SDU Summer School

Deep Learning

Summer 2018

Machine Learning Basics

Machine Learning Basics

- Introduction
- Statistical Learning
- Cross-Validation

Machine Learning

- An ML algorithm is an algorithm that is able to learn from data
  - A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at task T, as measured by P, improves with experience E

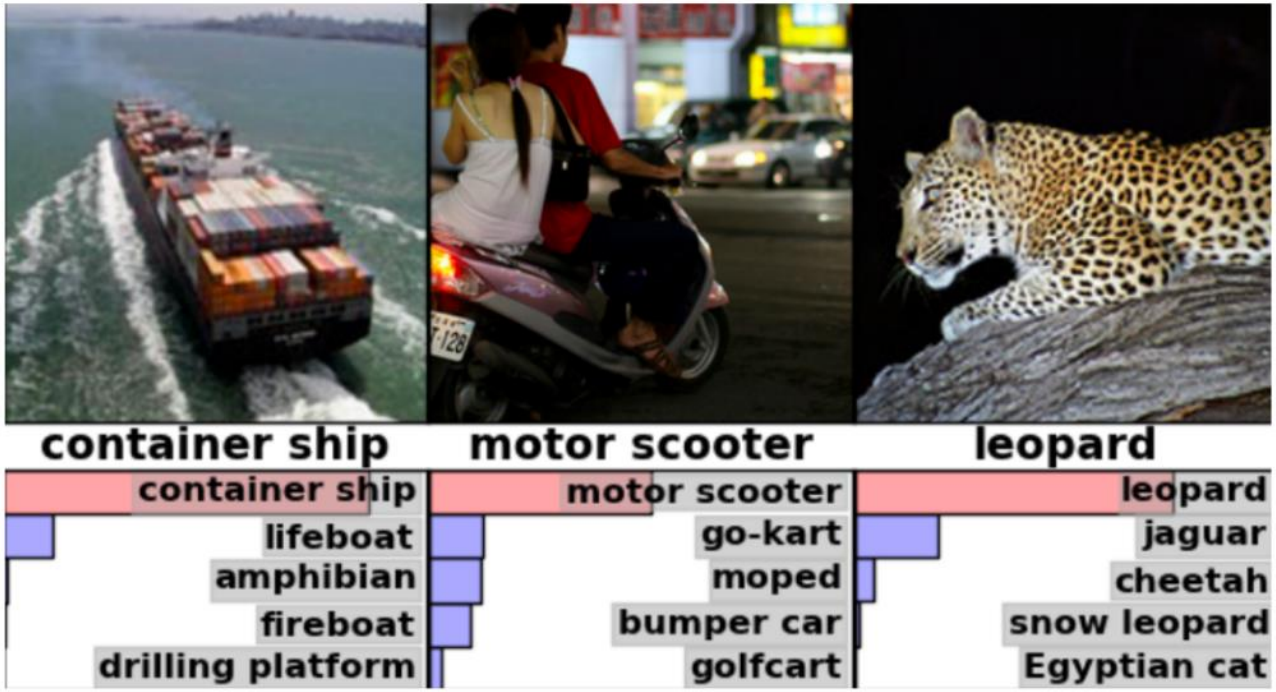
- The Task T:
  - Process of learning itself is not the task
  - Learning is our means of attaining ability to perform the task
  - Usually described in terms of how the machine learning system should process an example
  - Typically represent an example as a vector $x \in \mathbb{R}^n$ where each entry x_i of the vector is another feature

Kinds of Tasks solved using ML

- Classification
- Regression
- Transcription
- Machine Translation
- Structured Output
- Anomaly Detection
- Synthesis and Sampling
- Imputation of Missing Values
- Denoising
- Density Estimation
- …

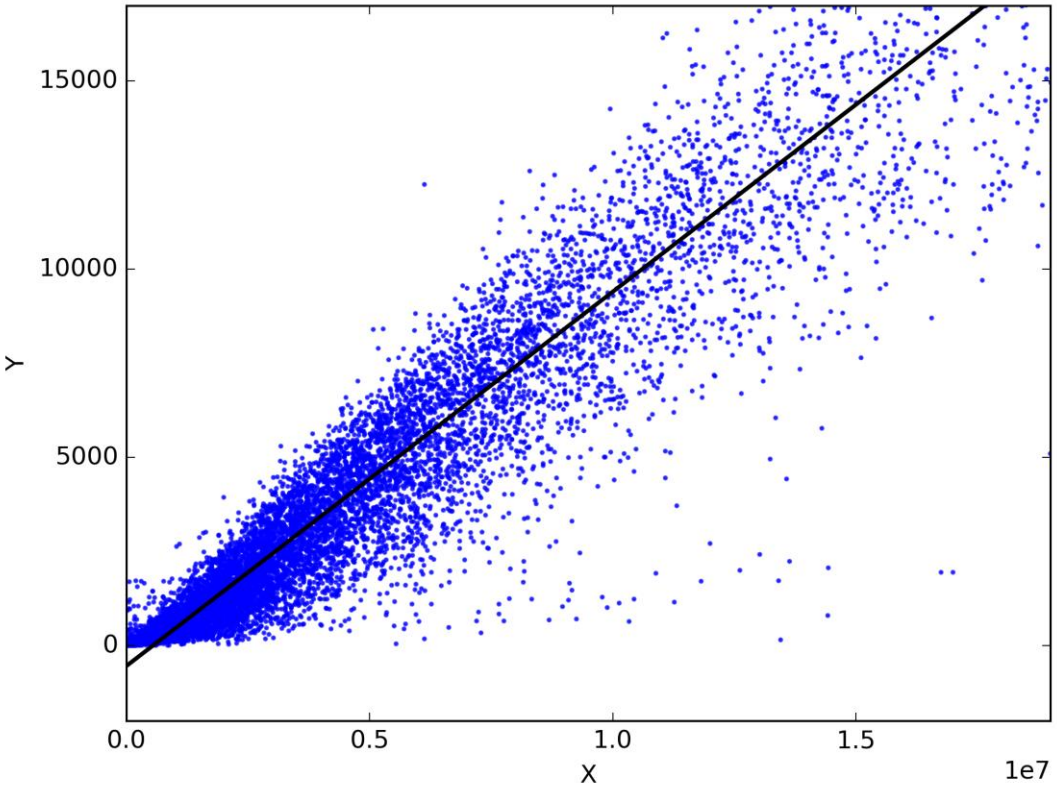
Kinds of Tasks solved using ML

- **Classification**
- Regression
- Transcription
- Machine Trar
- Structured O
- Anomaly Det
- Synthesis and
- Imputation o
- Denoising
- Density Estin
- …



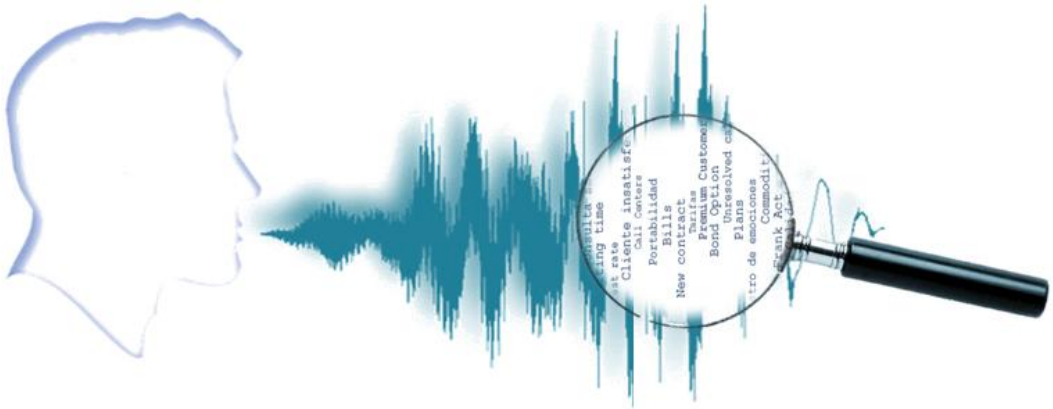
Kinds of Tasks solved using ML

- Classification
- **Regression**
- Transcription
- Machine Translation
- Structured Output
- Anomaly Detection
- Synthesis and Samplir
- Imputation of Missing
- Denoising
- Density Estimation
- …



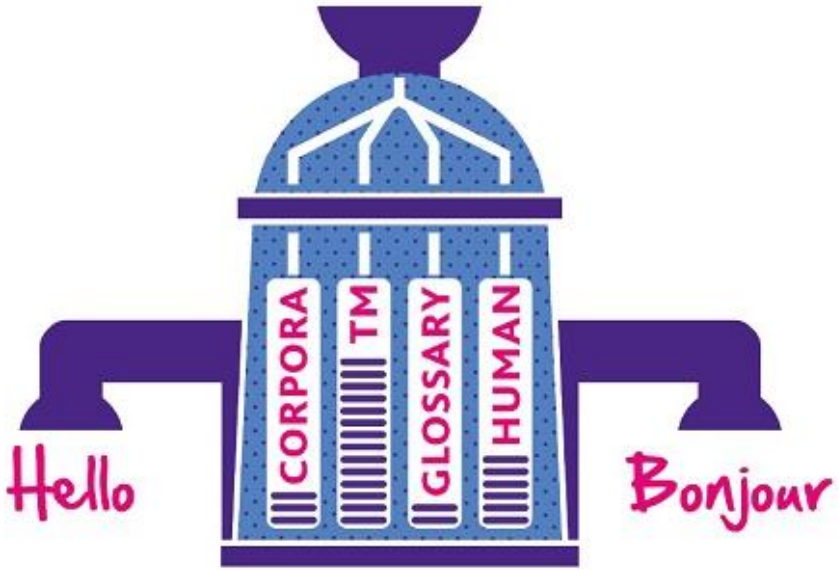
Kinds of Tasks solved using ML

- Classification
- Regression
- **Transcription**
- Machine Translation
- Structured Output
- Anomaly Detection
- Synthesis and Sampling
- Imputation of Missing Values
- Denoising
- Density Estimation
- …



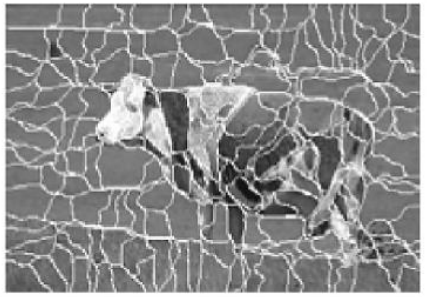
Kinds of Tasks solved using ML

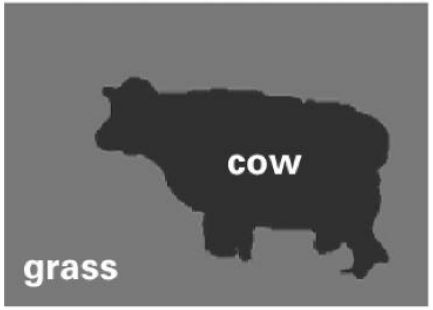
- Classification
- Regression
- Transcription
- **Machine Translation**
- Structured Output
- Anomaly Detection
- Synthesis and Sampling
- Imputation of Missing Values
- Denoising
- Density Estimation
- …



Kinds of Tasks solved using ML

- Classification
- Regression
- Transcription
- Machine Translation
- **Structured Output**
- Anomaly Detection
- Synthesis and Sampli
- Imputation of Missin
- Denoising
- Density Estimation
- …

