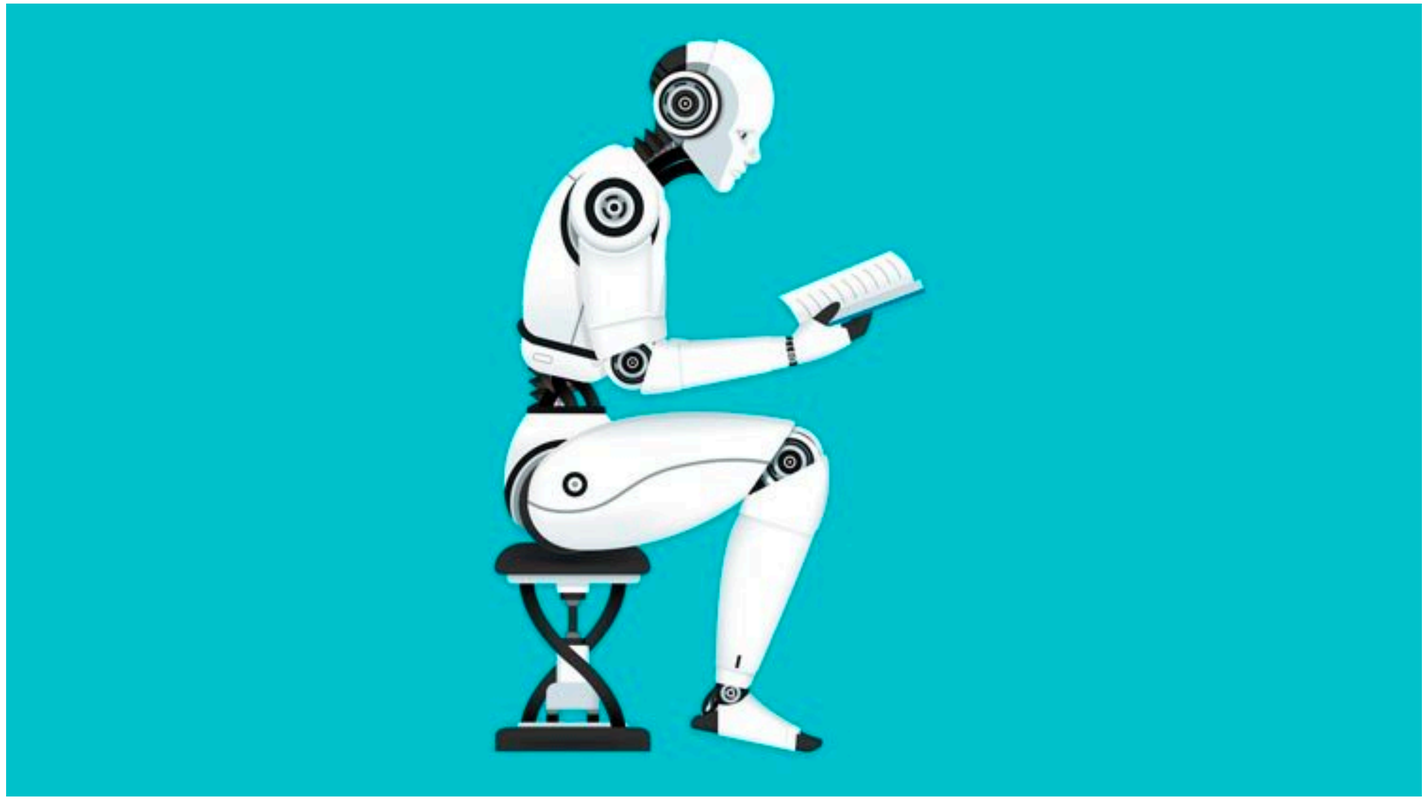
UNIVERSITAT DE BARCELONA



What is Learning?

