Statistical Learning for Data Science

Lecture 04

唐晓颖
电子与电气工程系
南方科技大学

February 27, 2023

Statistical Learning VS. Machine Learning

- Machine learning arose as a subfield of *Artificial Intelligence*
- Statistical learning arose as a subfield of *Statistics*
- There is much overlap – both focus on supervised and unsupervised problems:
  - ❑ Machine learning has a greater emphasis on *large scale* applications and *prediction accuracy*.
  - ❑ Statistical learning emphasizes *models* and their interpretability, *precision* and *uncertainty*.
- The distinction has become more and more blurred

Statistical Learning VS. Machine Learning

Statistical Learning:

- Use statistical methods to analyze data and make predictions or decisions
- Focus on building statistical models to describe the relationships between variables in a dataset
- Used in fields such as economics, psychology, and social sciences, where understanding the underlying statistical relationships is important

Machine Learning:

- A more modern approach that uses algorithms to automatically learn patterns and relationships in data
- Designed to automatically improve the algorithm performance
- Used to make predictions or decisions in a wide range of applications, such as image recognition, natural language processing, and self-driving cars

Statistical Learning VS. Machine Learning

https://blogs.perficient.com/2018/01/29/machine-learning-vs-statistical-learning/

https://www.jiqizhixin.com/articles/2019-05-06-13

哪种方法更好?

其实这是个很蠢的问题。从关系角度看，没有统计学，机器学习是不存在的。然而，在当前人类所经历的这个信息爆炸的时代中，面对海量数据的涌入，机器学习倒是颇为有用。

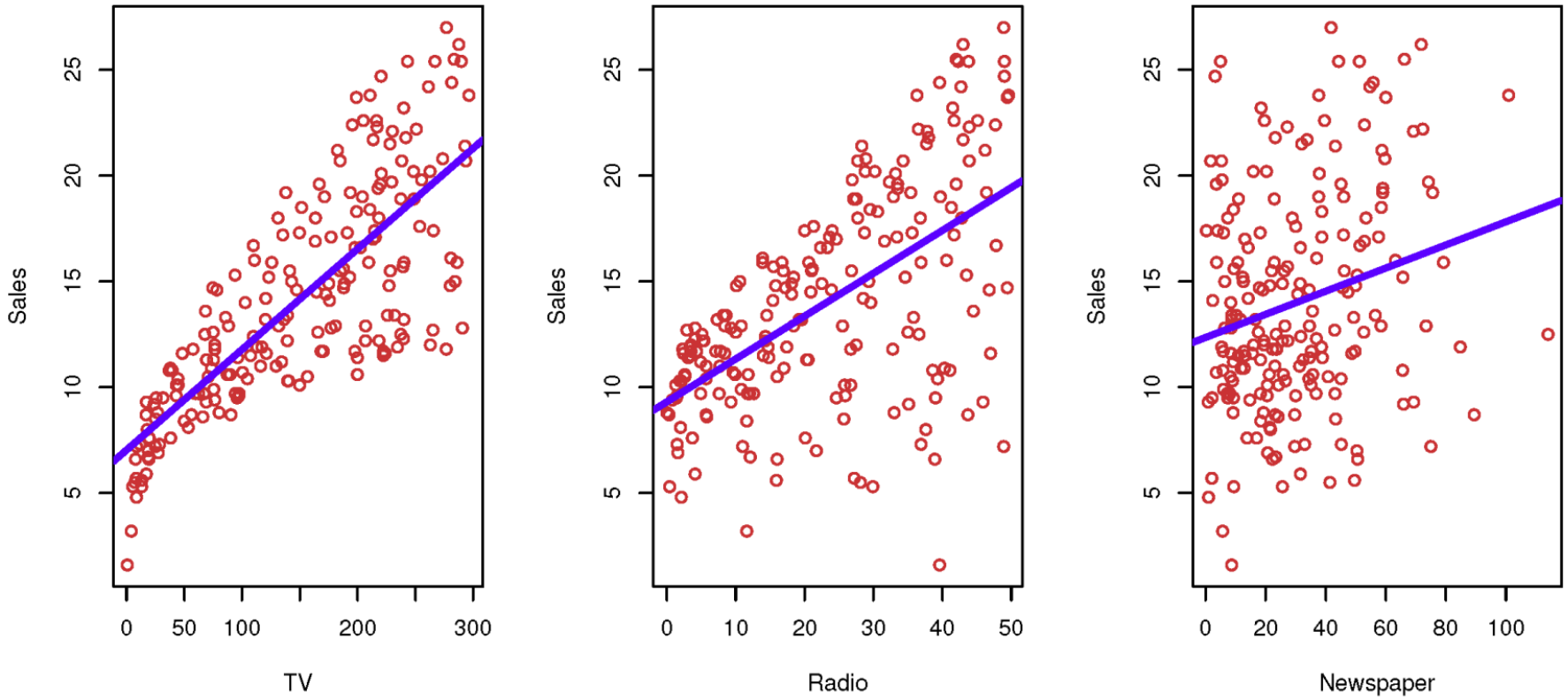
比较机器学习和统计模型确实有些困难。这主要取决于你的目的是什么。如果你想构建一种可以精确预测房价的算法，或是使用数据确定某人是否可能感染某种疾病的话，机器学习可能是更好的选择。如果你想证明变量间的关系或用数据进行推断，那么统计模型则会成为更好的选择。

Here are some of the differences:

1. Both methods are data dependent. However, Statistical Learning relies on rule-based programming; it is formalized in the form of relationship between variables, where Machine Learning learns from data without explicitly programmed instructions.
2. Statistical Learning is based on a smaller dataset with a few attributes, compared to Machine Learning where it can learn from billions of observations and attributes.
3. Statistical Learning operates on assumptions, such as normality, no multicollinearity, homoscedasticity, etc. when Machine Learning is not as assumptions dependent and in most of the cases ignores them.
4. Statistical Learning is mostly about inferences, most of the idea is generated from the sample, population, and hypothesis, in comparison to Machine Learning which emphasizes predictions, supervised learning, unsupervised learning, and semi-supervised learning.
5. Statistical Learning is math intensive which is based on the coefficient estimator and requires a good understanding of your data. On the other hand, Machine Learning identifies patterns from your dataset through the iterations which require a way less of human effort.

Statistical Learning Example

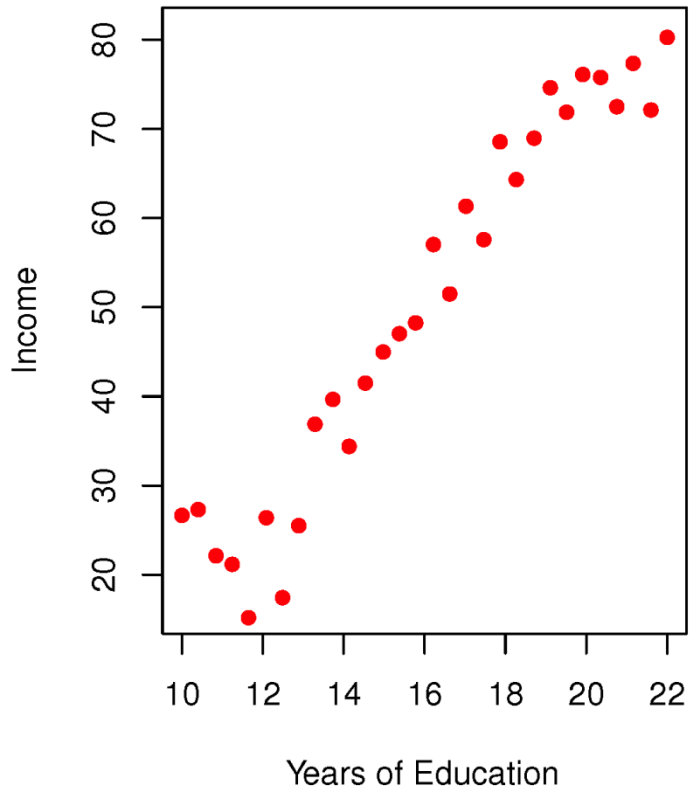
- Provide advice on how to improve sales of a particular product



Can we predict *Sales* using *TV*, *Radio*, and *Newspaper*?

Statistical Learning Example

- Understand the relationship between income and education



Can we predict *Income* using *Years of Education*?

Notation

Y : *Sales or Income* (output/response)

X_1: *TV or Years of Education* (input/predictor)

X_2: *Radio or …* (input/predictor)

X_3: *Newspaper or …* (input/predictor)

.

.

.

We can refer to the *input vector* collectively as:

$$X = \left(X_1, X_2, ..., X_p \right)$$

p : the total number of inputs or predictors

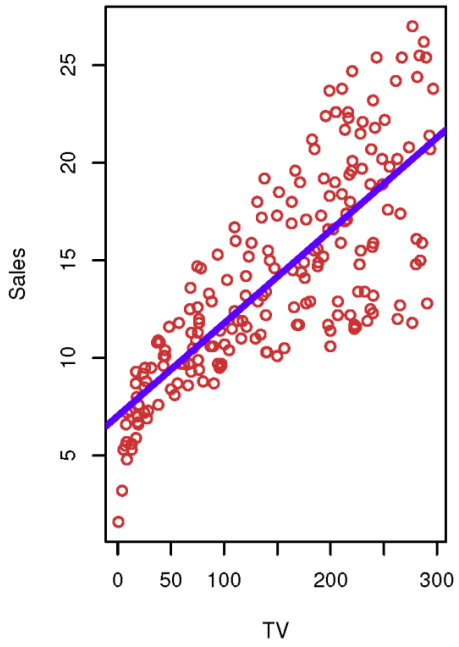
Model

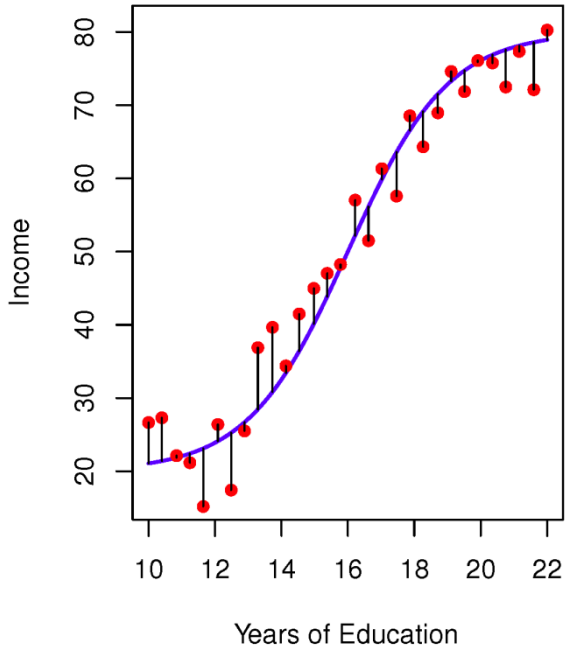
$$Y = f(X) + \varepsilon$$

f : some fixed but unknown function

ε : a random *error term* (independent of the input, has mean zero)

f represents the systematic information that X provides about Y





In essence, statistical learning refers to a set of approaches for estimating f

Why Estimate f

Two important goals in statistical learning: prediction and inference

Prediction:

- Task of making predictions about future observations based on a set of input-output pairs
- Goal: to create a model that accurately predicts the output for new input values that were not present in the training data
- Models are typically evaluated based on their ability to accurately predict the output for new data

Why Estimate f

Two important goals in statistical learning: prediction and inference

Inference:

- Task of understanding the relationship between the input and output variables in a statistical model
- Goal: to identify which variables are important in predicting the output, and to understand the strength and direction of the relationships between variables
- Models are typically evaluated based on their ability to provide insights into the underlying mechanisms that generate the data

Why Estimate f

Two important goals in statistical learning: prediction and inference

Main difference:

- Prediction is focused on accurately predicting the output for new data
- Inference is focused on understanding the relationships between the variables in the model
- Prediction models prioritize accuracy over interpretability, while inference models prioritize interpretability over accuracy

Why Estimate f

1. Prediction

   - With a good one we can make predictions of Y at new points $X = x$

     $\hat{Y} = \hat{f}(X)$

     $\hat{f}$: our estimation of f

     $\hat{Y}$: the resulting prediction for Y

     Example: predict a patient's risk for a severe adverse reaction to a particular drug based on his/her blood sample characteristics.

Why Estimate f

1. Prediction
   - The accuracy of $\hat{Y}$ is very important
     - *Reducible error*

       Error induced by the estimation of f

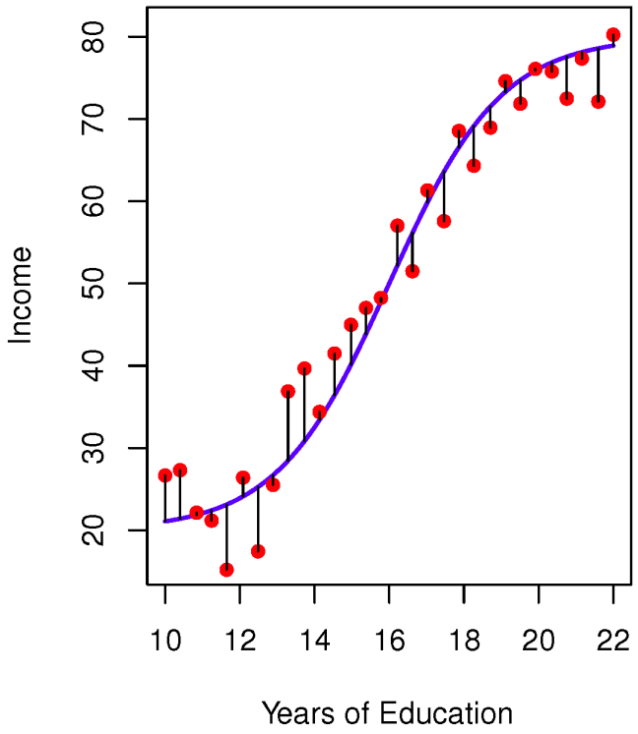
     - *Irreducible error*

       Error induced by ε

Model

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

ε : a random *error term* (independent of the input, has mean zero)



The vertical lines represent the error terms ε

The random error is caused by some uncontrollable reasons, such as the environment, the temperature, when real data is generated. It can be positive and negative. It is not measurable.