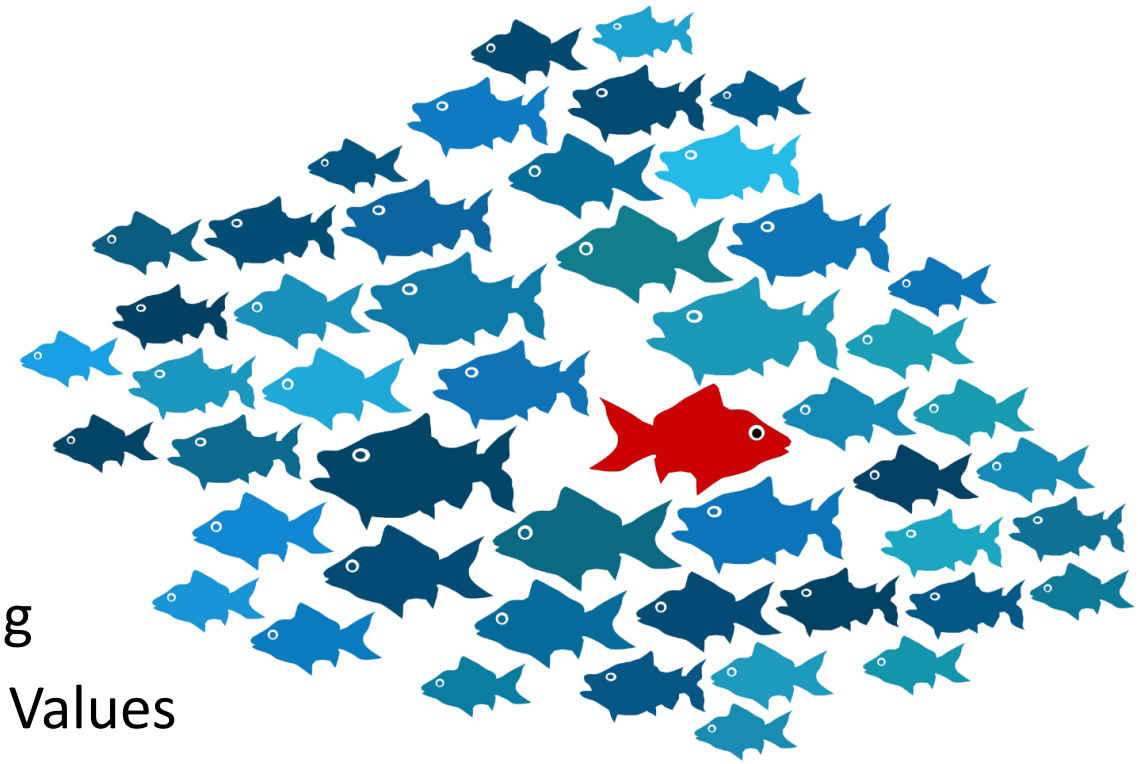




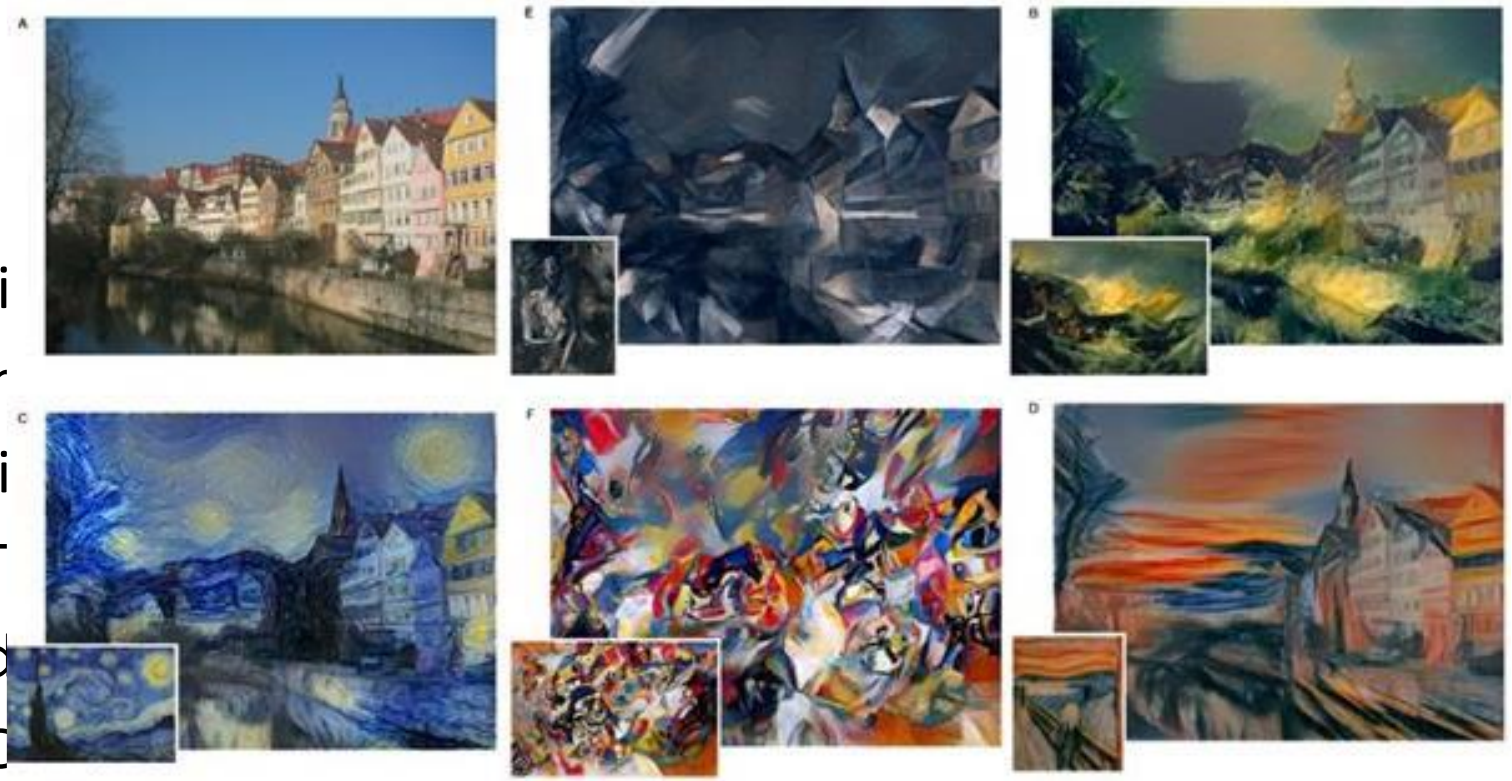
The man at bat readies to swing at the pitch while the umpire looks on.

Kinds of Tasks solved using ML

- Classification
- Regression
- Transcription
- Machine Translation
- Structured Output
- **Anomaly Detection**
- Synthesis and Sampling
- Imputation of Missing Values
- Denoising
- Density Estimation
- …

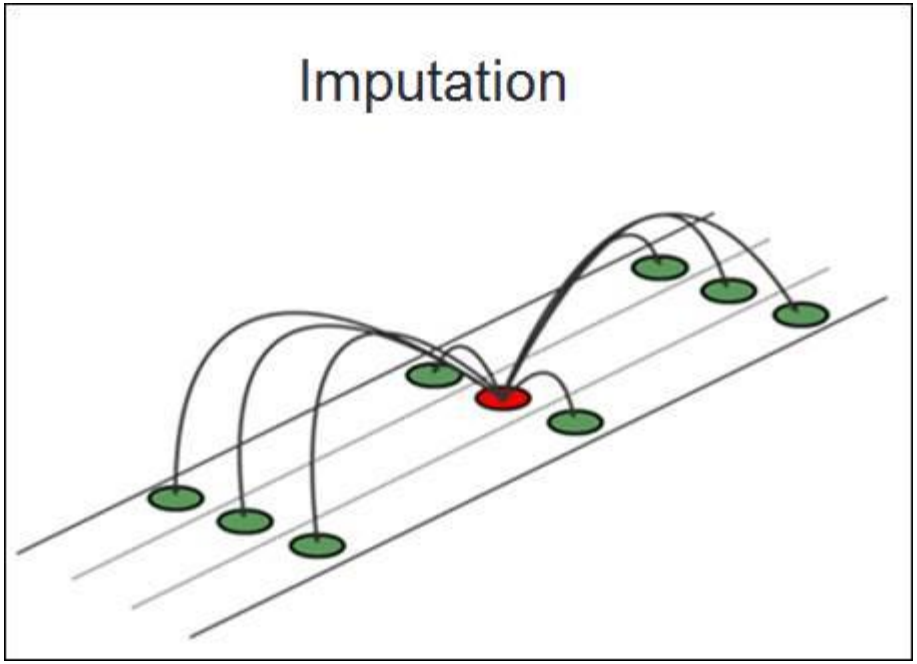


- Classificati
- Regressior
- Transcripti
- Machine T
- Structured
- Anomaly D
- **Synthesis and Sampling**
- Imputation of Missing Values
- Denoising
- Density Estimation
- …



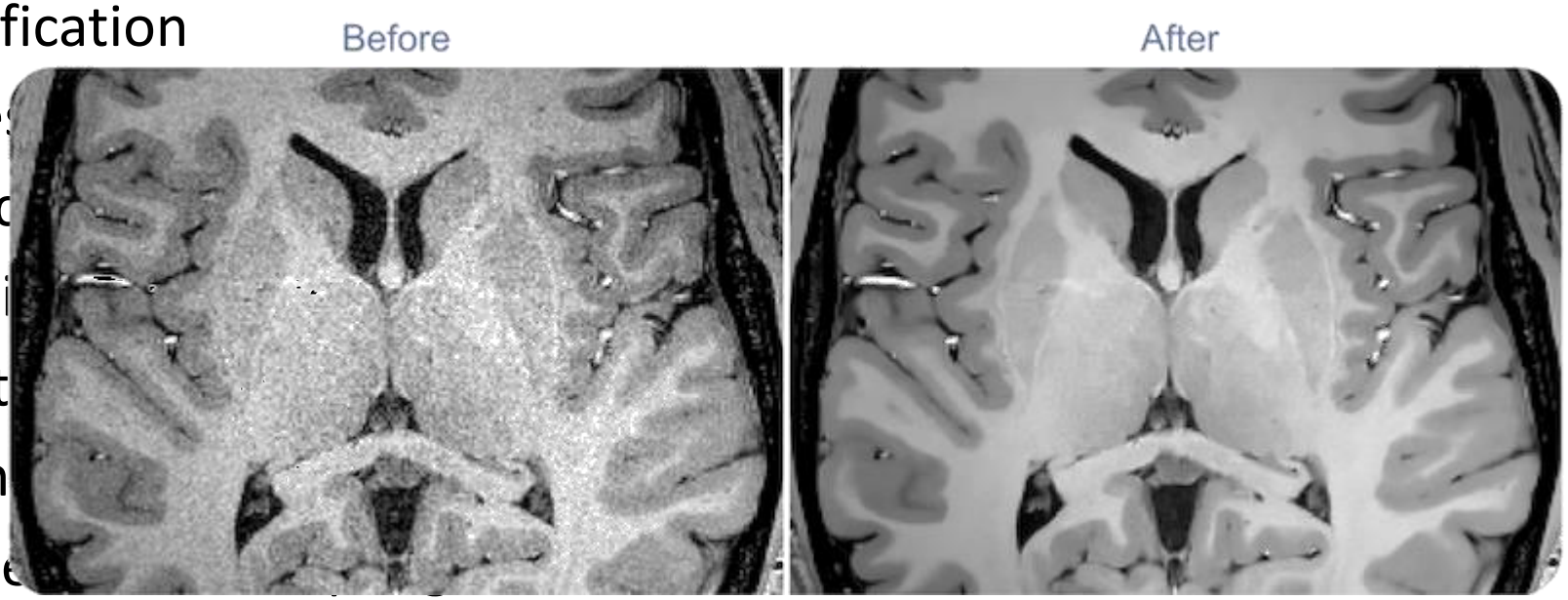
Kinds of Tasks solved using ML

- Classification
- Regression
- Transcription
- Machine Translation
- Structured Output
- Anomaly Detection
- Synthesis and Sampling
- **Imputation of Missing Values**
- Denoising
- Density Estimation
- …



