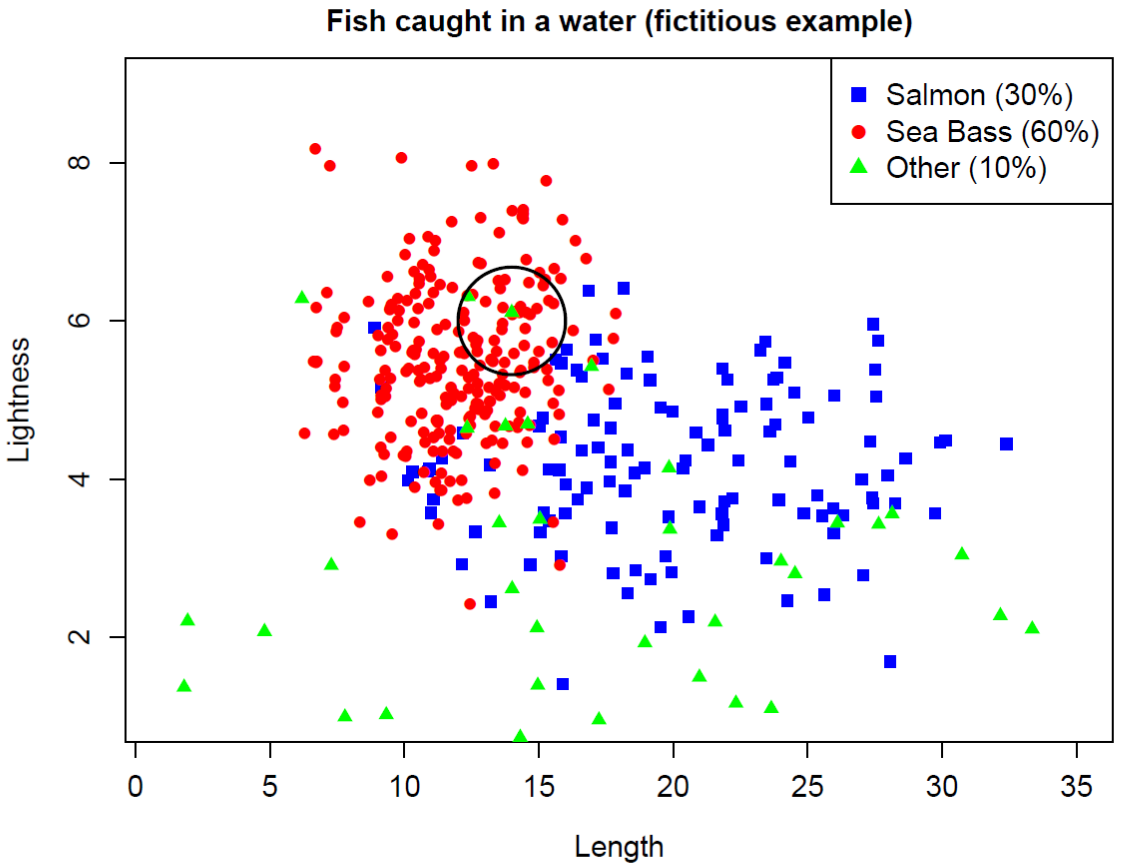
- Given some labeled samples, you are asked to build a classifier.
- Two approaches you can take:
  - **Model based**: assume that the observations (samples) come from some underlying distributions (model), use the samples to reconstruct (i.e., estimate the parameters of) the distributions, then apply the model to compute conditional probabilities, and use Bayes Decision Rule to classify the new fish. Example: **Linear Discriminant Analysis (LDA)**
  - **Instance based**: predict the class of the new fish by examining the classes of other fish of a similar length. Example: **K-nearest neighbors**
- Example: how would you classify a 14-inch fish?