Machine Learning | Enginyeria Informàtica

Santi Seguí | 2019-2020

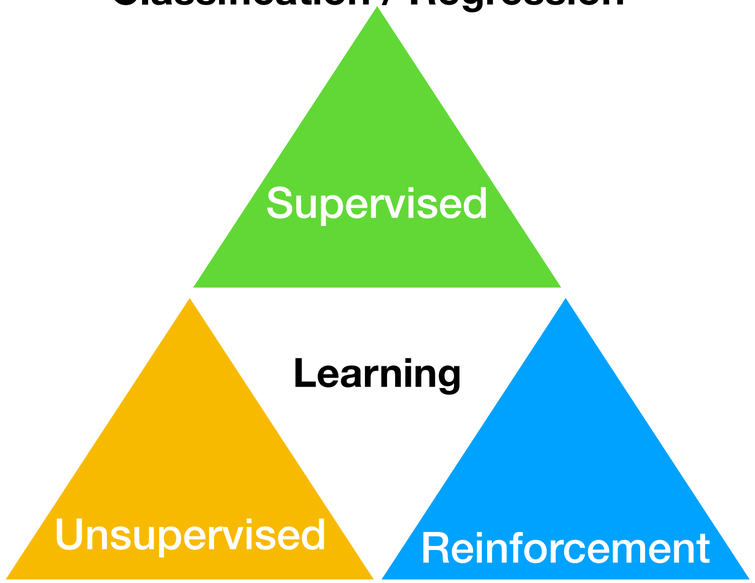
- **Data** are predicted
  - on the basis of a set of **features** (e.g. clinical measurements);
  - from a set of (observed) **training data** on these features;
  - for a set of **objects/instances** (e.g. people).

- **Inputs** for the problems are also called predictor or independent variables
- **Outputs** are also called responses or dependent variables

- The predictor model is also called estimator

Type of Machine Learning

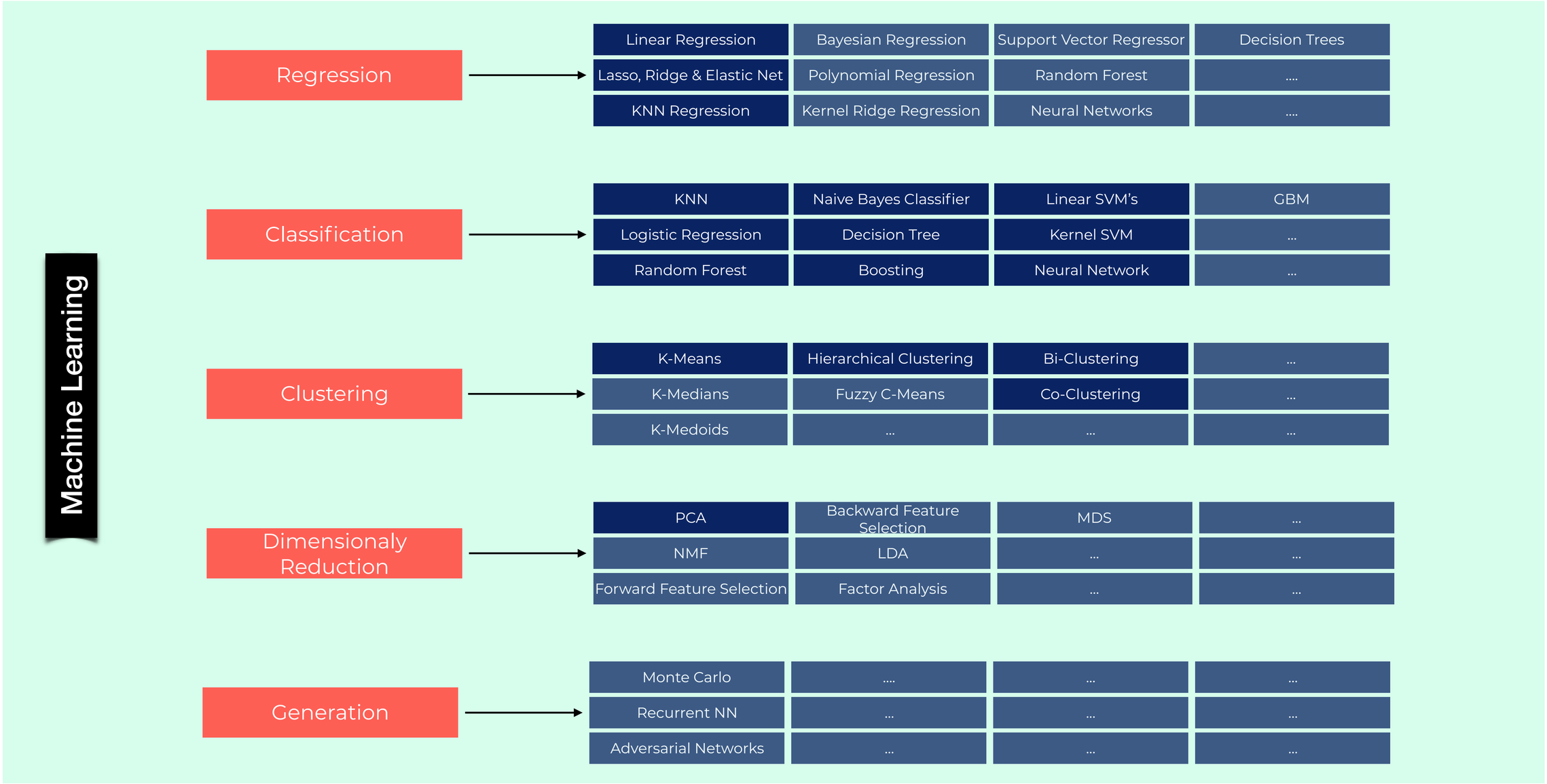
Labeled Data
Direct Feedback
Predict outcome/future
Classification / Regression