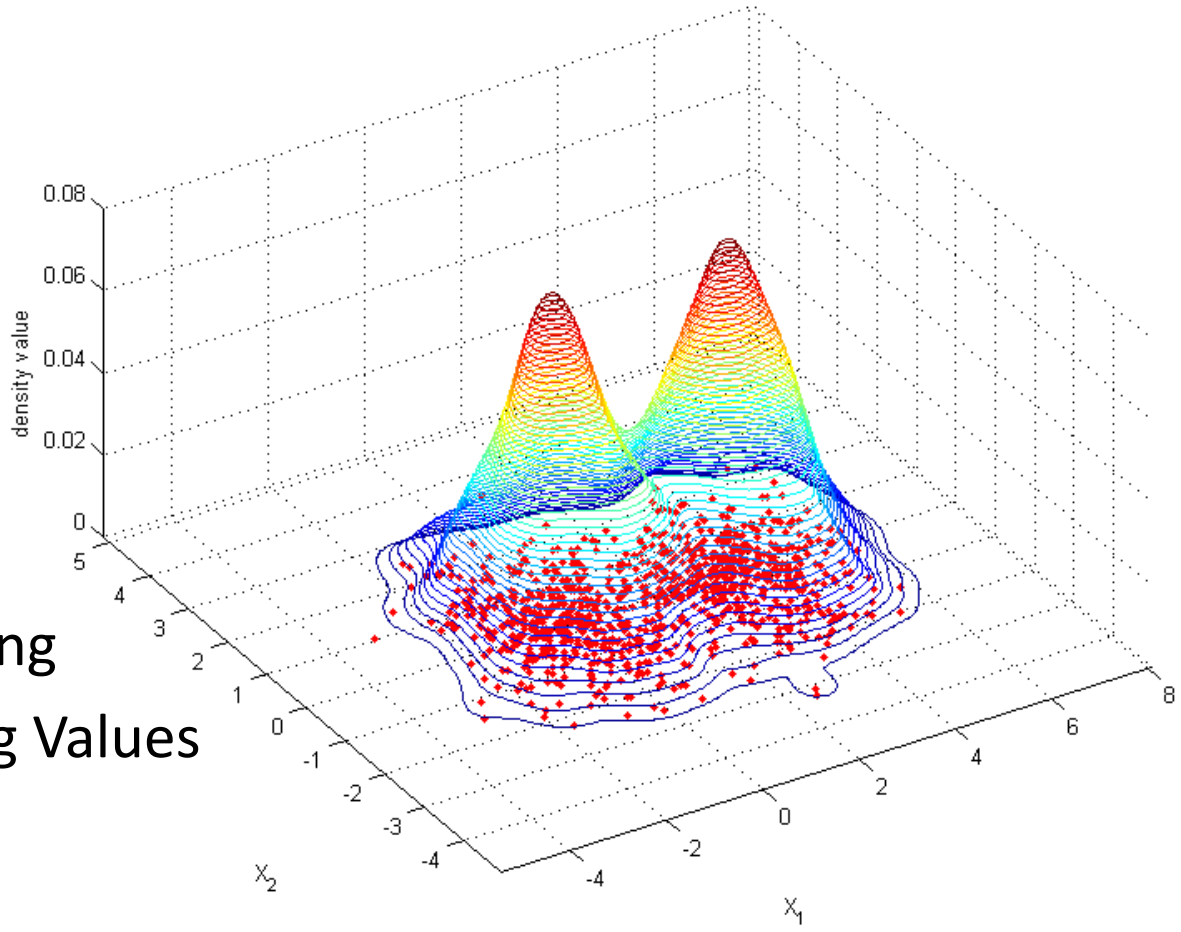
Kinds of Tasks solved using ML

- Classification
- Regres
- Transc
- Machi
- Struct
- Anom
- Synthe
- Imputation of Missing Values
- **Denoising**
- Density Estimation
- …



Kinds of Tasks solved using ML

- Classification
- Regression
- Transcription
- Machine Translation
- Structured Output
- Anomaly Detection
- Synthesis and Sampling
- Imputation of Missing Values
- Denoising
- **Density Estimation**
- …



Performance of a Method

- For classification, transcription, etc. we often measure the accuracy of the model.
  - Proportion of examples for which the model produces the correct output
- Equivalently, we can measure the error rate
  - Often referred to as the expected 0-1 loss
- Generally, there are hundreds of different performance measures
  - Often difficult to find a good performance measure
    - should we penalize the system more if it frequently makes medium-sized mistakes or if it rarely makes very large mistakes
    - The quantity we would ideally like to measure is impractical
    - …

On Method Performance

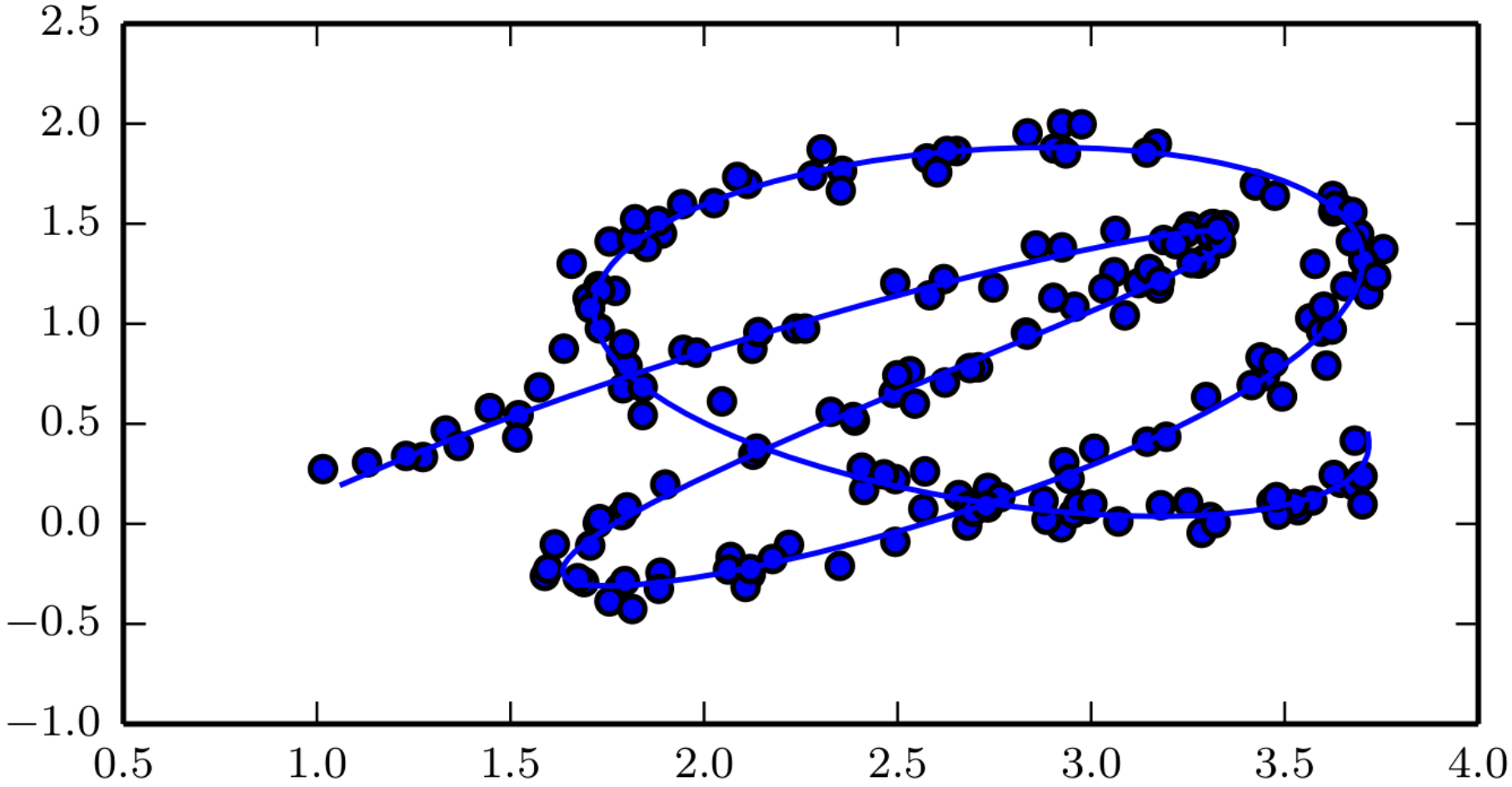
- The No Free Lunch Theorem [Wolpert , 1996]

“averaged over all possible data generating distributions, every classification algorithm has the same error rate when classifying previously unobserved points”

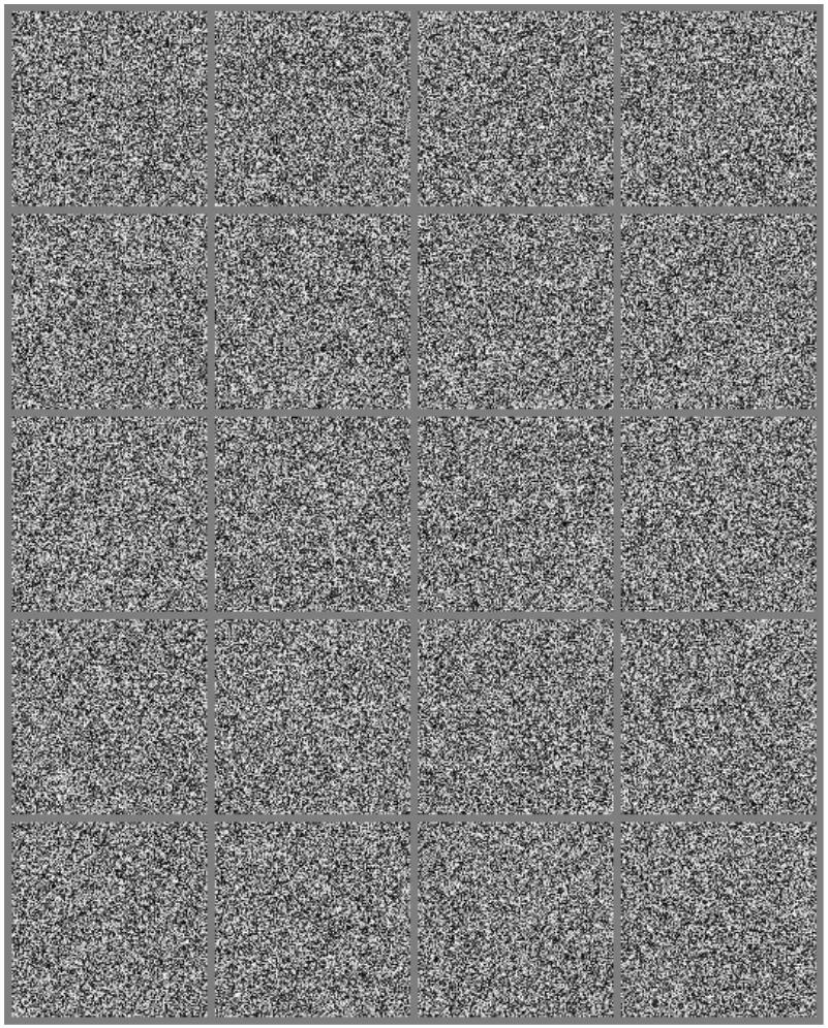
- In other words, no machine learning technique is per se better than another

- **BUT**: In really, we do not encounter ALL possible distributions but only a few on which we need to perform good!