Why Estimate f

1. Prediction

   The mean squared error (MSE)

$$E(Y-\hat{Y})^2 = E\left[f(X)+\varepsilon-\hat{f}(X)\right]^2$$

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

$$\begin{aligned}\text{Var}(X) &= \text{E}\left[(X-\text{E}[X])^2\right] \\ &= \text{E}\left[X^2 - 2X\,\text{E}[X] + \text{E}[X]^2\right] \\ &= \text{E}\left[X^2\right] - 2\,\text{E}[X]\,\text{E}[X] + \text{E}[X]^2 \\ &= \text{E}\left[X^2\right] - \text{E}[X]^2\end{aligned}$$

$$\rightarrow = E\left\{\left[f(x)-\hat{f}(x)\right]^2 + 2\varepsilon\left[f(x)-\hat{f}(x)\right] + \varepsilon^2\right\}$$

$$= E\left\{\left[f(x)-\hat{f}(x)\right]^2\right\} + E\left\{2\varepsilon\left[f(x)-\hat{f}(x)\right]\right\} + E(\varepsilon^2)$$

$$= \left[f(x)-\hat{f}(x)\right]^2 + 0 + E(\varepsilon^2) = \left[f(x)-\hat{f}(x)\right]^2 + \text{Var}(\varepsilon)$$

Why Estimate f

1. Prediction

   - Our goal is to provide techniques for estimating f with the aim of minimizing the reducible error.

   - The irreducible error will always provide an upper bound on the accuracy of our prediction. This bound is almost always unknown in practice.

   - The prediction accuracy is the most important. We may not care the exact format/structure of $\hat{f}$.

Why Estimate f

2. Inference

Understand how Y changes as a function of $X_1, X_2, ..., X_p$

Now $\hat{f}$ cannot be treated as a black box, we need to know its exact form

- *Which predictors are associated with the response?*

  Identifying the few important predictors among a large set of possible variables can be extremely useful

- *The relationship between the response and each predictor*

  Some predictors may have a positive relationship with the output while some other may have a negative relationship. The relationship between the response and a given predictor may also depend on the values of the other predictors.

Why Estimate f

Prediction example

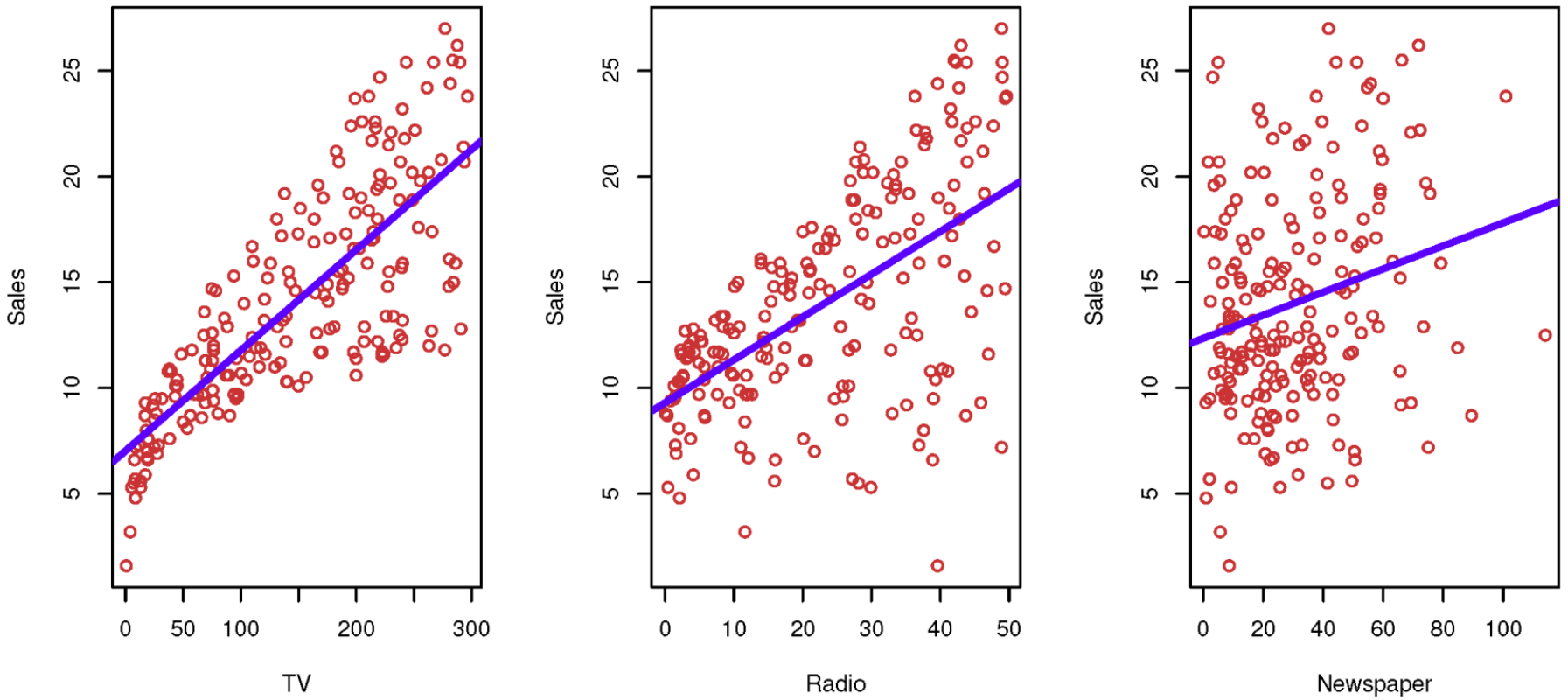
Identify individuals who will respond positively to a mailing, based on observations of demographic variables measured on each individual.

-- *Not interested in obtaining a deep understanding of the relationships between each individual predictor and the response.*

-- *Simply wants an accurate model to predict the response using the predictors.*

Why Estimate f

Inference example



-- Which media contribute to sales?

-- Which media generate the biggest boost in sales?

Why Estimate f

Prediction + Inference example

Relate values of homes to inputs such as crime rate, zoning, distance from a river, air quality, schools, income level of community, size of houses, and so forth.

-- *Interested in how the individual input variables affect the prices ("how much extra will a house be worth if it has a view of the river?")* Inference

-- *Interested in predicting the value of a home given its characteristics ("is this house under- or over- valued?")* Prediction

Why Estimate f

Depending on our ultimate goal (prediction, inference, or a combination of the two), different methods of estimating f may be appropriate.

Example: linear models allow for relatively simple and interpretable inferences but may not yield very accurate predictions. In contrast, some of the highly non-linear approaches can potentially provide quite accurate predictions, but this comes at the expense of a less interpretable model for which inference is more challenging.

How Do We Estimate f

Three key elements for statistical learning: Method = model + optimization + algorithm

Model

- A mapping from input to output
- Two forms of models:
  - Probability model (conditional probability distribution $P(Y|X)$)
  - Non-probability model (decision function $Y = f(X)$)
  - Set of conditional probabilities: $\mathcal{F} = \{P \mid P(Y \mid X)\}$
  - Set of decision functions: $\mathcal{F} = \{f \mid Y = f(X)\}$

How Do We Estimate f

Three key elements for statistical learning: Method = model + optimization + algorithm

Optimization

- Consider what criteria to learn or choose the best model.
- Loss function: evaluation of a prediction

$$L(Y, f(X)) = \begin{cases} 1, & Y \neq f(X) \\ 0, & Y = f(X) \end{cases}$$

0-1 loss function

$$L(Y, f(X)) = (Y - f(X))^2$$

Quadratic loss function

$$L(Y, f(X)) = |Y - f(X)|$$

Absolute loss function

How Do We Estimate f

Three key elements for statistical learning: Method = model + optimization + algorithm

Algorithm

- Algorithms for solving optimization problems
- If the optimization problem has an explicit analytical formula, the algorithm is relatively simple
- However, the analytical formula usually does not exist, so the numerical calculation method is needed

How Do We Estimate f

Two broad categories of statistical learning methods: parametric and non-parametric methods

Parametric methods:

- Make assumptions about the underlying distribution of the data
- Use a fixed set of parameters to describe the distribution
- The parametric model has a fixed form and the parameters are learned from the data
- Examples: linear regression, logistic regression, and the Naive Bayes classifier

Non-parametric methods:

- Use flexible models that can adapt to the data
- Do not make assumptions about the underlying distribution
- Examples: decision trees, random forests, and support vector machines

How Do We Estimate f

- Parametric Methods (model-based approach)

  1. Make an assumption about the functional form, or shape, of f

     Example: linear model

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

     Instead of having to estimate an entirely arbitrary p-dimensional function $f(X)$, one only needs to estimate the coefficients $\beta_0, \beta_1, \ldots, \beta_p$

How Do We Estimate f

- Parametric Methods (model-based approach)

  2. Use the training data $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ to fit or train the model

$$
\begin{cases}
y_1 = \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + ... + \beta_p x_{1p} \\
y_2 = \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + ... + \beta_p x_{2p} \\
\quad . \\
\quad . \\
\quad . \\
y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + ... + \beta_p x_{np}
\end{cases}
$$

The most common approach is least squares

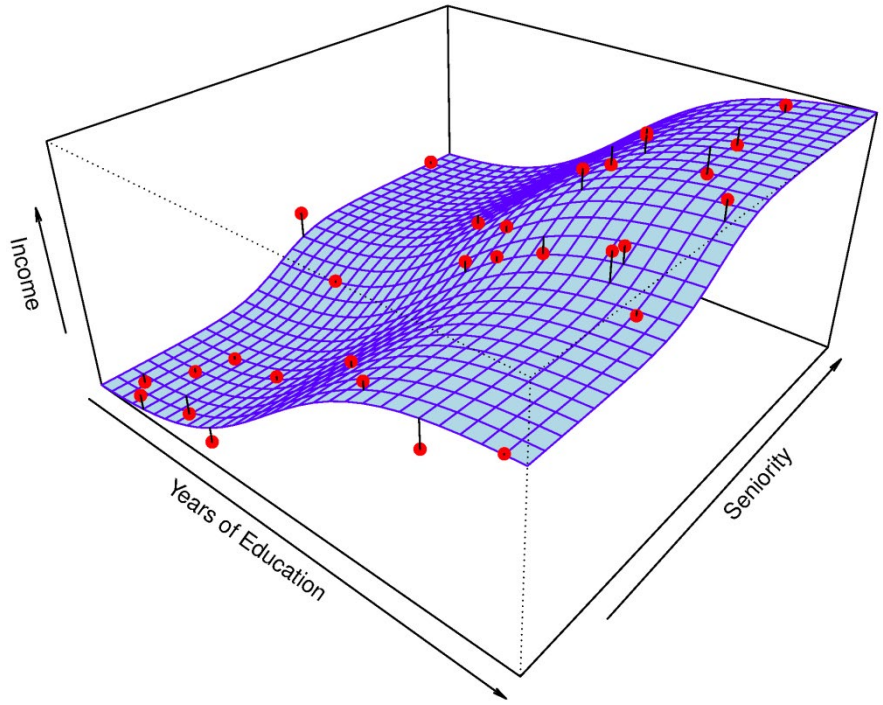
How Do We Estimate f

- Parametric Methods (model-based approach)

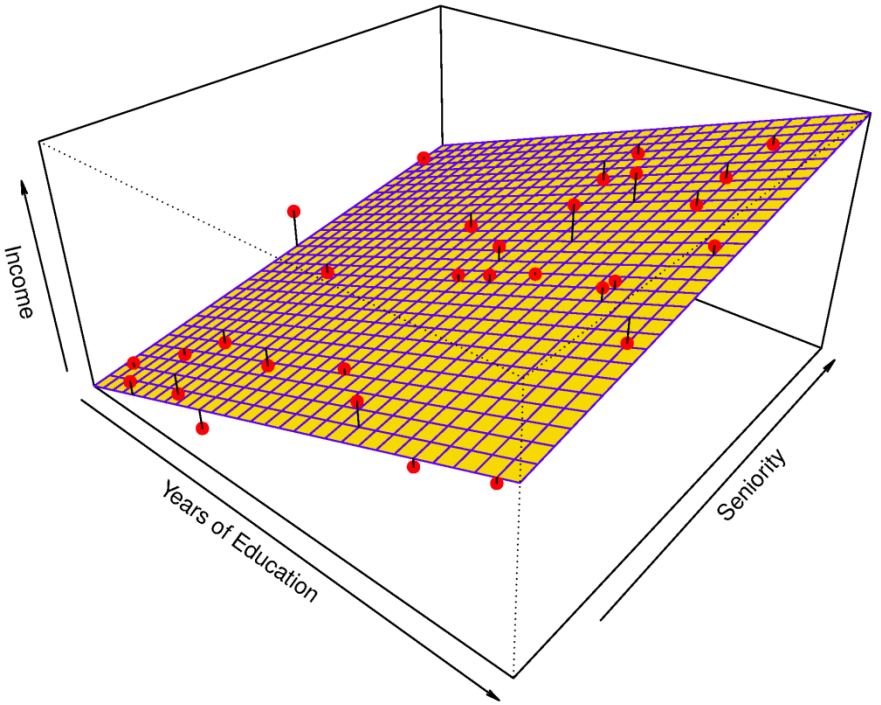
  Pro: Simplifies the estimation problem