No labels
No feedback
“find hidden structure”
Clustering

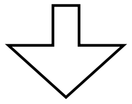
Decision process
Reward system
Learn series of actions
Learn from mistakes

Classification	**To which data category does this data point belongs?**	**Medical diagnosis**: Does this tissue show signs of diseases? **Banking**: Is this transaction fraudulent? **Computer Vision**: What type of object is in this picture? Is it a person? Is it a building?
Regression	**Given this input from a dataset, what is the likely value of a particular quantity?**	**Finance:** What is the value of this stock going to be tomorrow? **Housing:** What would be the price of this house if it was sold today? **Food Quality**: When should I pick this strawberry?
Clustering	**Which data point are similar to each other?**	**E-Commerce:** which customers are exhibiting similar behavior to each other, how do they group together? **Video Streaming**: what are the different types of video genres in our catalogue, and which ones are in the same genre?
Dimensionality reduction	**What are the most significant features of this data and how can be summarized?**	**E-Commerce:** What combinations of features allows us to summarize the behavior of our customers? **Molecular biology:** How can scientists summarize the behavior of all 20.000 human gens in a particular diseases?
Semi-supervised learning	**How can be labelled and unlabelled data be combined?**	**Computer Vision:** How can an object detection be developed, with only a small training data set? **Drug Discovery:** Which of the million possible drugs could be effective agains disease, we have so far only tested a few?
Reinforcement learning	**What actions will most effectively achieve a desired endpoint?**	**Robots:** How can a robot move through its environment? **Games:** Which moves were more important in helping the computer win a particular game?

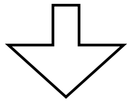

Machine Learning
Regression
Linear Regression
Bayesian Regression
Support Vector Regressor
Decision Trees
Lasso, Ridge & Elastic Net
Polynomial Regression
Random Forest
....
KNN Regression
Kernel Ridge Regression
Neural Networks
....
Classification
KNN
Naive Bayes Classifier
Linear SVM's
GBM
Logistic Regression
Decision Tree
Kernel SVM
...
Random Forest
Boosting
Neural Network
...
Clustering
K-Means
Hierarchical Clustering
Bi-Clustering
...
K-Medians
Fuzzy C-Means
Co-Clustering
...
K-Medoids
...
...
...
Dimensionaly Reduction
PCA
Backward Feature Selection
MDS
...
NMF
LDA
...
...
Forward Feature Selection
Factor Analysis
...
...
Generation
Monte Carlo
....
...
...
Recurrent NN
...
...
...
Adversarial Networks
...
...
...