Manifold Learning



Uniformly Sampled Images



Machine Learning Basics

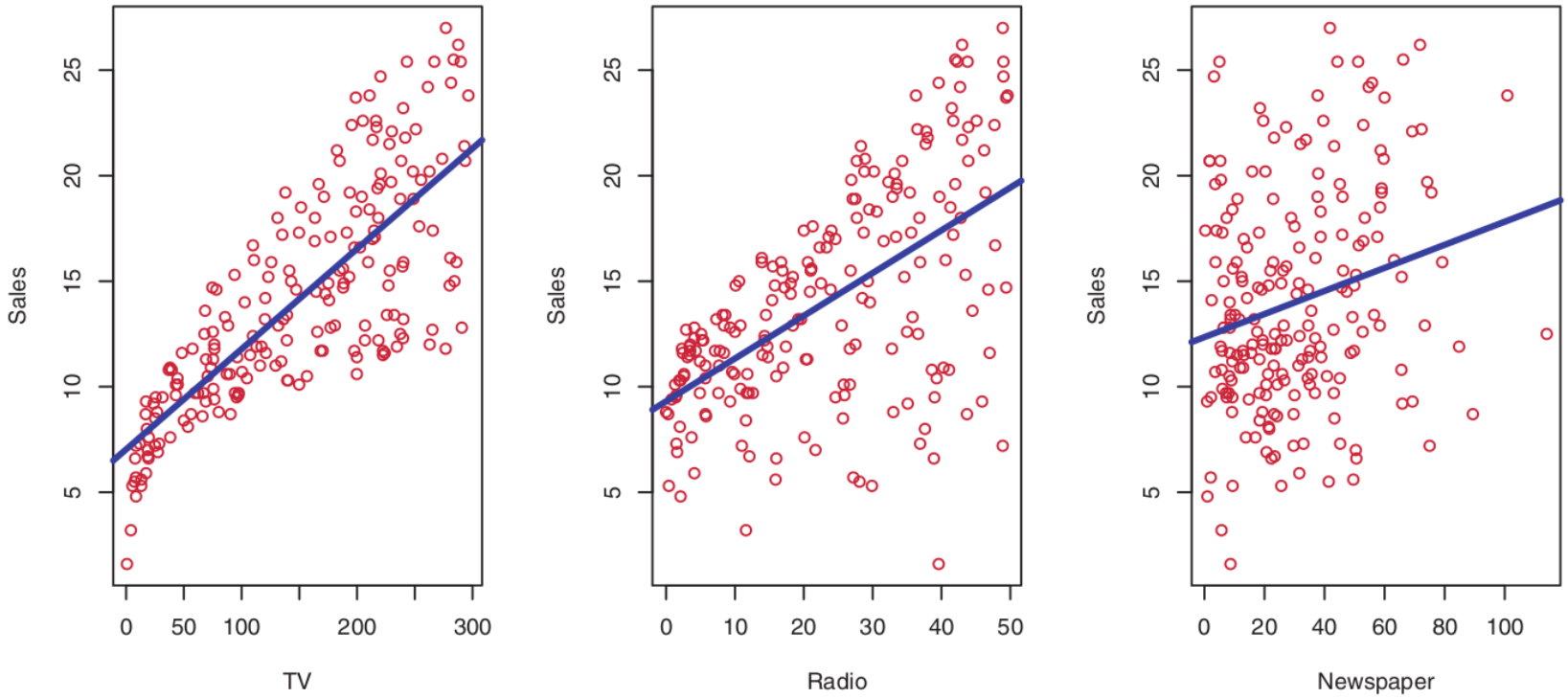
- Introduction
- Statistical Learning
- Cross-Validation

Wikipedia

Statistical learning theory is a framework for machine learning drawing from the fields of statistics and functional analysis.

Statistical learning theory deals with the problem of finding a predictive function based on data.

What is Statistical Learning?



- Shown are Sales vs. TV, Radio and Newspaper ads
- Can we predict Sales using these three?
- Perhaps we can do better using a model

Notation

- Here Sales is a response or target that we wish to predict. We generically refer to the response as Y.
- TV is a feature, or input, or predictor; we name it X_1.
  - Likewise name Radio as X_2, and so on.
- We can refer to the input vector collectively as

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

- Now we write our model as

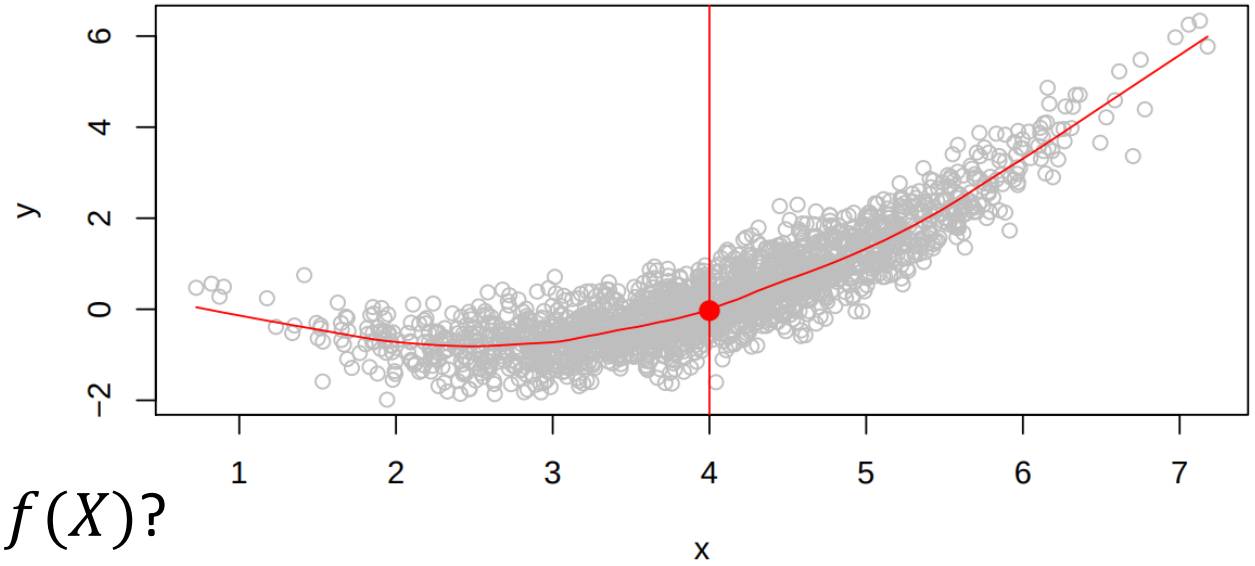
$$Y = f(X) + \epsilon$$

- where ϵ captures measurement errors and other discrepancies.

What is that good for?

- With a good f we can make predictions of Y at new points $X = x$.
- We can understand which components of $X = (X_1, X_2, \dots, X_p)$ are important in explaining Y, and which are irrelevant. e.g. Seniority and Years of Education have a big impact on Income, but Marital Status typically does not.
- Depending on the complexity of f, we may be able to understand how each component X_j of X affects Y.

The Regression Function



- Is there an optimal $f(X)$?
- What is the optimal value $f(X)$ at any selected value of X.
- For example: $X = 4$, then

$$f(4) = E[Y|X = 4]$$

- Ideally, we predict the **expected value** of Y for any given X
- This ideal $f(x) = E[Y|X = x]$ is called the **regression function**.

The Regression Function

- Is the ideal or optimal predictor of Y with regard to mean-squared prediction error: $f(x) = E(Y|X = x)$ is the function that minimizes $E\left[\left(Y - g(X)\right)^2 | X = x\right]$ over all functions g at all points $X = x$.

- $\epsilon = Y - f(x)$ is the irreducible error — i.e. even if we knew $f(x)$, we would still make errors in prediction, since at each $X = x$ there is typically a distribution of possible Y values.
- For any estimate $\hat{f}(x)$ of $f(x)$, we have

$$E\left[\left(Y - \hat{f}(X)\right)^2 | X = x\right] = \underbrace{\left[f(x) - \hat{f}(x)\right]^2}_{\text{Reducible}} + \underbrace{Var(\epsilon)}_{\text{Irreducible}}$$

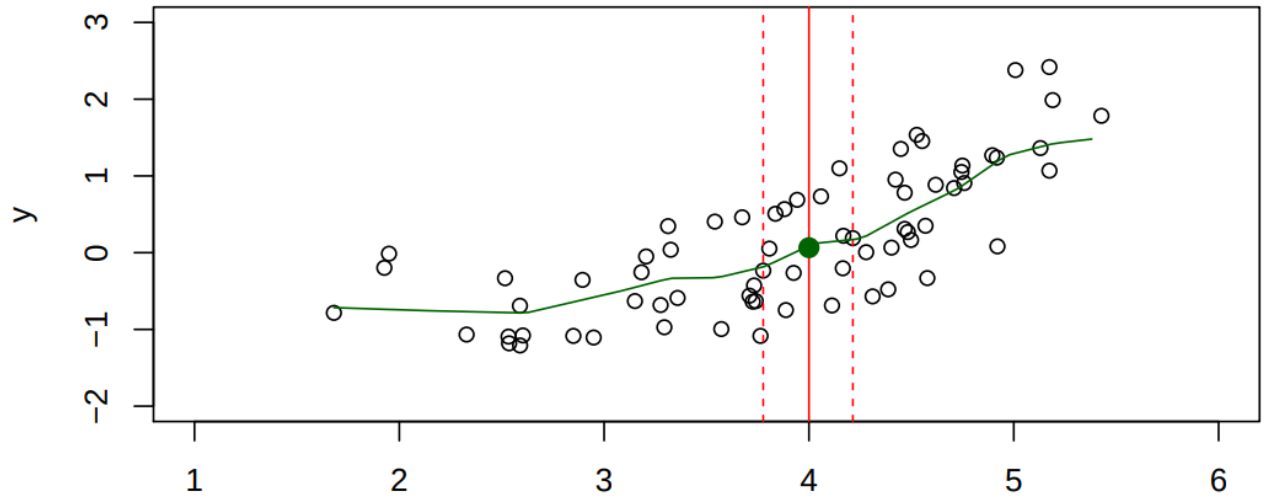
How to estimate f

- Typically we have few if any data points with $X = 4$ exactly.
  - So we cannot compute $E(Y|X = x)$!

- Relax the definition and let

$$\hat{f}(x) = Ave(Y|X \in \mathcal{N}(x))$$

where $\mathcal{N}(x)$ is some neighborhood of x.



How to estimate f

- Nearest neighbor averaging can be pretty good for small number of dimensions p, e.g. < 4 and large number of samples

- Nearest neighbor methods can be lousy when p is large.
  - Curse of dimensionality. Nearest neighbors tend to be far away in high dimensions.
  - We need to get a reasonable fraction of the N values of y_i to average to bring the variance down—e.g. 10%.
  - A 10% neighborhood in high dimensions need no longer be local, so we lose the spirit of estimating $E(Y|X = x)$ by local averaging.

Parametric and structured models

- The linear model is an important example of a parametric model:

$$f_L(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots + \beta_p X_p$$

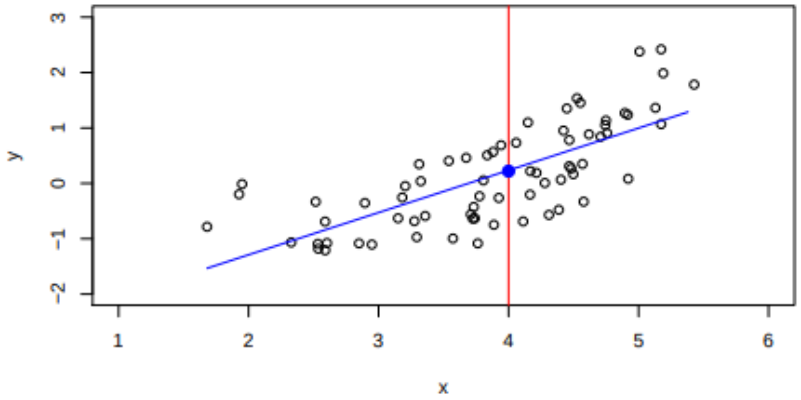
- A linear model is specified in terms of $p + 1$ parameters $\beta_0, \beta_1, \ldots, \beta_p$.