  Con: Model choosing is very important

True function



Estimated function using linear model



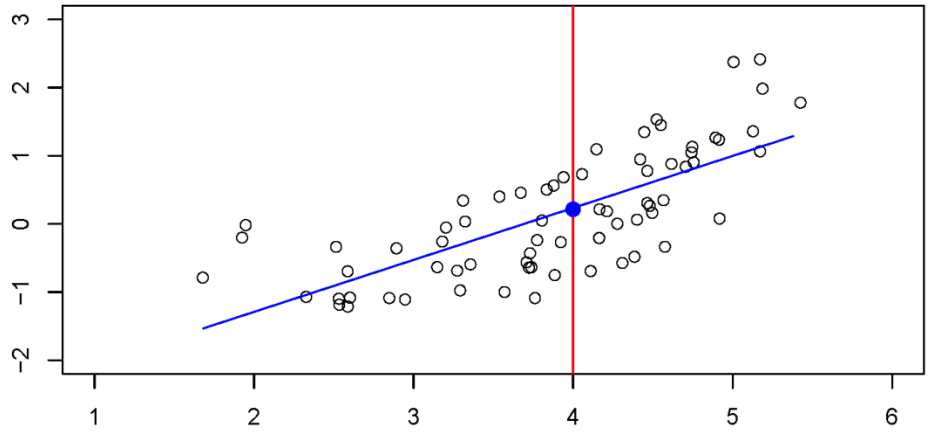
How Do We Estimate f

- Parametric Methods (model-based approach)

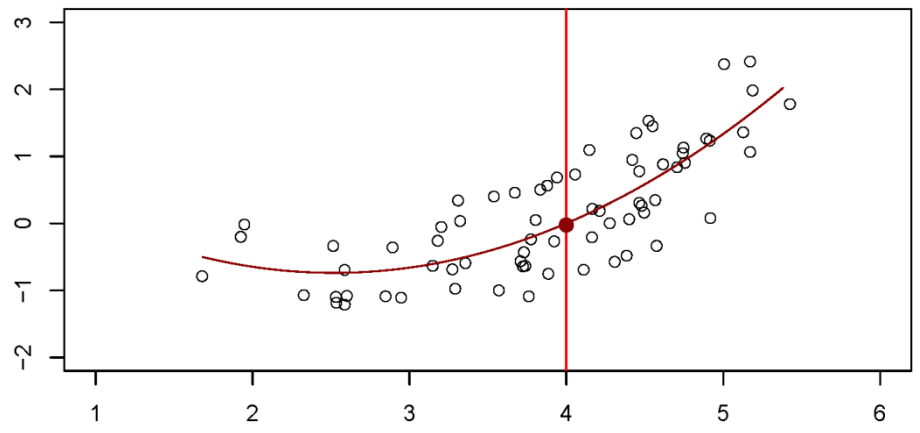
  Pro: Simplifies the estimation problem

  Con: Model choosing is very important

Linear model fitting



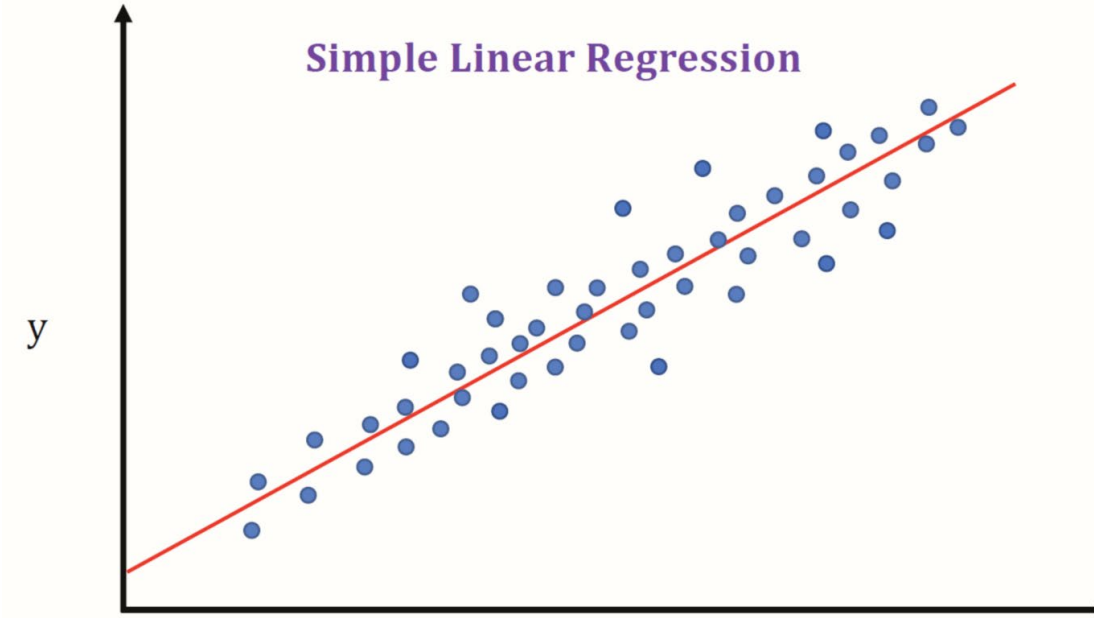
Quadratic model fitting



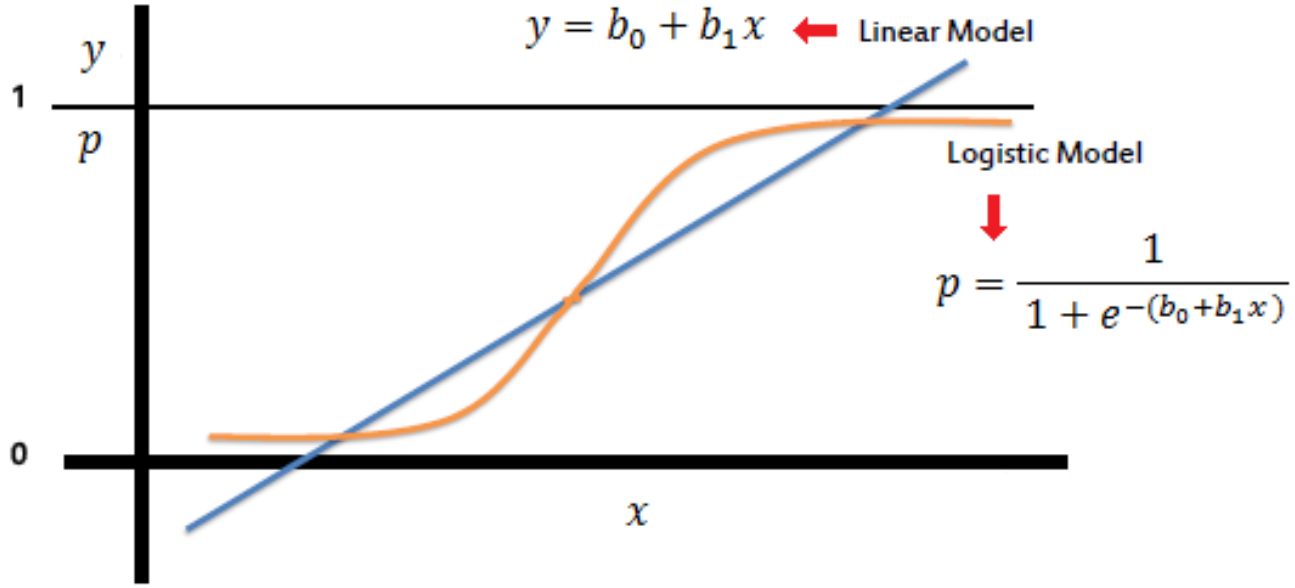
Although it is almost never correct, a linear model often serves as a good and interpretable approximation to the unknown true function

How Do We Estimate f

- Examples of Parametric Methods



Linear regression



Logistic regression

How Do We Estimate f

- Non-parametric Methods
  - No explicit assumptions about the functional form of f
  - Seek an estimate that gets as close to the data points as possible without being too rough or wiggly

Pro: Have the potential to accurately fit a wider range of possible shapes for f

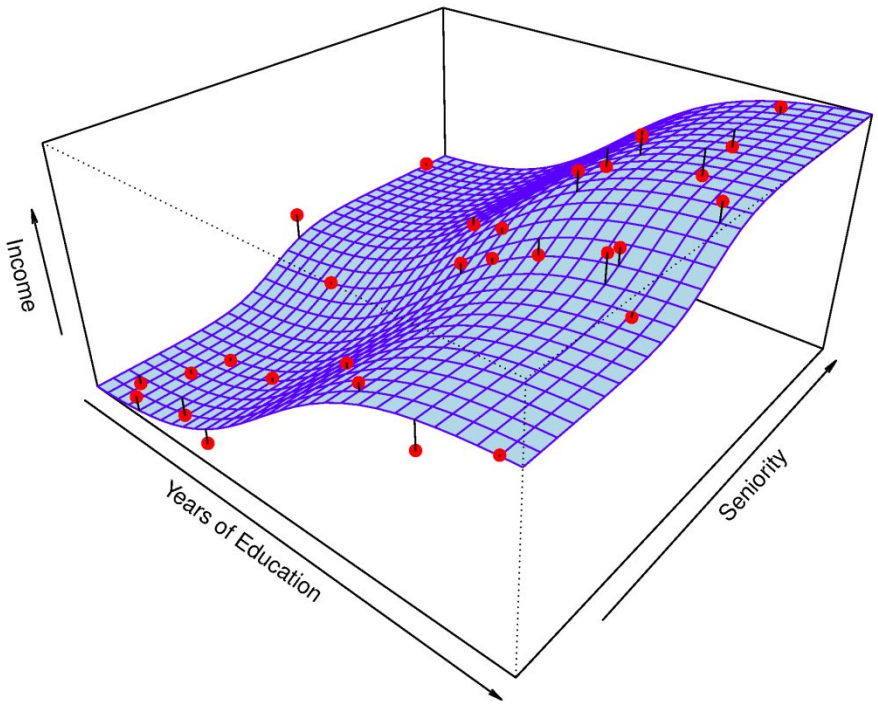
Con: A very large number of observations (far more than is typically needed for a parametric approach) is required in order to obtain an accurate estimate;

May lead to *overfitting* (fit the training data perfectly, but will not yield accurate estimate of the response on new observations that were not part of the original training dataset);