No Free Lunch Theorem

Our model is a simplification of reality

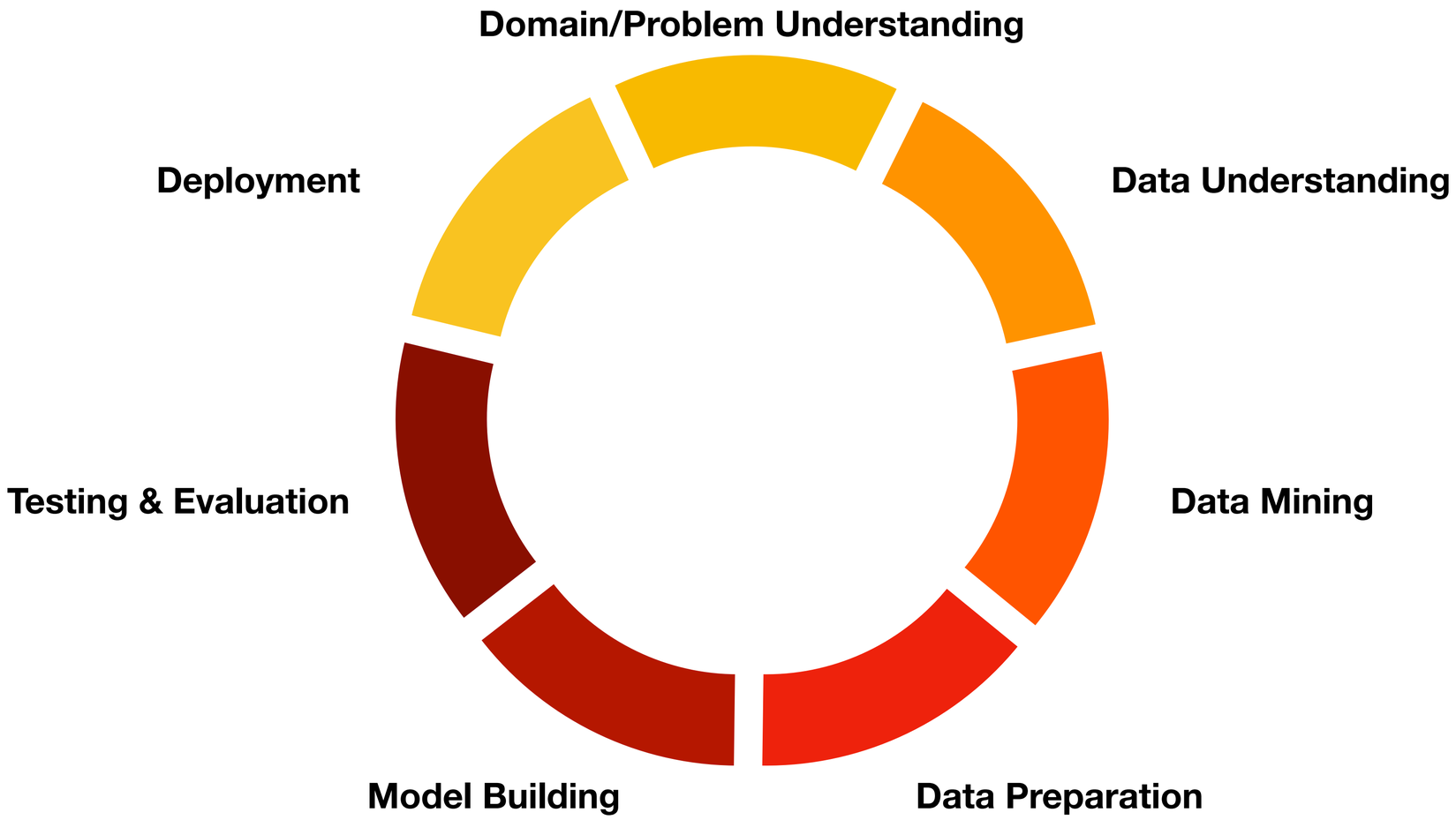


Simplification is based on assumptions (bias)