K-nearest neighbors algorithm



- Given a positive integer K and a test observation X_0
  1. Identify K points in the training data that are **closest to $\boldsymbol{X_0}$**
  2. Predict the class for x_0 as the majority class of these K points
- Example: $\mathrm{X}_0 = 14$
  - If you picked K = 3, the predicted class is Other
  - If you picked K = 6, what is the predicted class?
- What about for other X_0 values between, say 10 and 18?
- For K = 6, all fish of length 10 to 18 will be classified as Sea Bass, which matches the ground truth. But for K = 3, some fish in this range will be misclassified as Other.

A 2D example



- Feature: (Length, Lightness)
- Example: how would you classify a new case $X_0 = (14, 6)$ at K = 20?

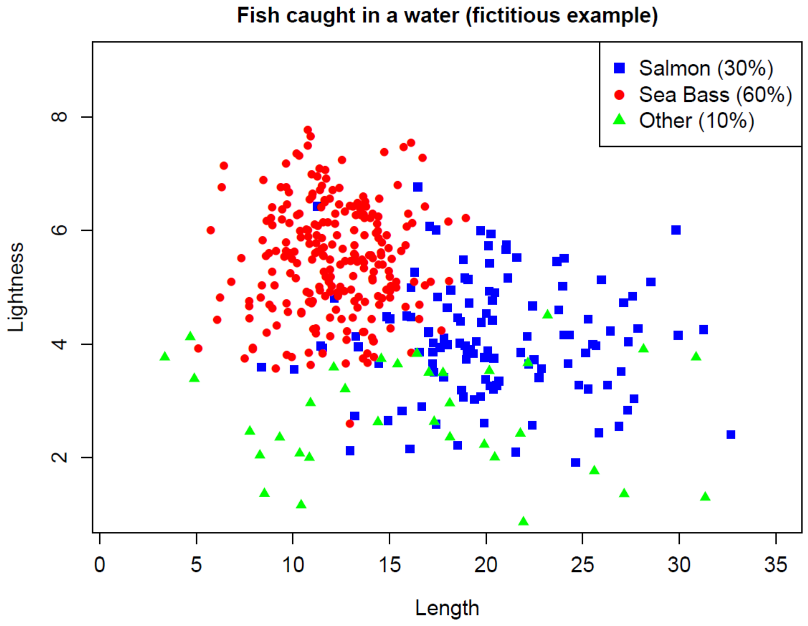
* The circle in the plot is for illustration only.

Similarity metrics

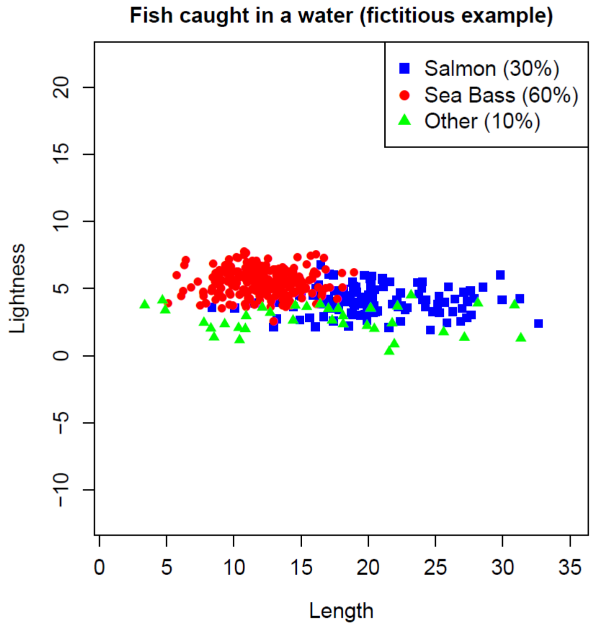
- To determine which points are **closest to $\boldsymbol{X_0}$**, we need a metric to measure the distance or similarity between any pair of points.
- For numeric features, **Euclidean distance** is the simplest metric.
  - $d(X, X') = \|X - X'\|_2 = \sqrt{\sum_{j=1}^{p}(X_j - X_j')^2}$
  - Normalization is typically need (scale each feature to [0, 1] or [-1, 1])
- Other similarity metrics:
  - Cosine similarity: $d(X, X') = \frac{X^T X'}{\|X\| \cdot \|X'\|}$
  - Radial basis function (RBF) kernel: $K(X, X') = \exp(-\gamma\|X - X'\|^2)$
  - Hamming distance for strings
  - Dynamic time warping (DTW) for temporal sequences

Why need feature scaling?