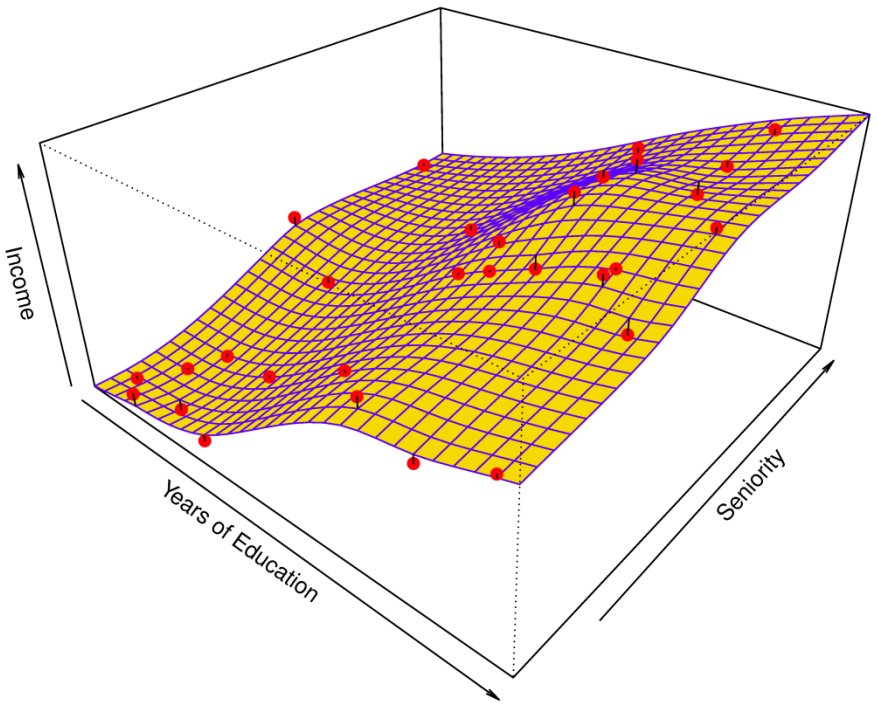
How Do We Estimate f

- Non-parametric Methods

True function

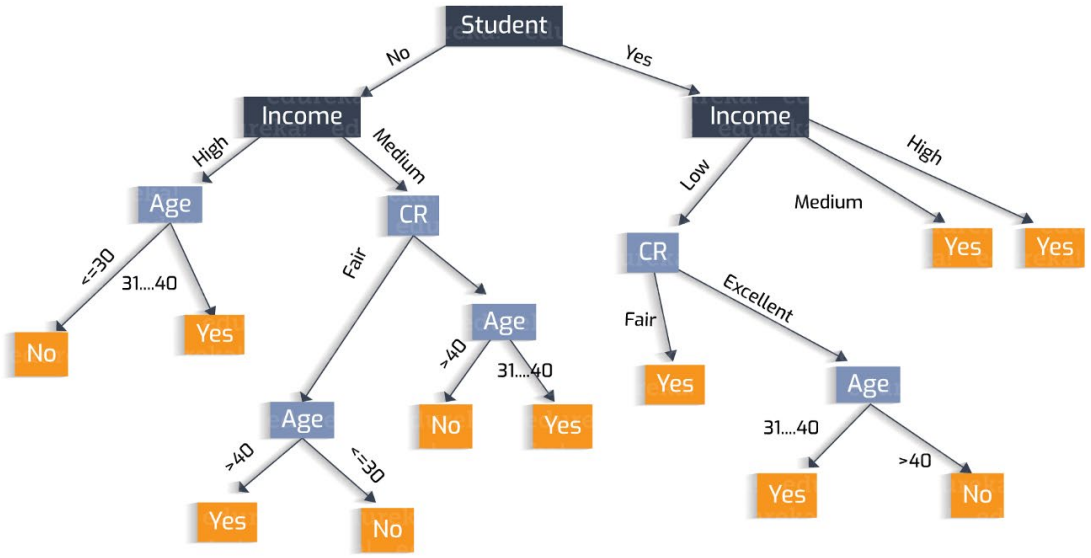


Non-parametrically estimated function

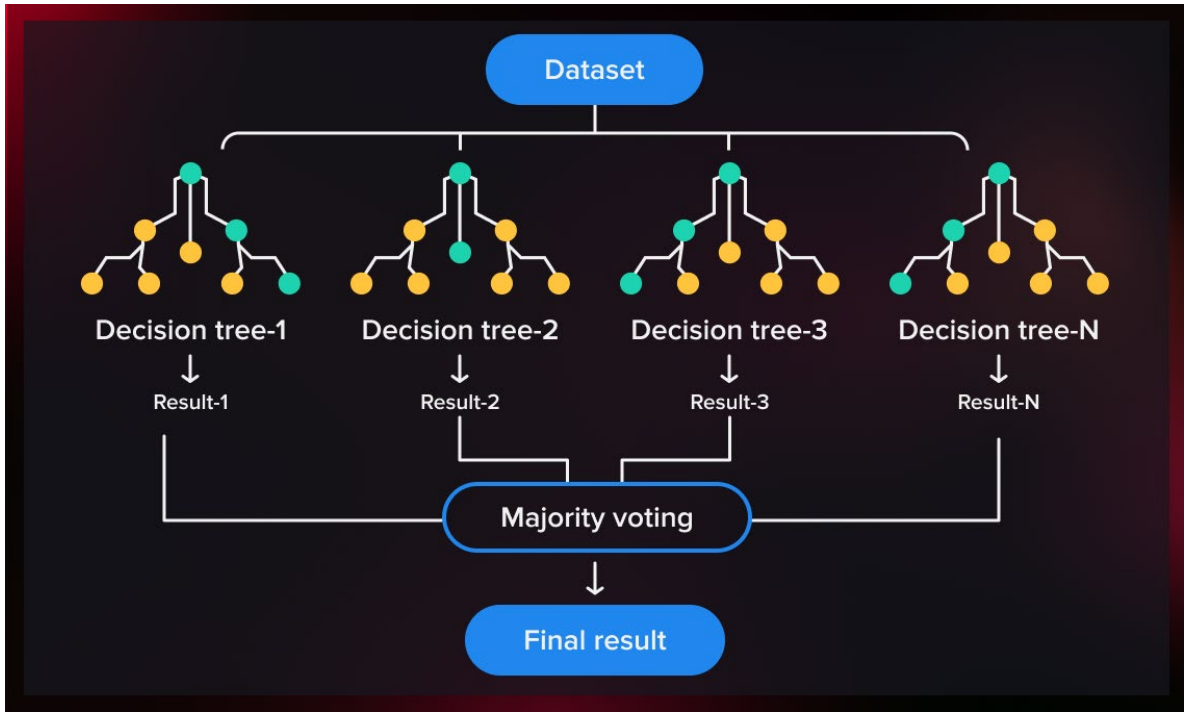


How Do We Estimate f

- Examples of Non-parametric Methods



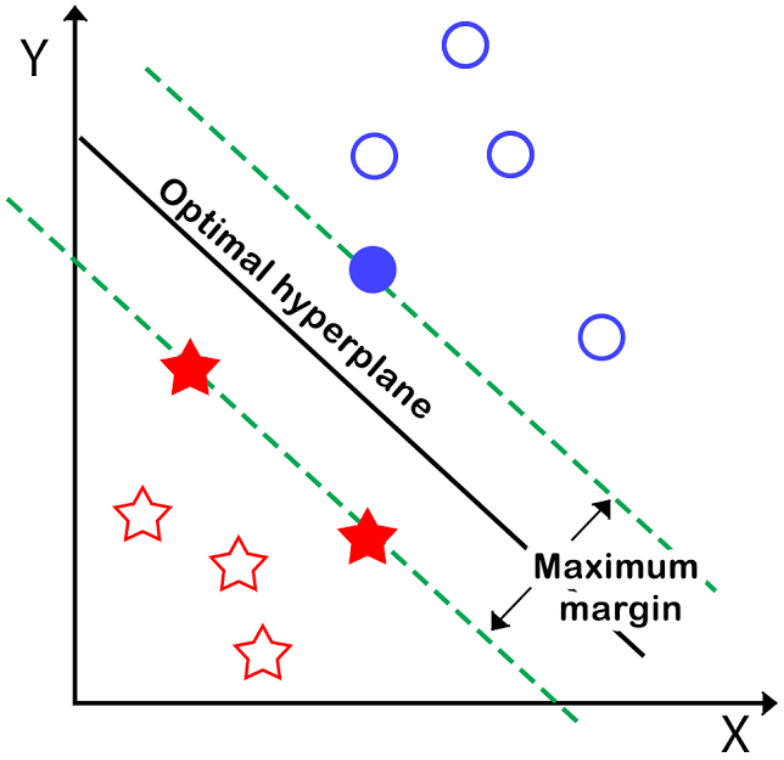
Decision tree



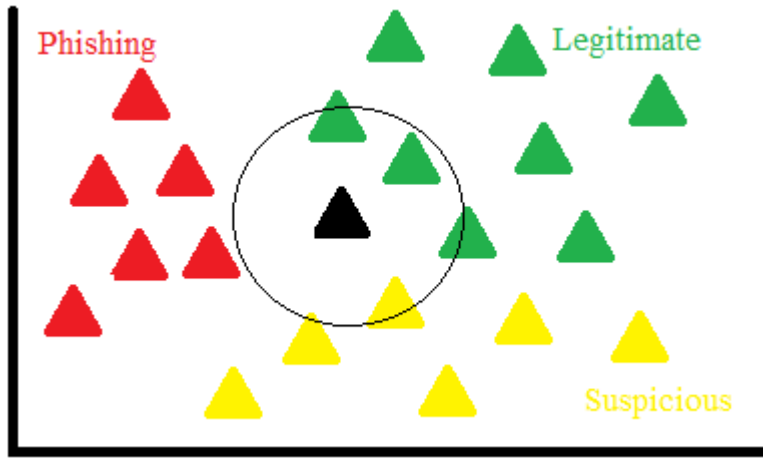
Random forest

How Do We Estimate f

- Examples of Non-parametric Methods



Support vector machine



K-nearest neighbor

Prediction Accuracy versus Model Interpretability

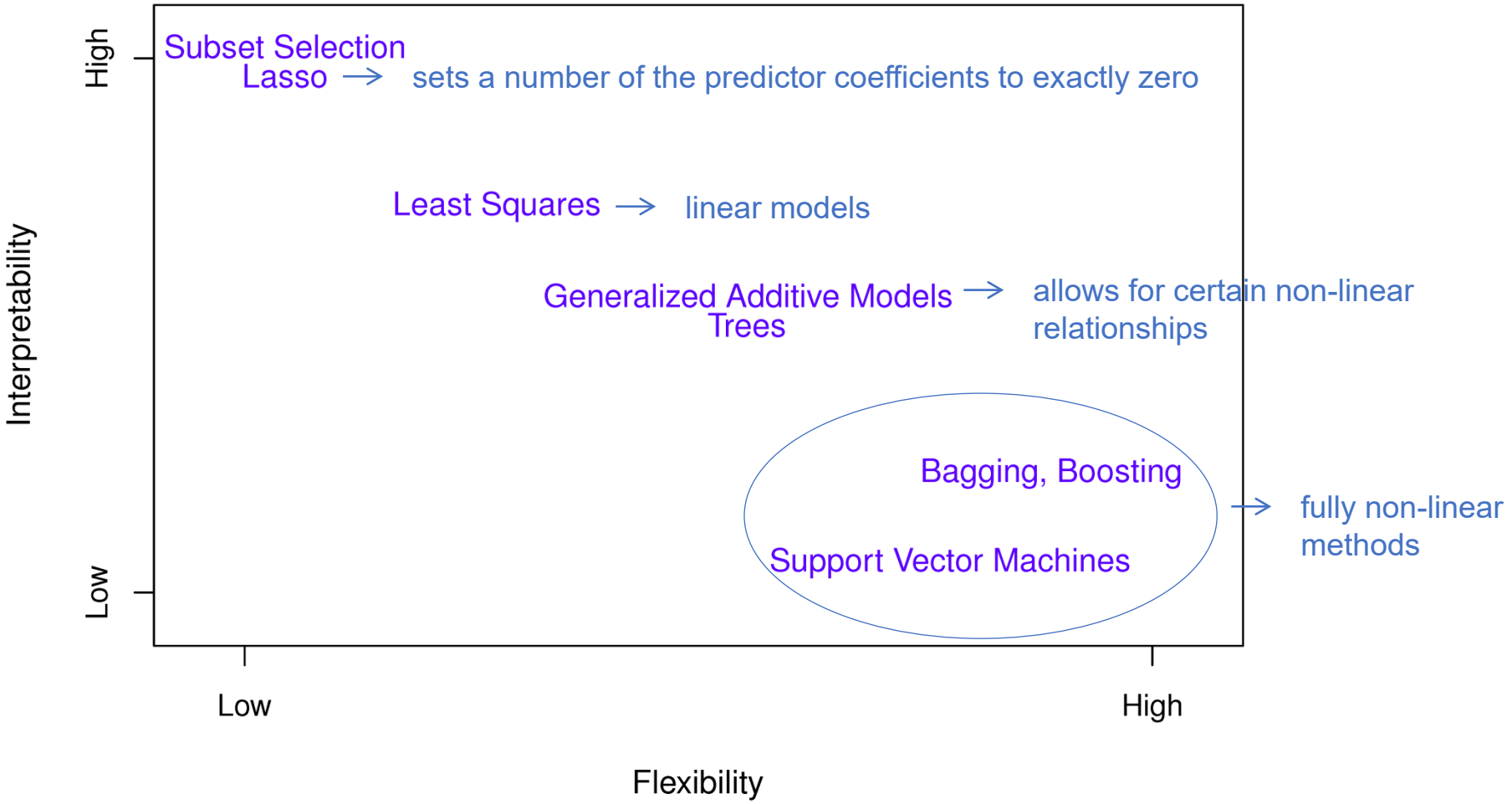
(flexible methods versus restricted methods)

- Flexible methods can generate a much wider range of possible shapes to estimate f than restricted methods do
- Restrictive models are much more interpretable than flexible ones in inference

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

Prediction Accuracy versus Model Interpretability

(flexible methods versus restricted methods)



Assessing Model Accuracy

Why?

- *There is no free lunch in statistics*: no one method dominates all others over all possible data sets

- For a given set of data, selecting the best approach can be one of the most challenging parts of performing statistical learning in practice.

Assessing Model Accuracy

How?

- *Measuring the Quality of Fit*

  How close the predicted response value is close to the true response value

  In regression, use the *mean squared error* (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

Assessing Model Accuracy

How?

- *Measuring the Quality of Fit*

$$MSE_{train} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2 \qquad MSE_{test} = Ave\left(y_0 - \hat{f}(x_0)\right)^2$$

Train a regression model based on the training data may induce *over-fitting*; a method with the training set MSE small can have a large testing set MSE.

Reason for over-fitting: the statistical learning procedure is working too hard to find patterns in the training data, and may be picking up some patterns that are just caused by random chance rather than by true properties of the unknown function.

Assessing Model Accuracy

How?