- We estimate the parameters by fitting the model to training data.

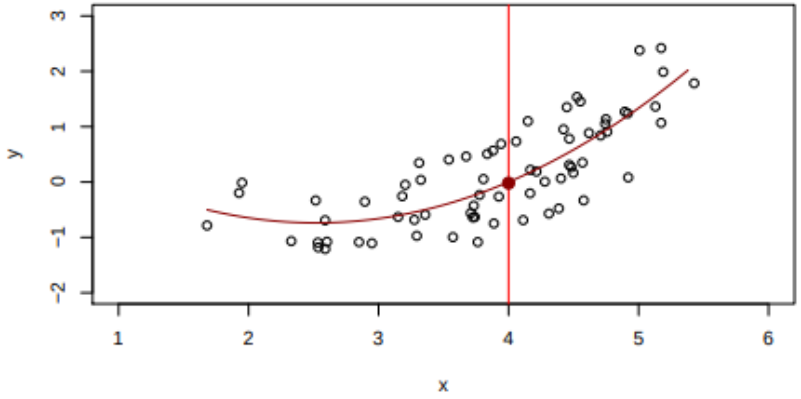
- Although it is almost never correct, a linear model often serves as a good and interpretable approximation to the unknown true function f(X).

Different Models

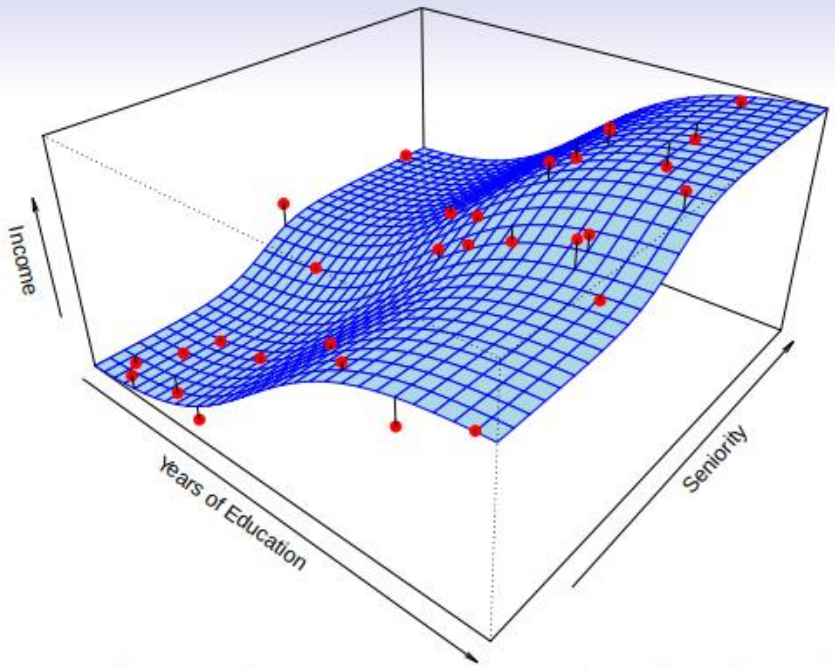
A linear model $\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ gives a reasonable fit here



A quadratic model $\hat{f}_Q(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$ fits slightly better.



Different Models

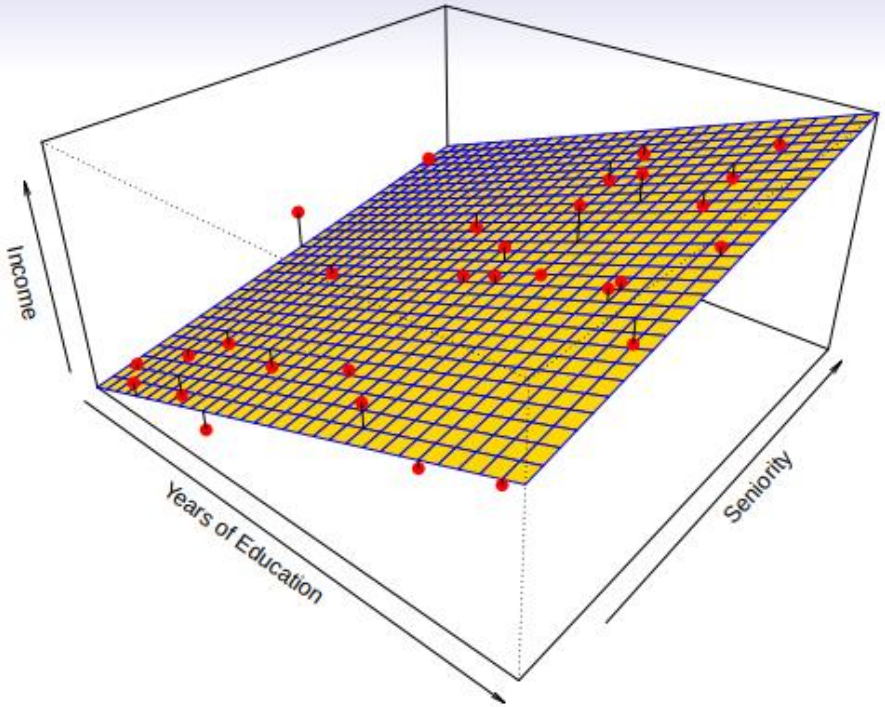


Simulated example. Red points are simulated values for income from the model

$$\texttt{income} = f(\texttt{education}, \texttt{seniority}) + \epsilon$$

f is the blue surface.

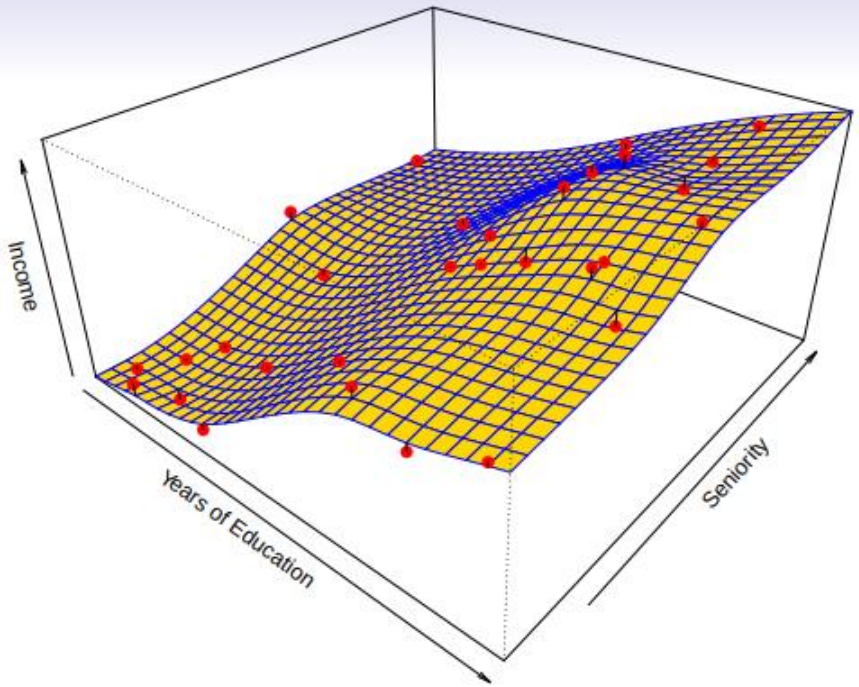
Model Complexity



Linear regression model fit to the simulated data.

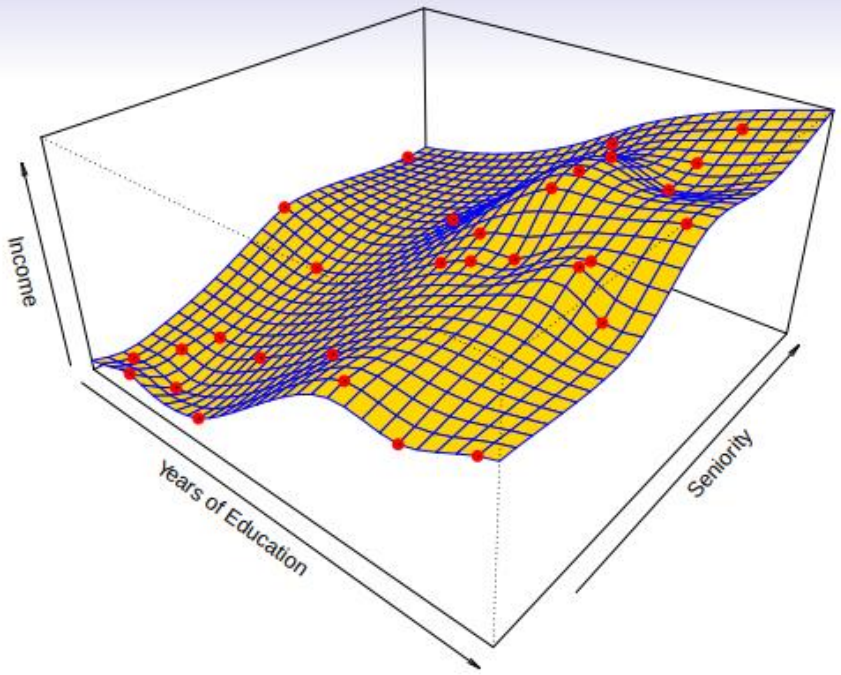
$$\hat{f}_L(\texttt{education}, \texttt{seniority}) = \hat{\beta}_0 + \hat{\beta}_1 \times \texttt{education} + \hat{\beta}_2 \times \texttt{seniority}$$

Model Complexity



More flexible regression model $\hat{f}_S(\texttt{education}, \texttt{seniority})$ fit to the simulated data. Here we use a technique called a *thin-plate spline* to fit a flexible surface.

Model Complexity



Even more flexible spline regression model $\hat{f}_S(\texttt{education}, \texttt{seniority})$ fit to the simulated data. Here the fitted model makes no errors on the training data! Also known as *overfitting*.

Model Complexity

Trade Offs:

- Prediction accuracy versus interpretability.
  - Linear models are easy to interpret; thin-plate splines are not. Deep Neural Networks are even harder to interpret
- Good fit versus over-fit or under-fit.
  - How do we know when the fit is just right?
- Parsimony versus black-box.
  - We often prefer a simpler model involving fewer variables over a black-box predictor involving them all.