Scaled plot

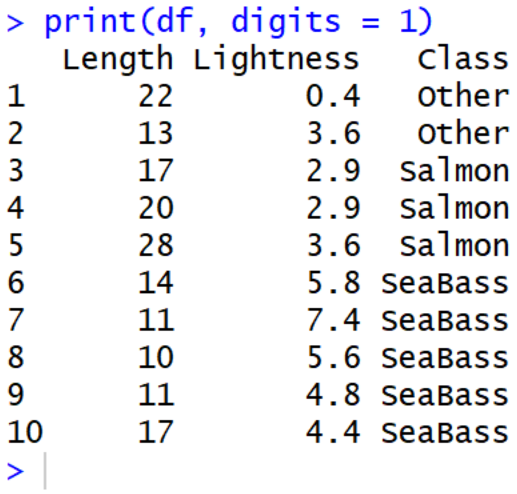


Unscaled plot



- Without scaling, features with small numerical values (e.g., Lightness) are disadvantaged in the distance calculation.
- Two approaches to address this issue:
  - **Normalization** (or min-max scaling): subtract the min, then divide by the max minus the min. All features will end up in [0,1].
  - **Standardization**: subtract the mean, then divide by the standard deviation. Less sensitive to outliers.

Euclidean distance example

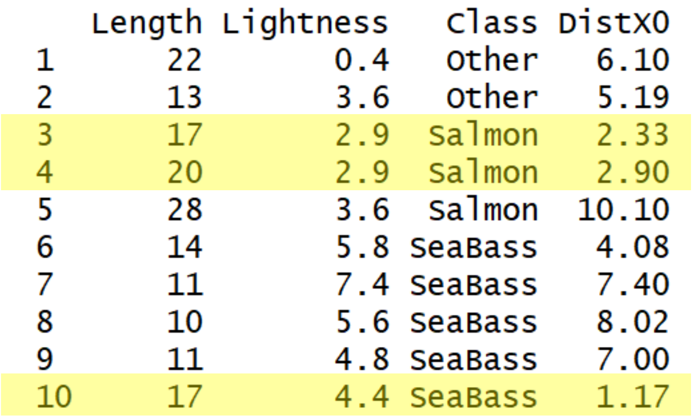


```
> print(df, digits = 1)
   Length Lightness   Class
1      22       0.4   Other
2      13       3.6   Other
3      17       2.9  Salmon
4      20       2.9  Salmon
5      28       3.6  Salmon
6      14       5.8 SeaBass
7      11       7.4 SeaBass
8      10       5.6 SeaBass
9      11       4.8 SeaBass
10     17       4.4 SeaBass
> 
```

- There is a sample of 10 fish in the training set.
- Suppose we have a new fish X_0 whose length = 18 and lightness = 5.0.
  1. Calculate the Euclidean distance between this fish and each fish in the training set.
  2. Using the K-nearest neighbors algorithm with K = 3, what is the predicted class of this fish?
- $d(X_0, X_1) = \sqrt{(18-22)^2 + (5.0-0.4)^2} = 6.10$
- $d(X_0, X_2) = \cdots$

Euclidean distance example

Calculated the distance between $X_0 = (18, 5)$ and all other points. The result is listed in column “DistXo”.