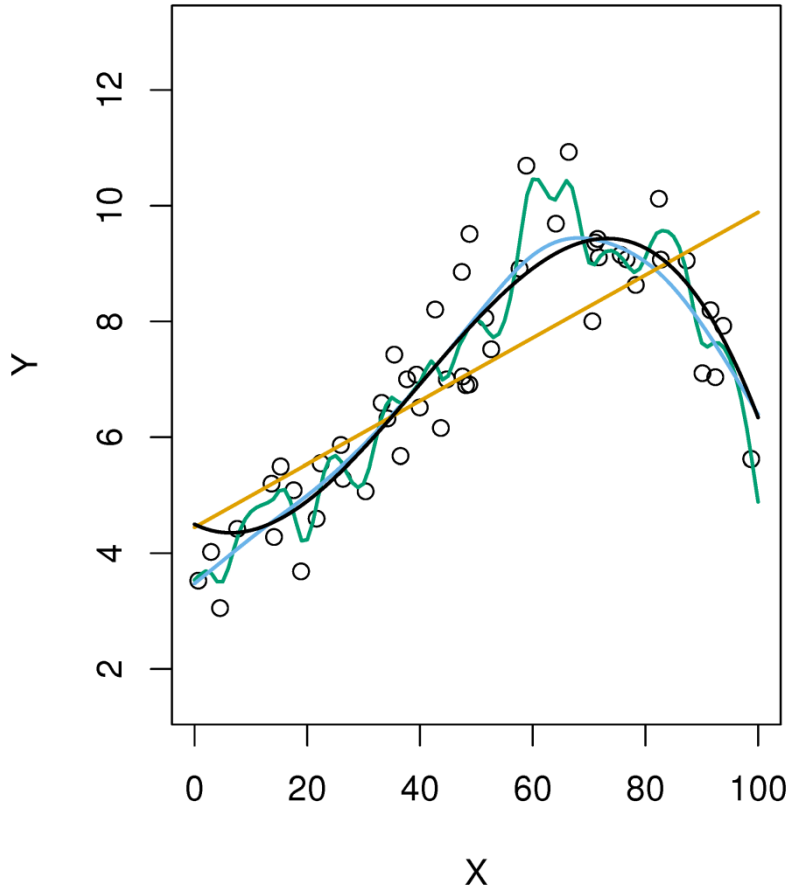
- *Measuring the Quality of Fit*

  Over-fitting

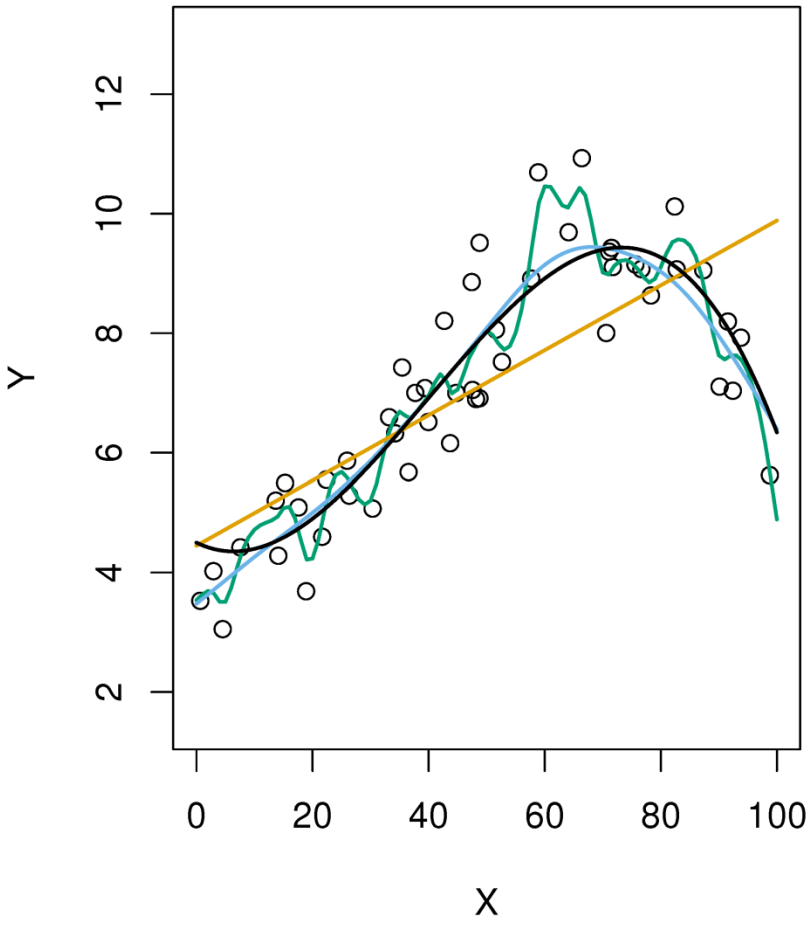
  --open circle : observations
  --orange line: linear regression
  --black line: true function
  --green line: non-linear fitting
  --blue line: non-linear fitting

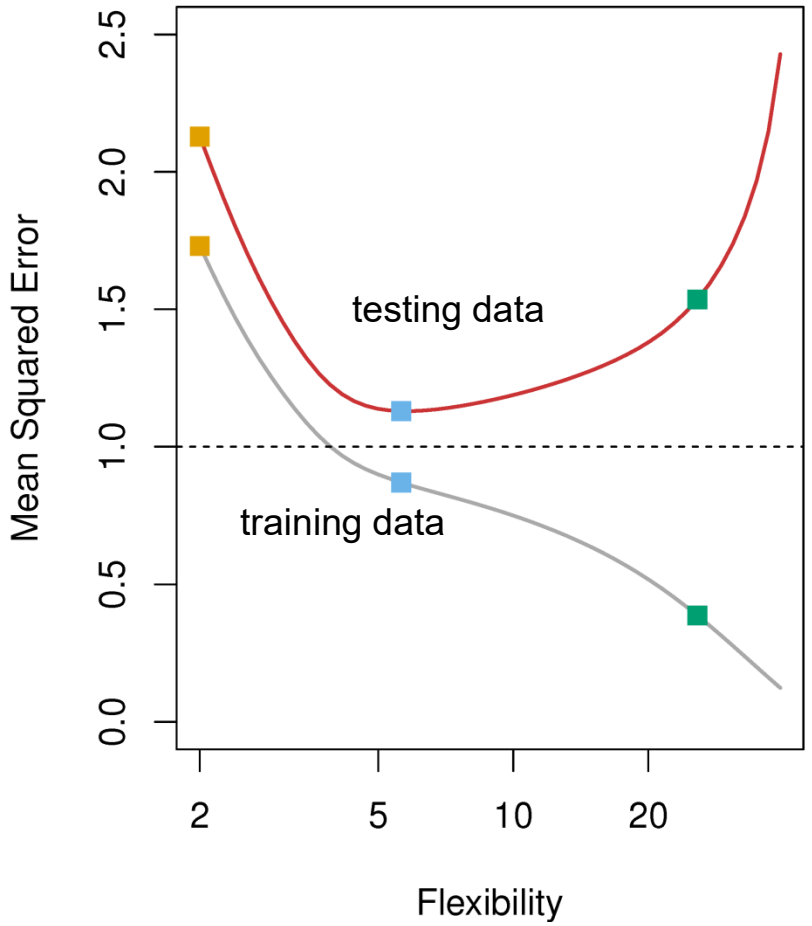
  Green line fits the best for the training data, but the blue line is the closest to the true function.

  OVER-FITTING



Assessing Model Accuracy





In reality, we don't know the red line nor the dash horizontal line

Assessing Model Accuracy

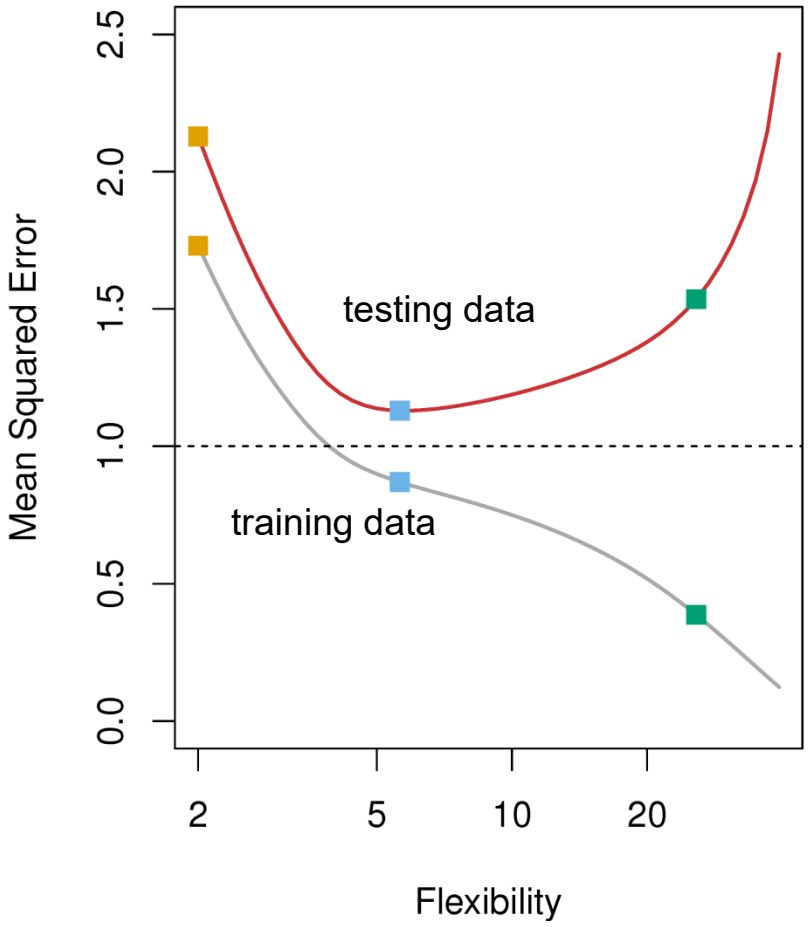
The mean squared difference (MSE)

$$E(Y-\hat{Y})^2 = E\left[f(X)+\varepsilon-\hat{f}(X)\right]^2$$

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

-- horizontal dashed line: $\text{Var}(\varepsilon)$

The lowest available test MSE among all possible methods



$$\begin{aligned}\text{Var}(X) &= \text{E}\left[(X-\text{E}[X])^2\right] \\ &= \text{E}\left[X^2 - 2X\,\text{E}[X] + \text{E}[X]^2\right] \\ &= \text{E}\left[X^2\right] - 2\,\text{E}[X]\,\text{E}[X] + \text{E}[X]^2 \\ &= \text{E}\left[X^2\right] - \text{E}[X]^2\end{aligned}$$

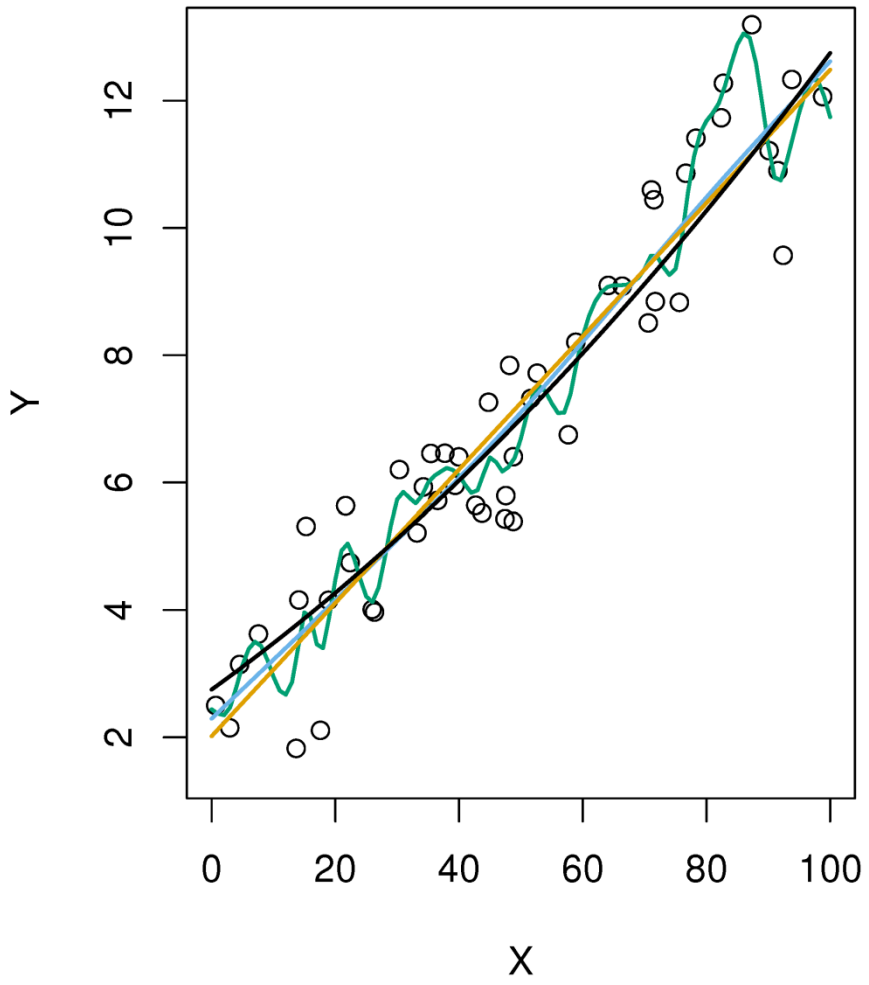
Assessing Model Accuracy

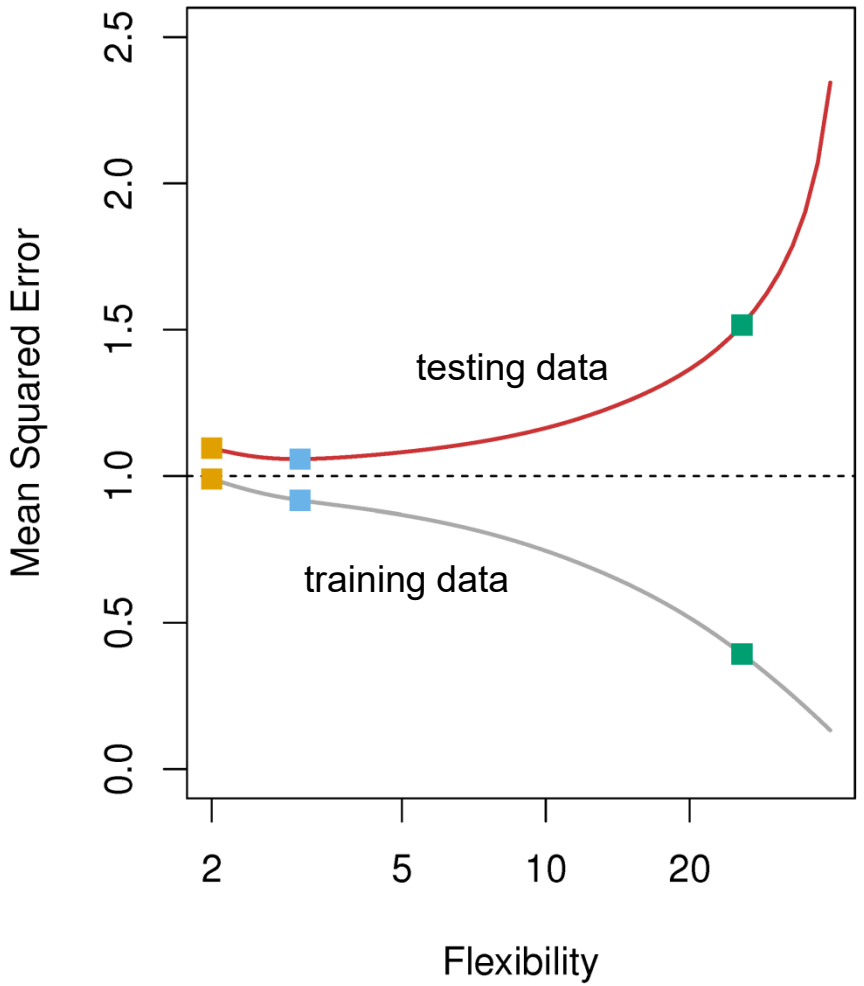
Observation:
"a monotone decrease in the training MSE and a U-shape in the testing MSE"

As model flexibility increases, training MSE will decrease, but the testing MSE may not.