Generalization

- Central challenge of ML is that the algorithm must perform well on new, **previously unseen** inputs

- Ability to perform well on previously unobserved inputs is called **generalization**

- Normally we have a test and training dataset
- Difference to normal optimization
  - We try to minimize the error on unseen data not on the training data, e.g. in the case of a linear regression: $\hat{y} = \boldsymbol{w}^T \boldsymbol{x} + b$

$$\frac{1}{m^{(\text{test})}} \left\| X^{(\text{test})} \boldsymbol{w} - \boldsymbol{y}^{(\text{test})} \right\|_2^2$$

Estimating the generalization error

- To sum up with the example of linear regression:

- In linear regression example we train model by minimizing the training error

$$\frac{1}{m^{(\text{train})}} \left\| X^{(\text{train})} \boldsymbol{w} - \boldsymbol{y}^{(\text{train})} \right\|_2^2$$

- **BUT**: We are actually interested in the test error:

$$\frac{1}{m^{(\text{test})}} \left\| X^{(\text{test})} \boldsymbol{w} - \boldsymbol{y}^{(\text{test})} \right\|_2^2$$

- **How can we affect performance when we observe only the training set?**

Under- and Over-fitting

- Factors determining how well an ML algorithm will perform are its ability to:
  1. Make the training error small
  2. Make gap between training and test errors small

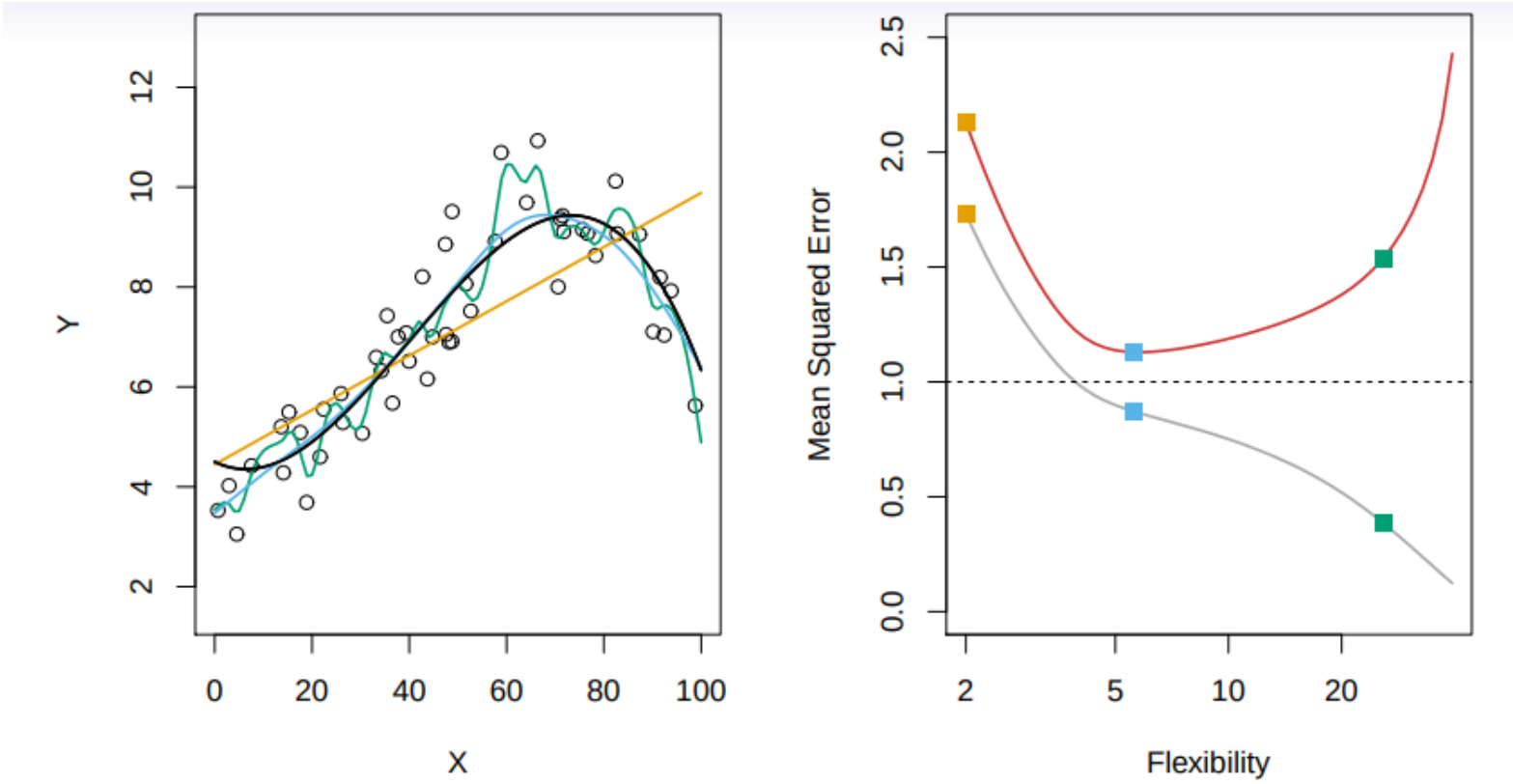
- They correspond to two ML challenges
  - **Underfitting**
    - Inability to obtain low enough error rate on the training set
  - **Overfitting**
    - Gap between training error and testing error is too large
    - We can control whether a model is more likely to overfit or underfit by altering its capacity

Capacity of a model

- Model capacity is ability to **fit variety of functions**
  - Model with Low capacity struggles to fit training set
  - A High capacity model can overfit by memorizing properties of training set which are not useful on test set

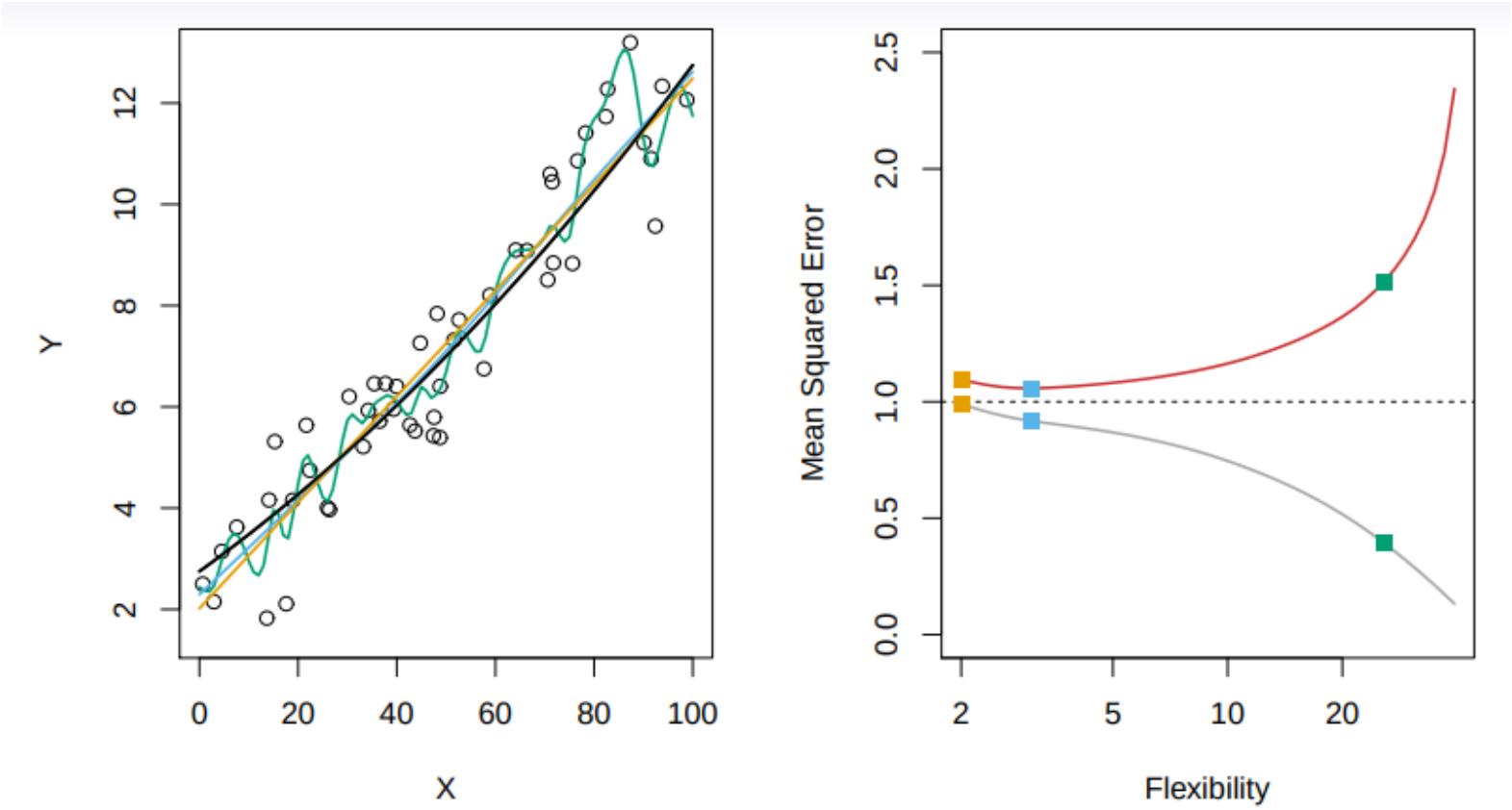
- When a model has higher capacity, it may overfit
  - One way to control capacity of a learning algorithm is by choosing the hypothesis space
    - i.e., set of functions that the learning algorithm is allowed to select as being the solution
    - e.g., the linear regression algorithm has the set of all linear functions of its input as the hypothesis space
  - **Regularization**

Assessing Model Accuracy



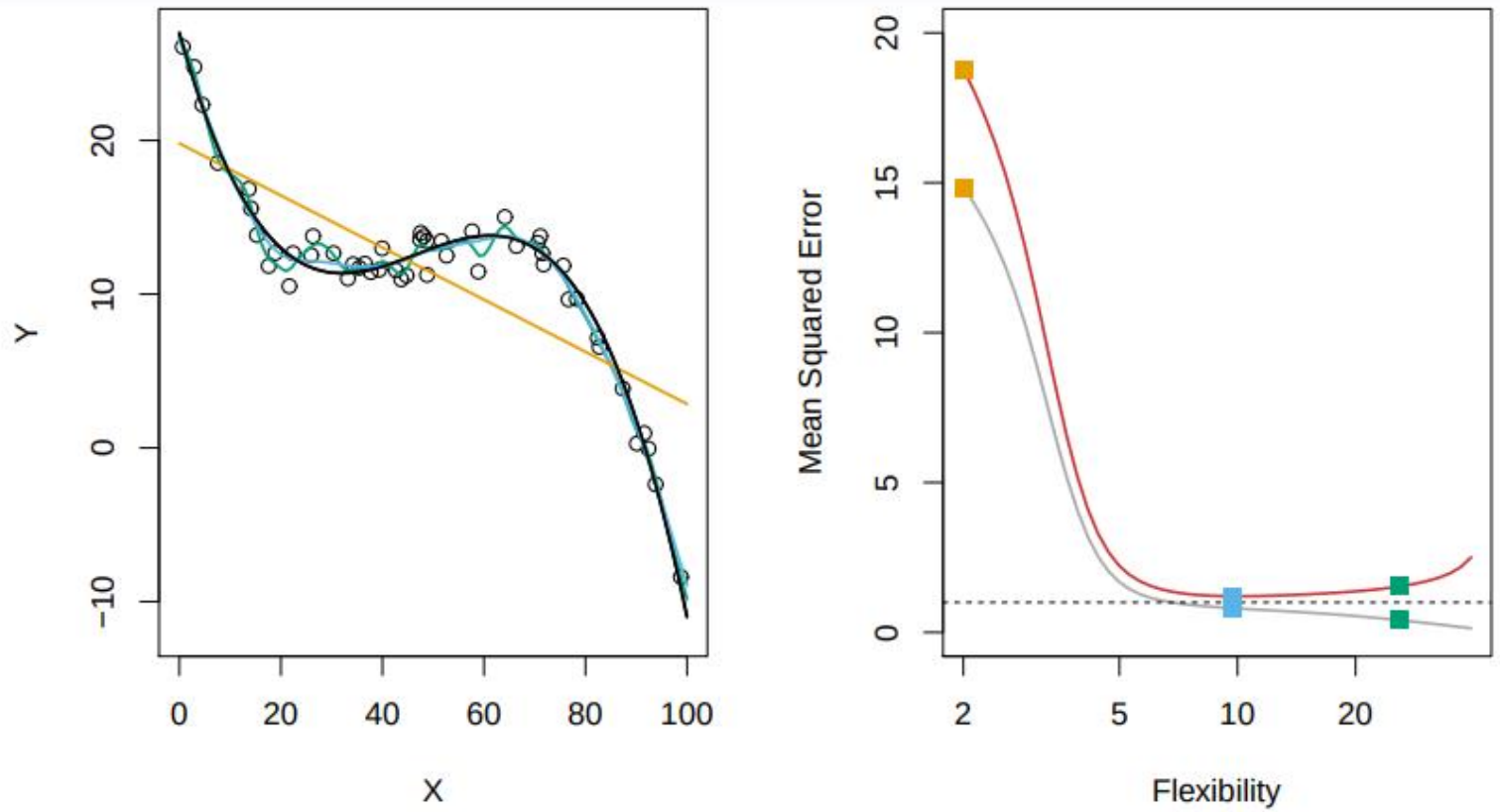
Black curve is truth. Red curve on right is MSE_{Te}, grey curve is MSE_{Tr}. Orange, blue and green curves/squares correspond to fits of different flexibility.