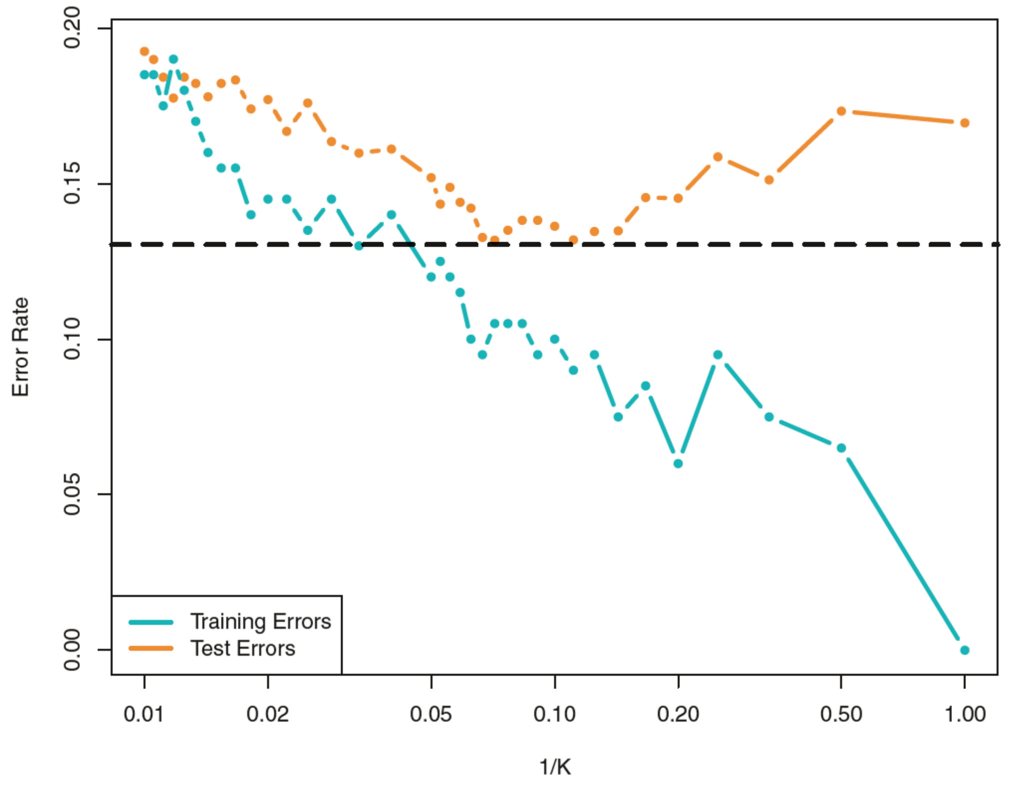


```
   Length Lightness   Class DistX0
1      22       0.4   Other   6.10
2      13       3.6   Other   5.19
3      17       2.9  Salmon   2.33
4      20       2.9  Salmon   2.90
5      28       3.6  Salmon  10.10
6      14       5.8 SeaBass   4.08
7      11       7.4 SeaBass   7.40
8      10       5.6 SeaBass   8.02
9      11       4.8 SeaBass   7.00
10     17       4.4 SeaBass   1.17
```

The three nearest neighbors of $X_0 = (18, 5.0)$ are X_3, X_4 and X_{10}.
The majority class of the nearest neighbors is Salmon.
Therefore, the predicted class of X_0 is Salmon.

Exercise: ISLR 2.4 Exercises 7.

The parameter K in the K-nearest neighbor algorithm



- A smaller K results in a more flexible model (i.e., more adaptable and sensitive to change in training sample, more variance than bias)
- A larger K results in a more stable model (i.e., less sensitive to both signal and noise in the training data, more bias than variance)

Intelligibility issue with instance-based learning

- In fields such as medicine and law, reasoning about similar historical cases is a natural way of coming to a decision about a new case.
- However, in business it may not work well, for example, “We declined your mortgage application because you remind us of the Smiths and the Mitchells, who both defaulted”.
- With a linear model, “all else held equal, if your income had been $20,000 higher you would have been granted this particular mortgage”.

Exercise

- On next Tuesday, we will work on some Exercise problems in ISLR Section 2.4.
  - Conceptual 1, 2, 3, 4, 5, 6, 7
  - Applied: 8