This is a fundamental property of statistical learning that holds regardless of the particular data set at hand and regardless of the statistical method being used.

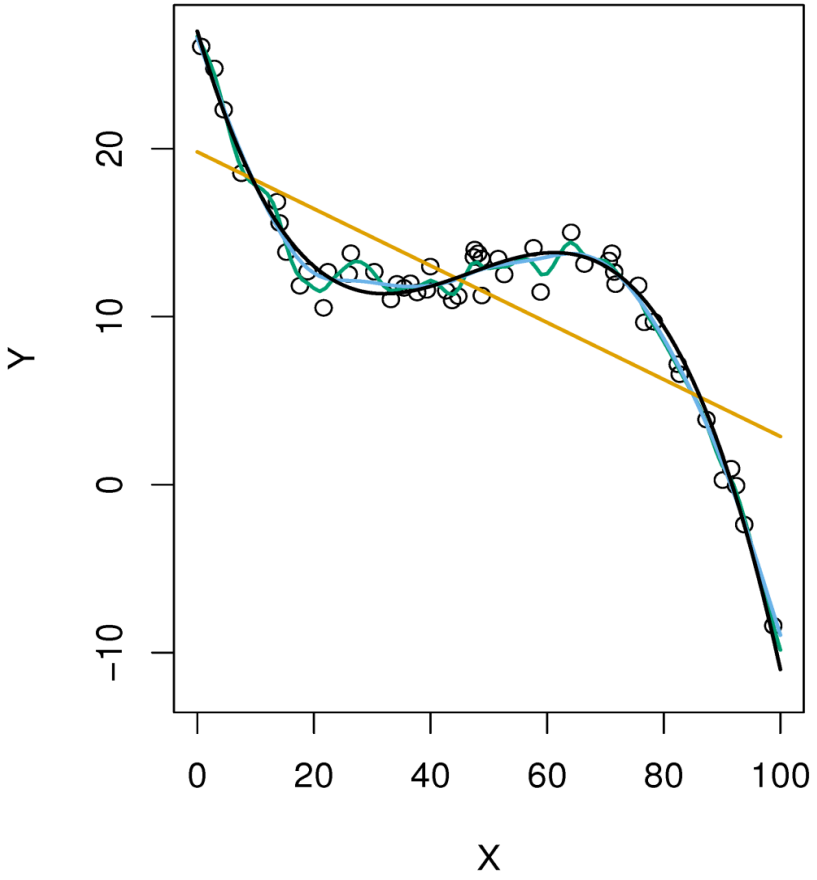
Assessing Model Accuracy

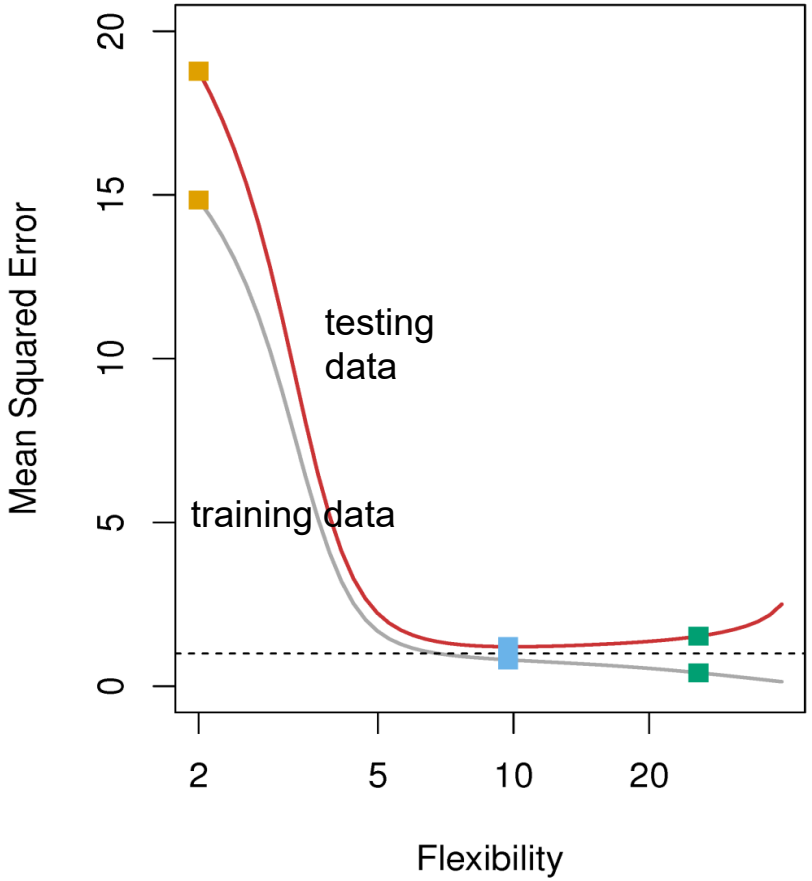




--open circle : observations
--orange line: linear regression
--black line: true function
--green line: non-linear fitting
--blue line: non-linear fitting

Assessing Model Accuracy



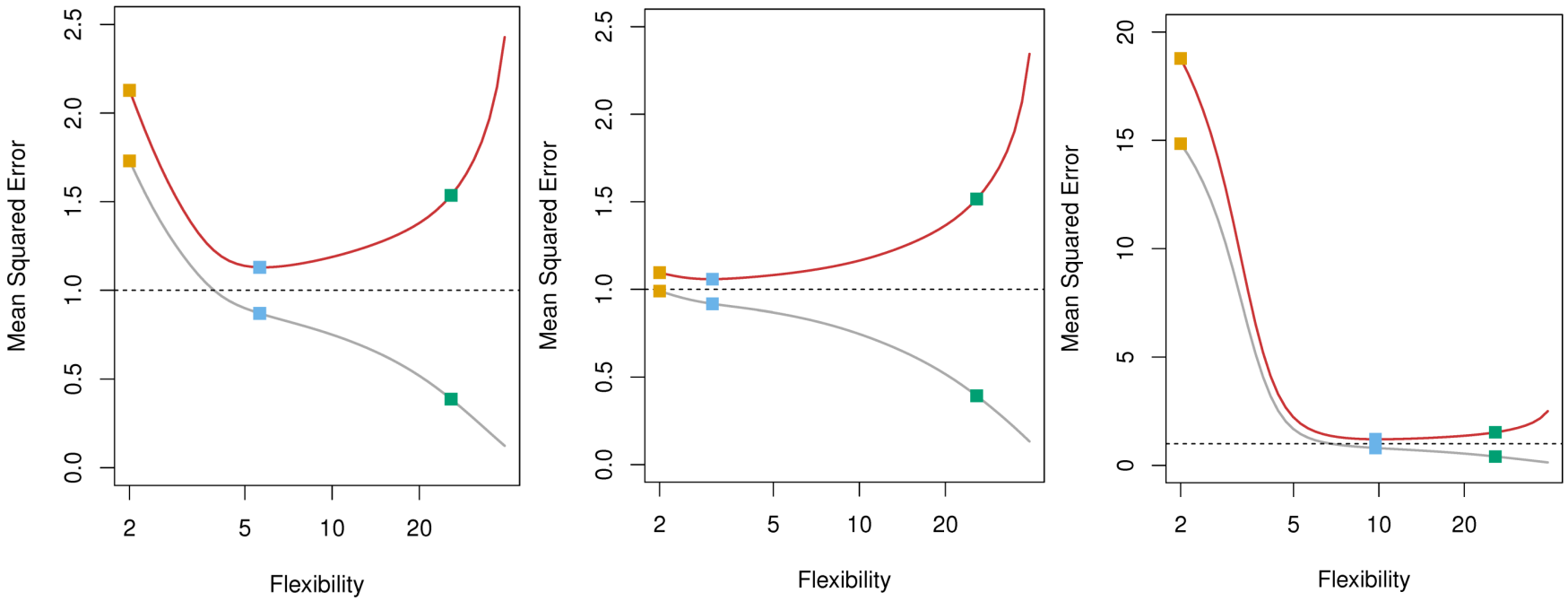


--open circle : observations
--orange line: linear regression
--black line: true function
--green line: non-linear fitting
--blue line: non-linear fitting

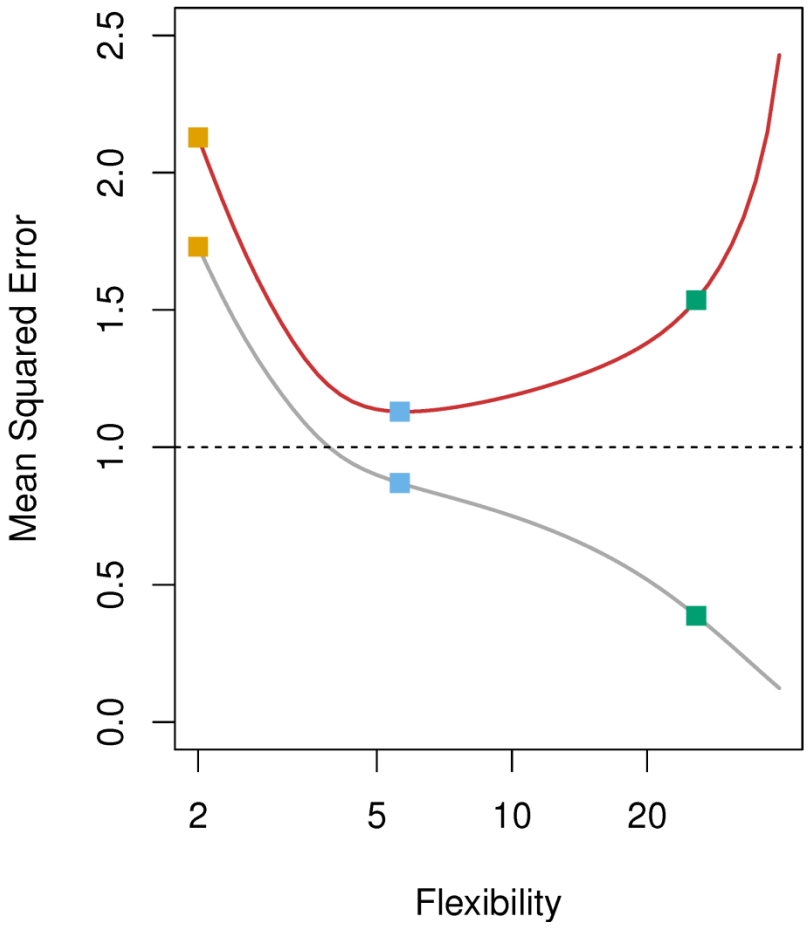
Assessing Model Accuracy

The flexibility level corresponding to the model with the minimal test MSE can vary considerably among data sets.

There are a variety of approaches to estimate this minimum point, such as *cross-validation*.



The Bias-Variance Trade-off



The U-shape in the test MSE curve is the result of two competing properties of statistical learning methods.