Assessing Model Accuracy



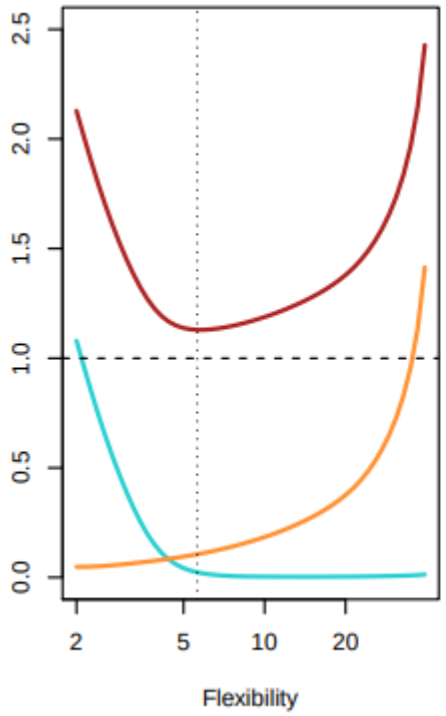
Here the truth is smoother, so the smoother fit and linear model do really well.

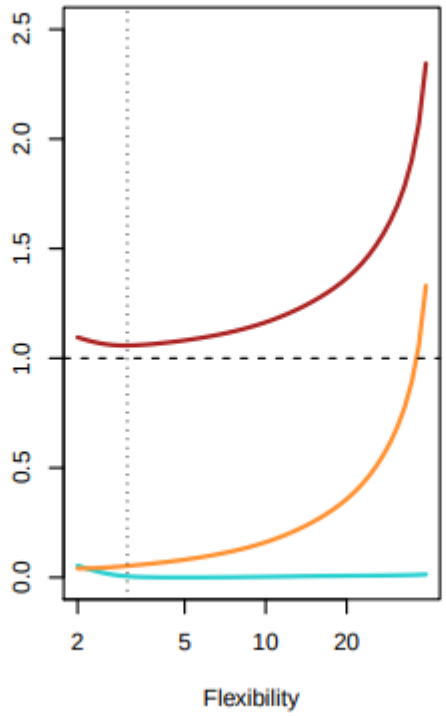
Assessing Model Accuracy

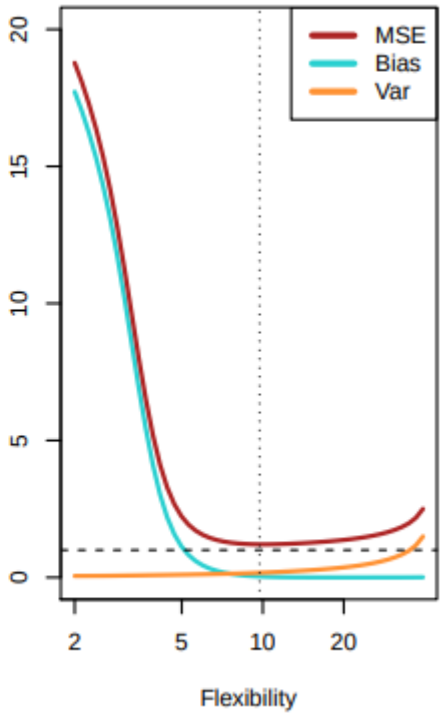


Here the truth is wiggly and the noise is low, so the more flexible fits do the best.

Bias Variance Trade-Off







Appropriate Capacity

- Machine Learning algorithms will perform well when their capacity is appropriate for the true complexity of the task that they need to perform and the amount of training data they are provided with

- Models with insufficient capacity are unable to solve complex tasks

- Models with high capacity can solve complex tasks, but when their capacity is higher than needed to solve the present task, they may overfit

Representational and Effective Capacity

Representational capacity:

- Specifies family of functions learning algorithm can choose from

Effective capacity:

- Imperfections in optimization algorithm can limit representational capacity

Occam's razor:

- Among competing hypotheses that explain known observations equally well, choose the simplest one

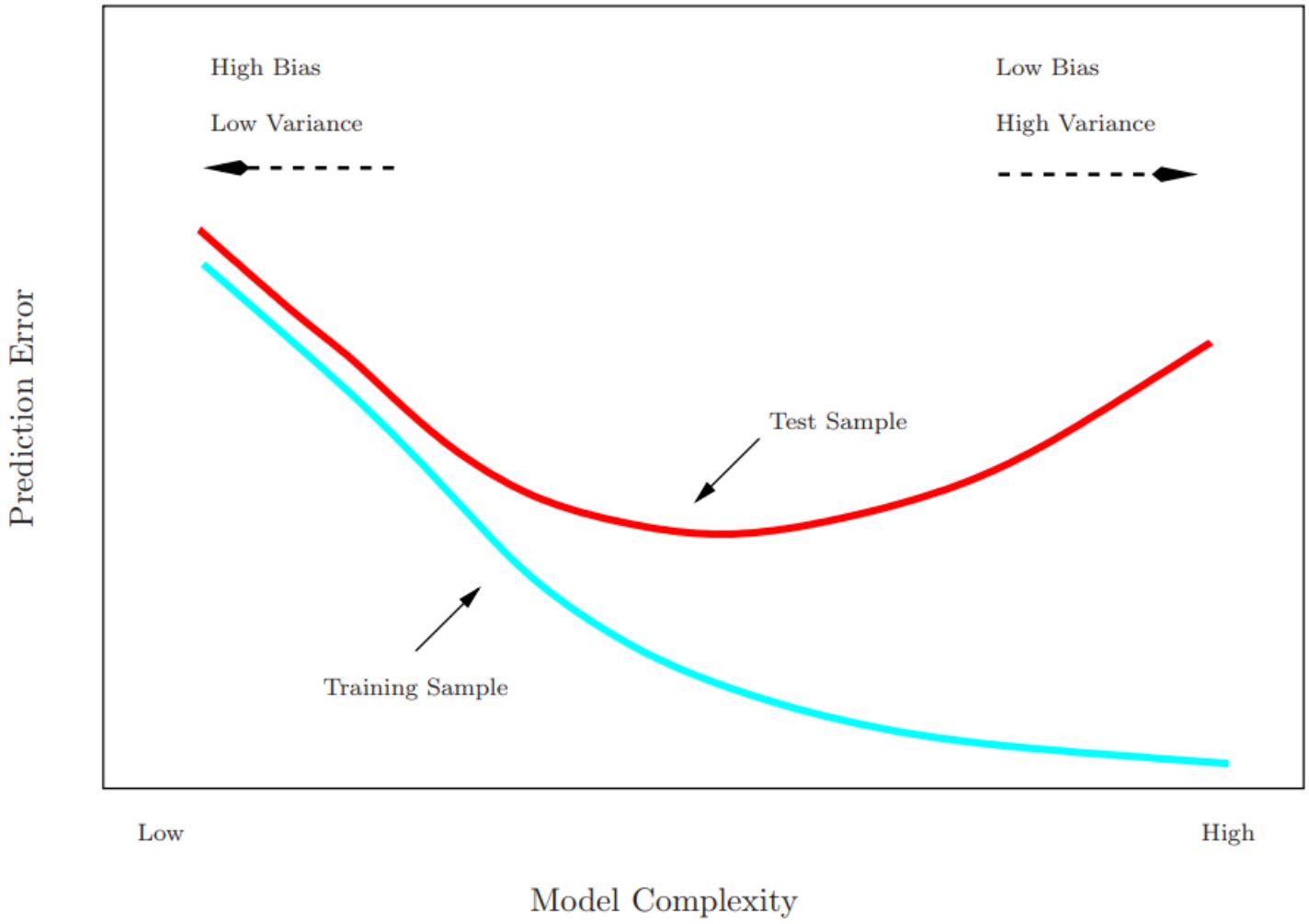
Machine Learning Basics

- Introduction
- Statistical Learning
- Cross-Validation

Training Error versus Test error

- Recall the distinction between the **test error** and the **training error**:
  - The test error is the average error that results from using a statistical learning method to predict the response on a new observation, one that was not used in training the method.
  - In contrast, the training error can be easily calculated by applying the statistical learning method to the observations used in its training.
  - **But the training error rate often is quite different from the test error rate, and in particular the former can dramatically underestimate the latter.**

Training- versus Test-Set Performance



Training- versus Test-Set Performance

- Best solution: a large designated test set.
  - Often not available

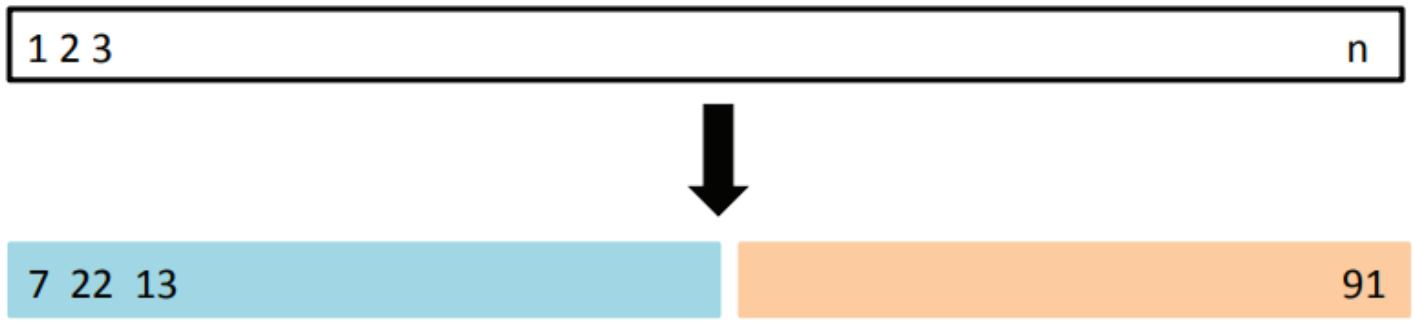
- Some methods make a mathematical adjustment to the training error rate in order to estimate the test error rate.

- Here we instead consider a class of methods that estimate the test error by holding out a subset of the training observations from the fitting process, and then applying the statistical learning method to those held out observations

Validation-set approach

- Here we randomly divide the available set of samples into two parts: a **training set** and a **validation** or **hold-out set**.
- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set.
- The resulting validation-set error provides an estimate of the test error. This is typically assessed using MSE in the case of a quantitative response and misclassification rate in the case of a qualitative (discrete) response.