The Bias-Variance Trade-off

- Bias/variance decomposition of MSE

$$E\left(y_0-\hat{f}(x_0)\right)^2 = E\left(f(x_0)+\varepsilon-\hat{f}(x_0)\right)^2$$

$$= E\left(f(x_0)-E(\hat{f}(x_0))+E(\hat{f}(x_0))-\hat{f}(x_0)+\varepsilon\right)^2$$

$$= E\begin{pmatrix}\left(f(x_0)-E(\hat{f}(x_0))\right)^2+\left(E(\hat{f}(x_0))-\hat{f}(x_0)\right)^2+2\left(f(x_0)-E(\hat{f}(x_0))\right)\left(E(\hat{f}(x_0))-\hat{f}(x_0)\right)\\ +\varepsilon^2+2\varepsilon\left(f(x_0)-E(\hat{f}(x_0))\right)+2\varepsilon\left(E(\hat{f}(x_0))-\hat{f}(x_0)\right)\end{pmatrix}$$

$$= \underbrace{\left(f(x_0)-E(\hat{f}(x_0))\right)^2}_{[\mathrm{Bias}(\hat{f}(x_0))]^2} + \underbrace{E\left(\left(E(\hat{f}(x_0))-\hat{f}(x_0)\right)^2\right)}_{\mathrm{Var}(\hat{f}(x_0))} + \underbrace{E(\varepsilon^2)}_{\mathrm{Var}(\varepsilon)}$$

https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff

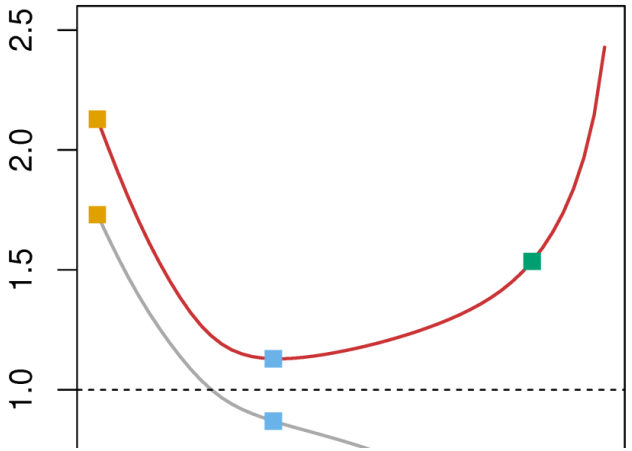
The Bias-Variance Trade-off

❑ Bias/variance decomposition of MSE

$$\text{expected test MSE} = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon)$$

- To minimize the expected test error, we need to select a statistical learning method that simultaneously achieves *low variance* and *low bias*.

- We always have $\text{expected test MSE} \geq \underbrace{\text{Var}(\varepsilon)}_{\text{Irreducible error}}$

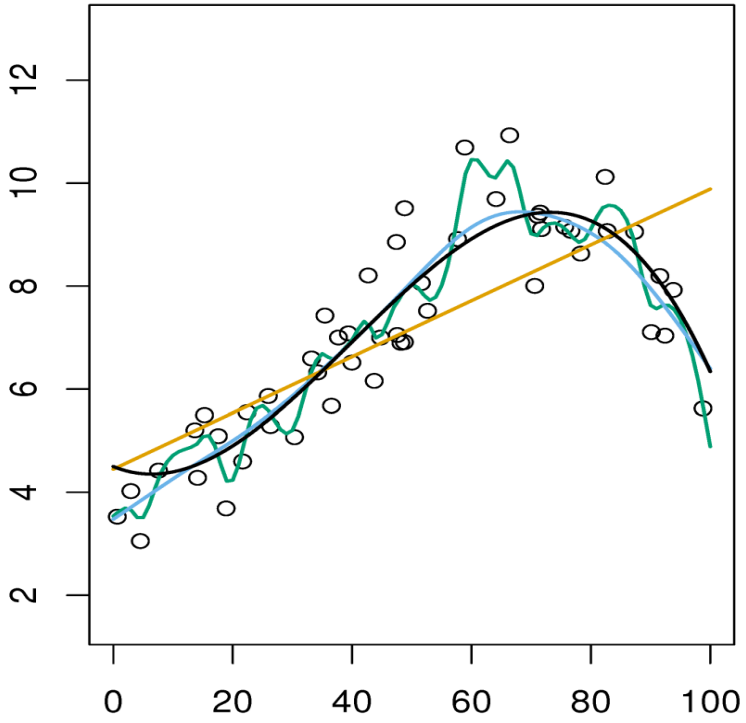


The Bias-Variance Trade-off

- The *variance* of a statistical learning method

The amount by which $\hat{f}$ would change if we estimated it using a different training data set.

In general, more flexible statistical methods have higher variance



- The **flexible** green curve is following the observations very closely. It has **high variance** because changing any one of these data points may cause the estimated line to change considerably. The orange linear line is **inflexible** and has **low variance** in that moving any single observation will likely cause only a small shift in the position of the line.

The Bias-Variance Trade-off

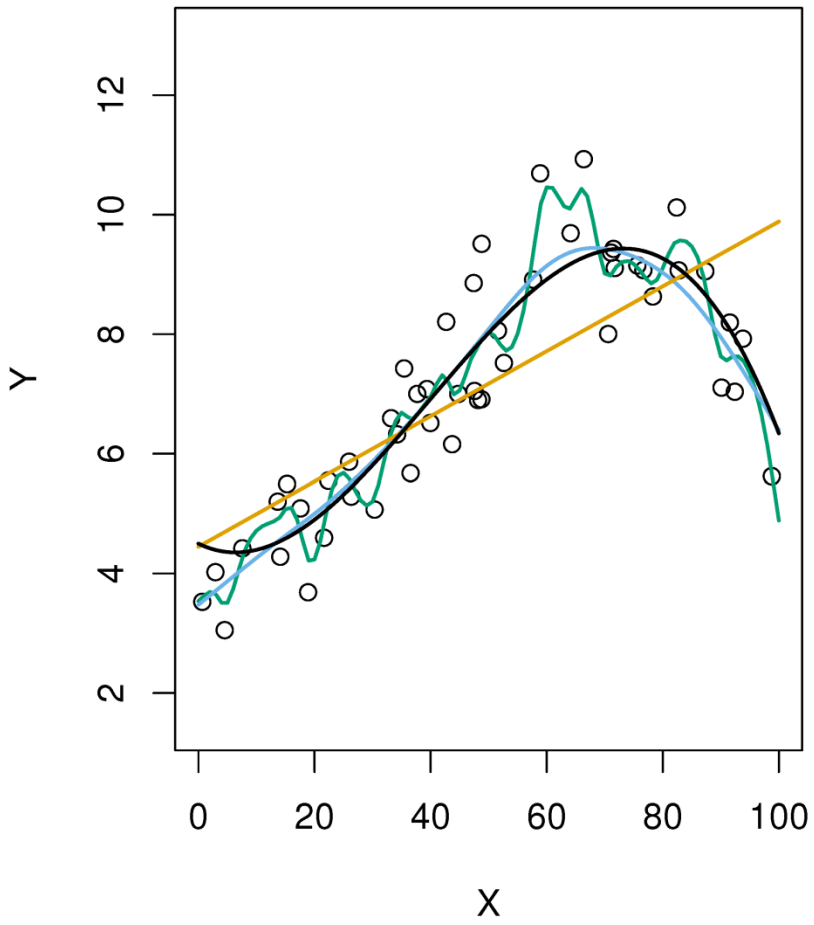
- The *bias* of a statistical learning method

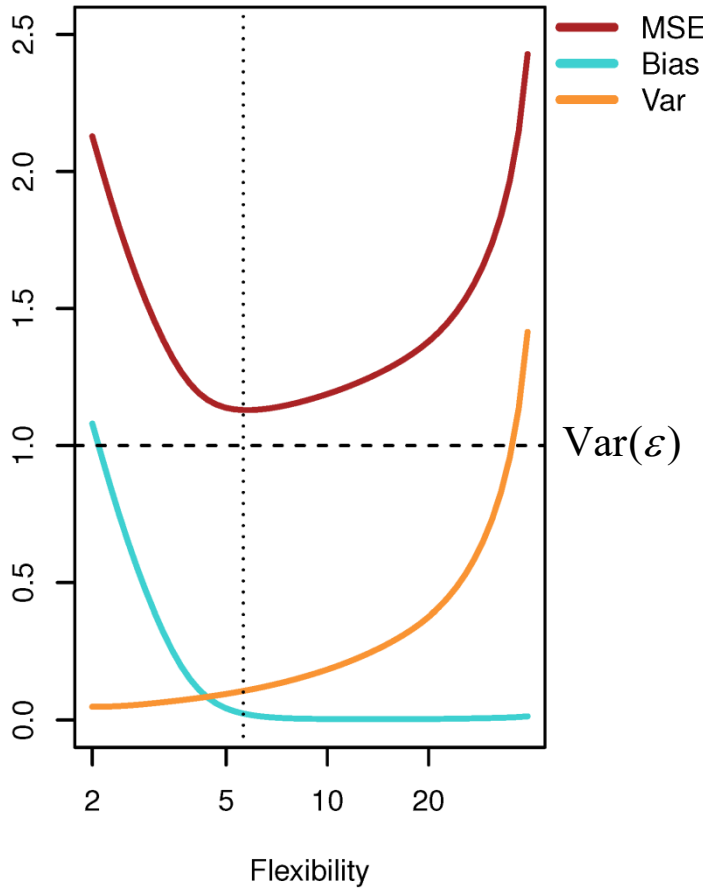
  The error that is introduced by approximating a real-life problem

  In general, more flexible statistical methods have lower bias

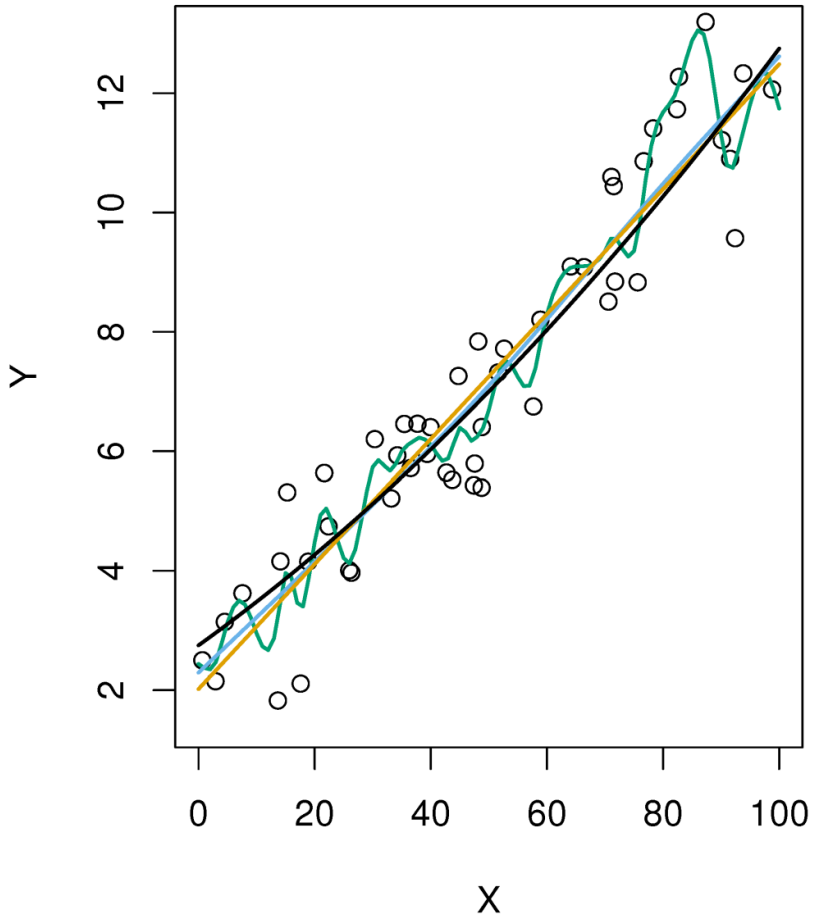
  Typically as the flexibility increases, its variances increases, and its bias decreases. So choosing the flexibility based on average test error amounts to an bias-variance trade-off.