Validation-set approach



A random splitting into two halves: left part is training set, right part is validation set

Drawbacks of Validation Set Approach

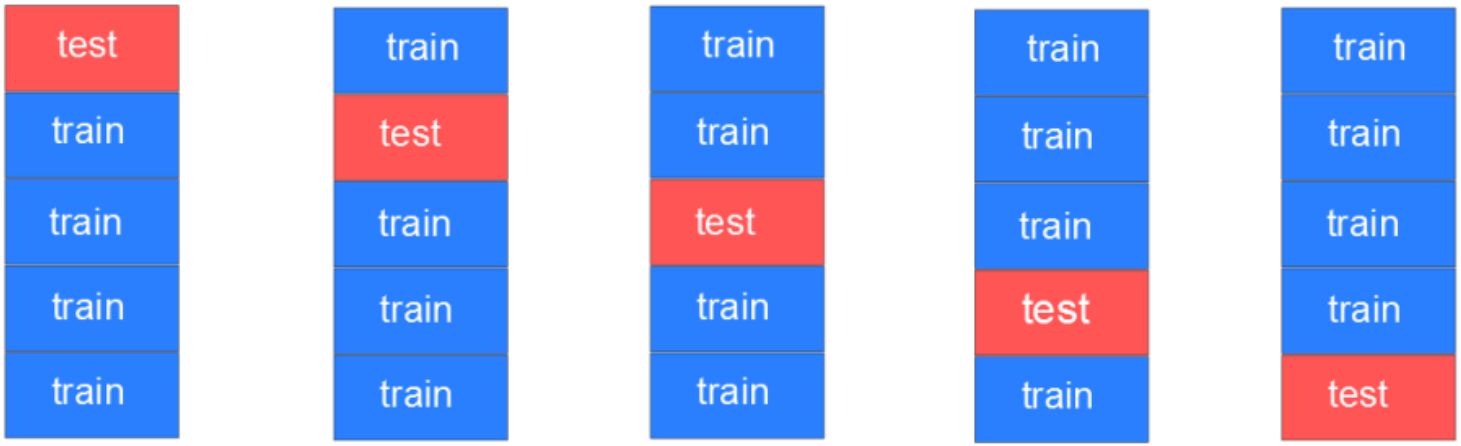
- The validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set.

- In the validation approach, only a subset of the observations - those that are included in the training set rather than in the validation set - are used to fit the model.

- This suggests that the validation set error may tend to overestimate the test error for the model fit on the entire data set.

Cross-Validation

- When data set is too small, a fixed test set is problematic
- k-fold cross-validation:
  - All available data is partitioned into k groups
  - $k-1$ groups are used to train and evaluated on remaining group
  - Repeat for all k choices of held-out group
  - Performance scores from k runs are averaged



The details

- Let the K parts be $C_1, C_2, \ldots, C_K$, where C_k denotes the indices of the observations in part k. There are $n_k = n/K$ observations in part k.
- Compute

$$\mathrm{CV}_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{MSE}_k$$

- Setting $K = n$ yields n-fold or leave-one out cross-validation (LOOCV).
  - LOOCV sometimes useful, but typically doesn't shake up the data enough. The estimates from each fold are highly correlated and hence their average can have high variance.

Some Notes on Cross Validation

- Since each training set is only $(K - 1)/K$ as big as the original training set, the estimates of prediction error will typically be biased upward.
- This bias is minimized when $K = n$ (LOOCV), but this estimate has high variance, as noted earlier.
- $K = 5$ or 10 provides a good compromise for this bias-variance tradeoff.

Cross-validation: right and wrong

- Consider a simple classifier applied to some two-class data:

1. Starting with 5000 predictors (for example genes) and 50 samples, find the 100 predictors having the largest correlation with the class labels.

2. We then apply a classifier such as logistic regression, using only these 100 predictors.

- How do we estimate the test set performance of this classifier?

Cross-validation: right and wrong

- Consider a simple classifier applied to some two-class data:

1. Starting with 5000 predictors (for example genes) and 50 samples, find the 100 predictors having the largest correlation with the class labels.

2. We then apply a classifier such as logistic regression, using only these 100 predictors.

- How do we estimate the test set performance of this classifier?
  - Can we apply cross-validation in step 2, forgetting about step 1?

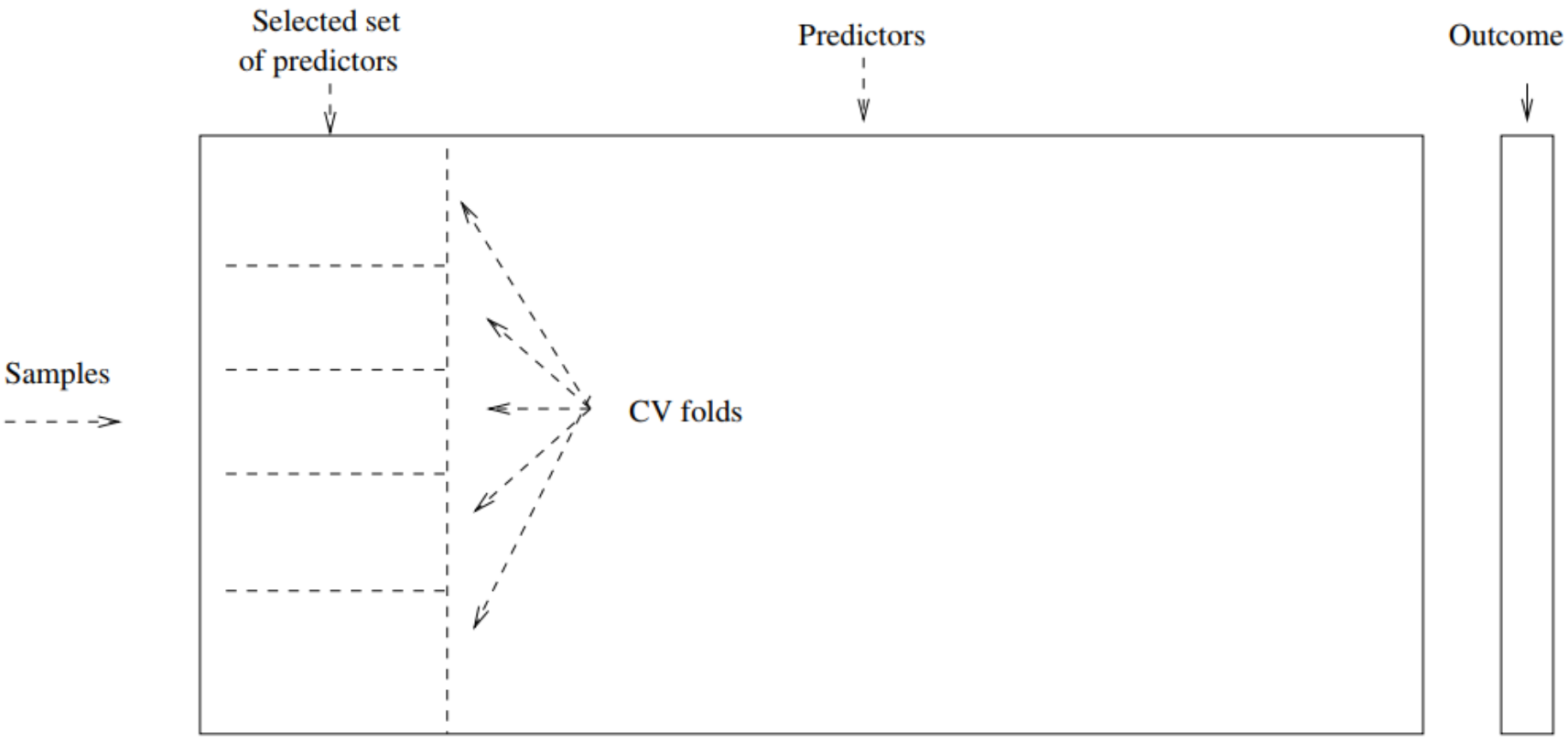
Answer: Hell no!

- This would ignore the fact that in Step 1, the procedure has already seen the labels of the training data, and made use of them. This is a form of training and must be included in the validation process.

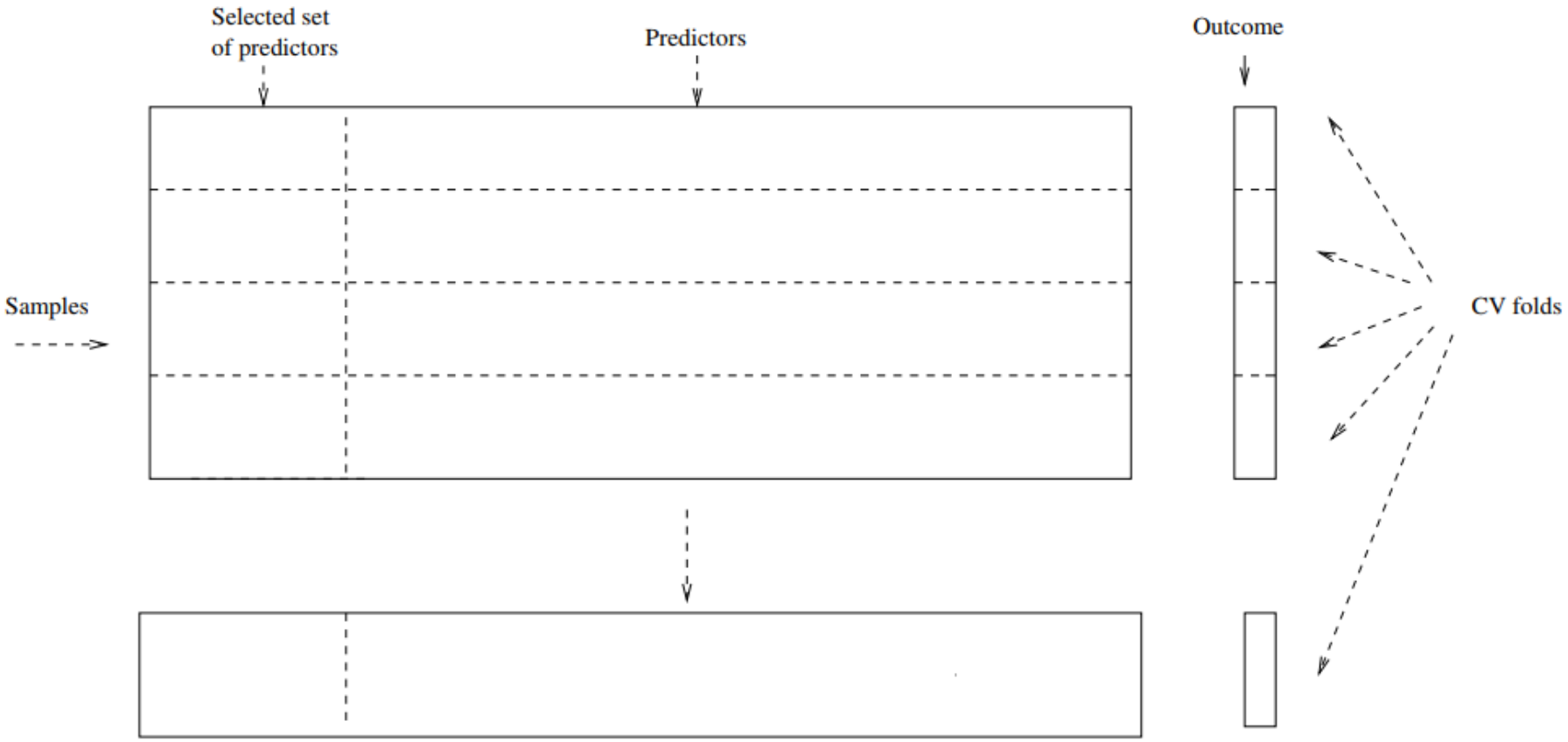
- It is easy to simulate realistic data with the class labels independent of the outcome, so that true test error =50%, but the CV error estimate that ignores Step 1 is zero!

- We have seen this error made in many high profile genomics papers

Wrong (but often used) way



Right Way


Selected set
of predictors
Predictors
Outcome
Samples
CV folds