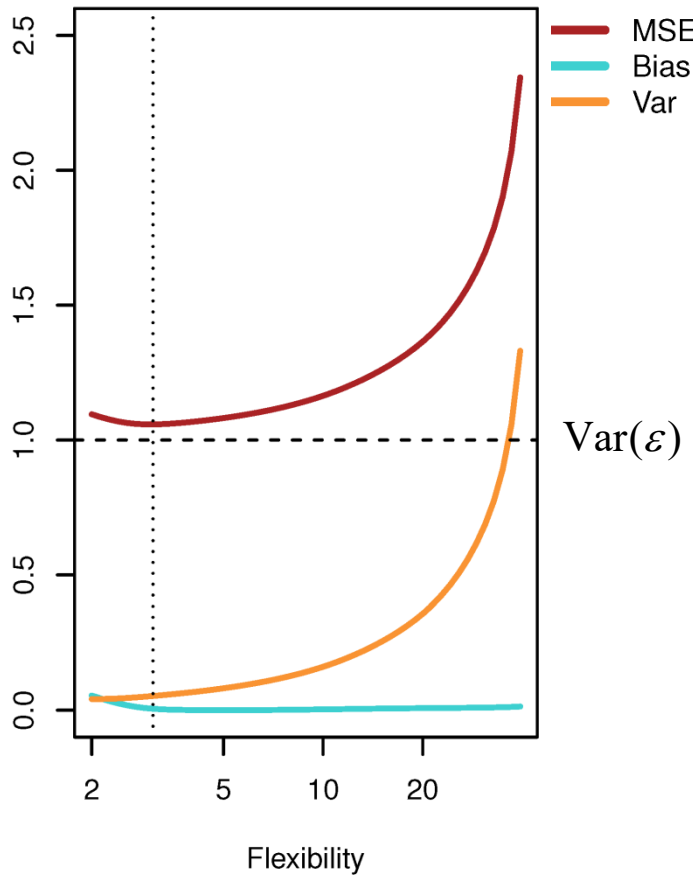
The Bias-Variance Trade-off





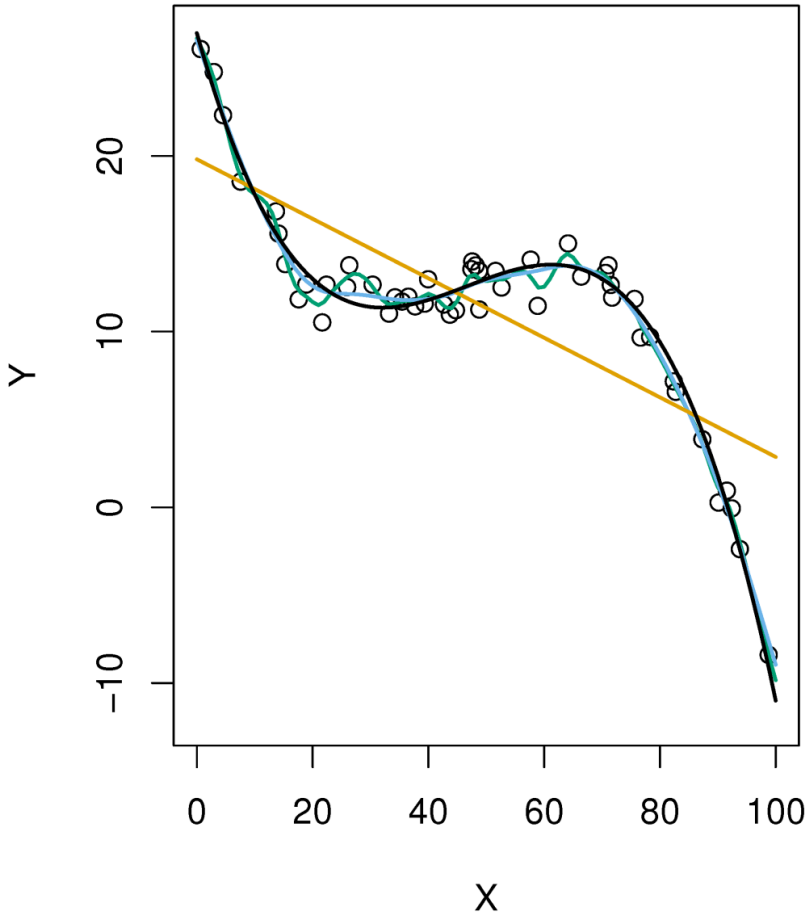
The Bias-Variance Trade-off

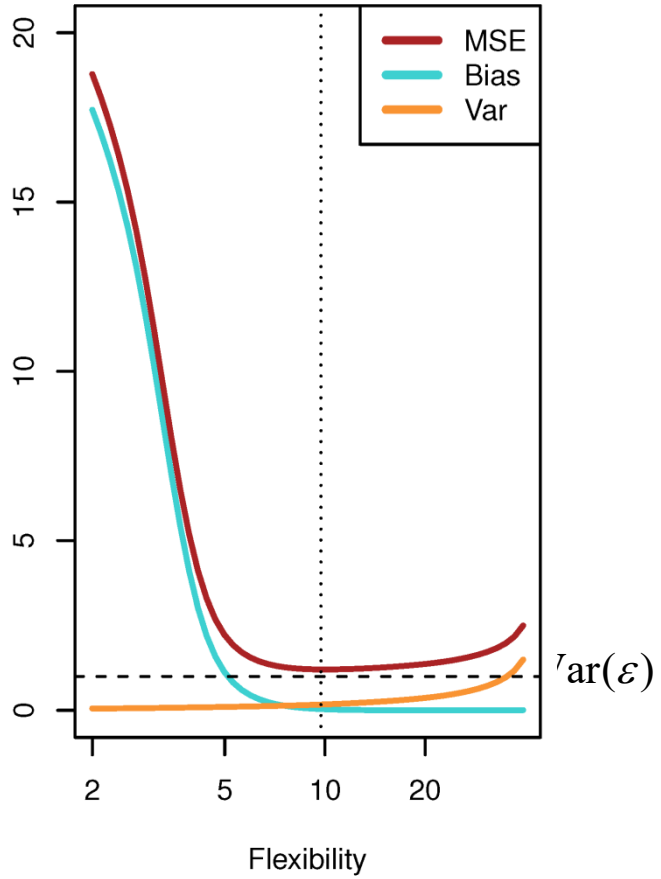




$$\text{expected test MSE} = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\varepsilon)$$

The Bias-Variance Trade-off





The Bias-Variance Trade-off

We will explore methods that are extremely flexible and hence can essentially eliminate bias. However, this does not guarantee that they will outperform a much simpler method such as linear regression.

Supplementary: Overfitting in statistical/machine learning

- The high variance of the model performance is an indicator of an overfitting problem.

Overfitting happens when:

The data used for training is not cleaned and contains garbage values. The model captures the noise in the training data and fails to generalize the model's learning.

The model has a high variance.

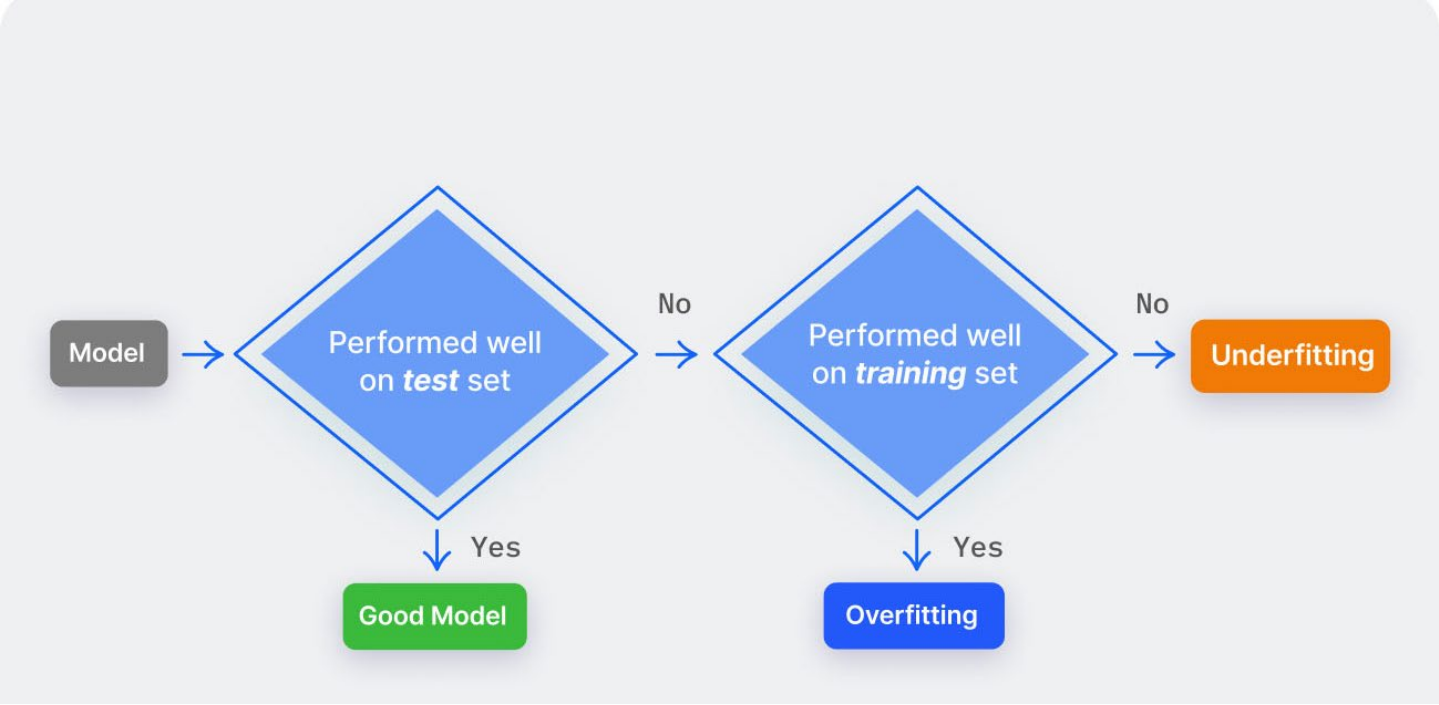
The training data size is not enough, and the model trains on the limited training data for several epochs.

The architecture of the model has several neural layers stacked together. Deep neural networks are complex and require a significant amount of time to train, and often lead to overfitting the training set.

Supplementary: Overfitting in statistical/machine learning

- Underfitting happens when:

1. Unclean training data containing noise or outliers can be a reason for the model not being able to derive patterns from the dataset.
2. The model has a high bias due to the inability to capture the relationship between the input examples and the target values.
3. The model is assumed to be too simple. For example, training a linear model in complex scenarios.



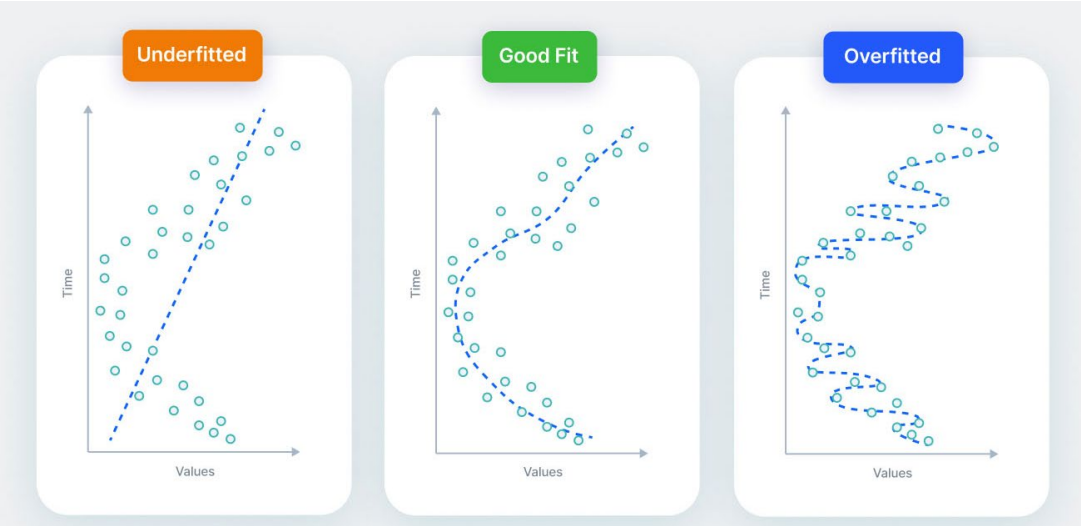
Supplementary: Overfitting in statistical/machine learning

How to detect overfit models?

K-fold cross-validation is one of the most popular techniques commonly used to detect overfitting.

We split the data points into k equally sized subsets in K-folds cross-validation, called "folds." One split subsets act as the testing set, and the remaining folds will train the model.

The model is trained on a limited sample to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. One fold acts as a validation set in each turn.



Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

1. Train with more data

With the increase in the training data, the crucial features to be extracted become prominent. The model can recognize the relationship between the input attributes and the output variable. The only assumption in this method is that the data to be fed into the model should be clean; otherwise, it would worsen the problem of overfitting.

2. Data augmentation

An alternative method to training with more data is data augmentation, which is less expensive and safer than the previous method. Data augmentation makes a sample data look slightly different every time the model processes it.

Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

3. Addition of noise to the input data

Another similar option as data augmentation is adding noise to the input and output data. Adding noise to the input makes the model stable without affecting data quality and privacy while adding noise to the output makes the data more diverse. Noise addition should be done in limit so that it does not make the data incorrect or too different.

4. Feature selection

Every model has several parameters or features depending upon the number of layers, number of neurons, etc. The model can detect many redundant features or features determinable from other features leading to unnecessary complexity. We very well know that the more complex the model, the higher the chances of the model to overfit.

Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

5. Cross-validation

Cross-validation is a robust measure to prevent overfitting. The complete dataset is split into parts. In standard K-fold cross-validation, we need to partition the data into k folds. Then, we iteratively train the algorithm on k-1 folds while using the remaining holdout fold as the test set. This method allows us to tune the hyperparameters of the neural network or machine learning model and test it using completely unseen data.

6. Simplify data

Till now, we have come across model complexity to be one of the top reasons for overfitting. The data simplification method is used to reduce overfitting by decreasing the complexity of the model to make it simple enough that it does not overfit. Some of the procedures include pruning a decision tree, reducing the number of parameters in a neural network, and using dropout on a neutral network.

Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

7. Regularization

If overfitting occurs when a model is too complex, reducing the number of features makes sense. Regularization methods like Lasso, L1 can be beneficial if we do not know which features to remove from our model. Regularization applies a "penalty" to the input parameters with the larger coefficients, which subsequently limits the model's variance.

8. Ensembling

It is a machine learning technique that combines several base models to produce one optimal predictive model. In Ensemble learning, the predictions are aggregated to identify the most popular result. Well-known ensemble methods include bagging and boosting, which prevents overfitting as an ensemble model is made from the aggregation of multiple models.

Supplementary: Overfitting in statistical/machine learning

10 techniques to avoid overfitting

9. Early stopping

This method aims to pause the model's training before memorizing noise and random fluctuations from the data. There can be a risk that the model stops training too soon, leading to underfitting. One has to come to an optimum time/iterations the model should train.

10. Adding dropout layers

Large weights in a neural network signify a more complex network. Probabilistically dropping out nodes in the network is a simple and effective method to prevent overfitting. In regularization, some number of layer outputs are randomly ignored or “dropped out” to reduce the complexity of the model.

Our tip: If one has two models with almost equal performance, the only difference being that one model is more complex than the other, one should always go with the less complex model. In data science, it's a thumb rule that one should always start with a less complex model and add complexity over time.