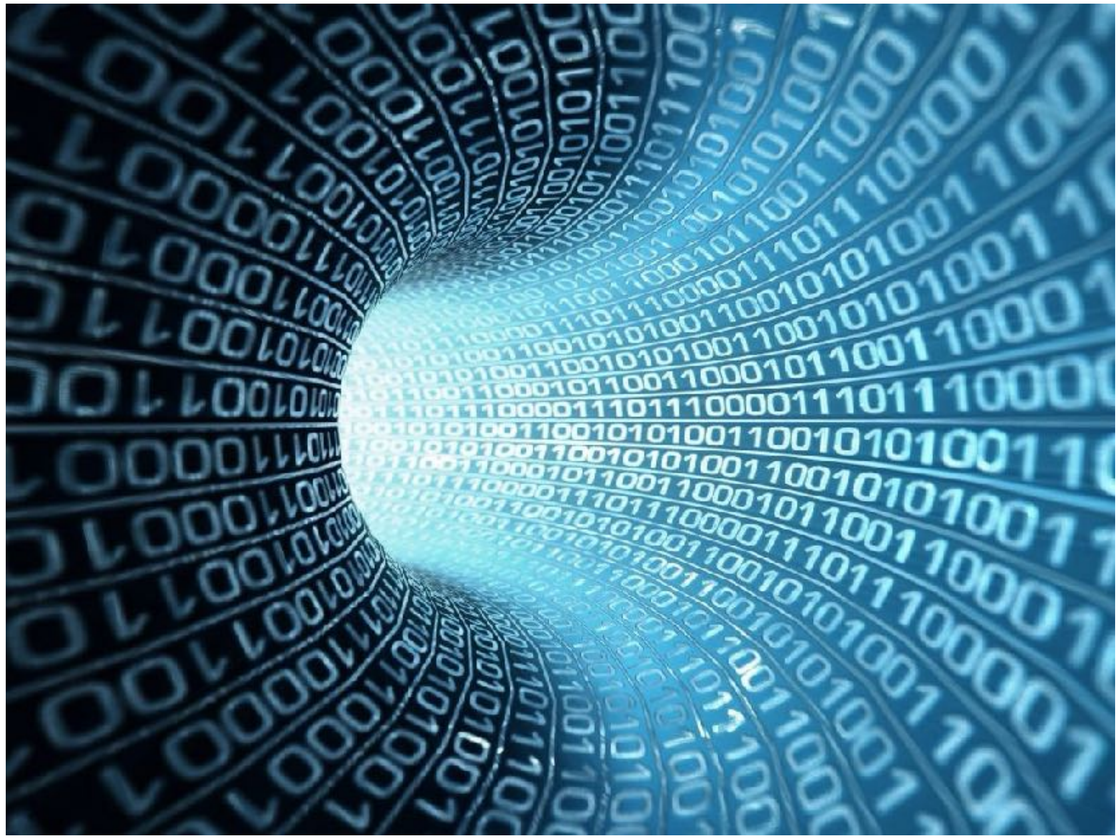
Assumptions fail in certain situations

Roughly speaking: "**There is not a model that works best for all possible situations**."


Domain/Problem Understanding
Data Understanding
Data Mining
Data Preparation
Model Building
Testing & Evaluation
Deployment

Data?

- What is data? Which types of data exist?

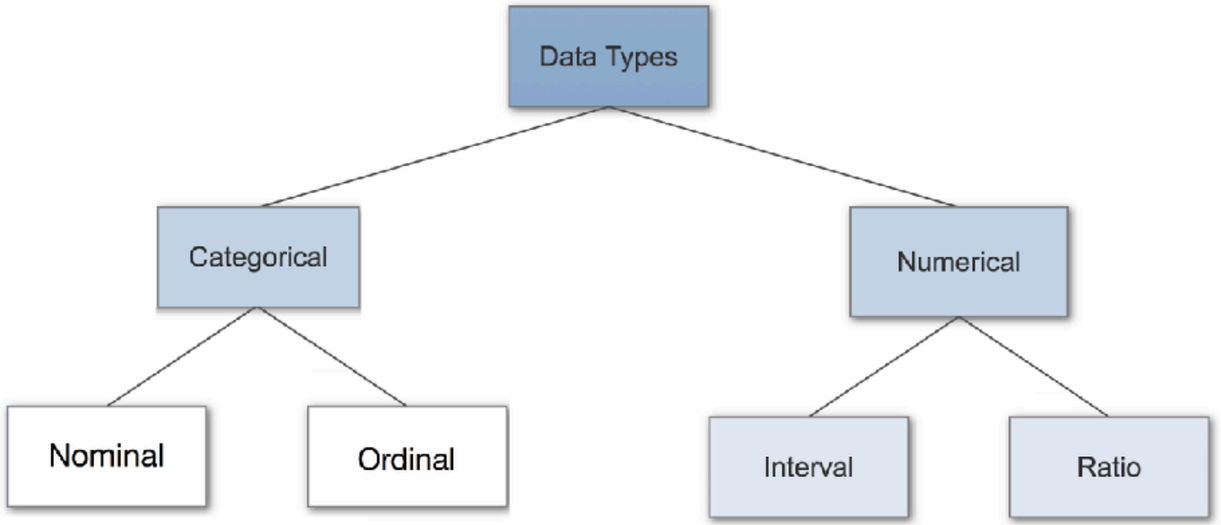


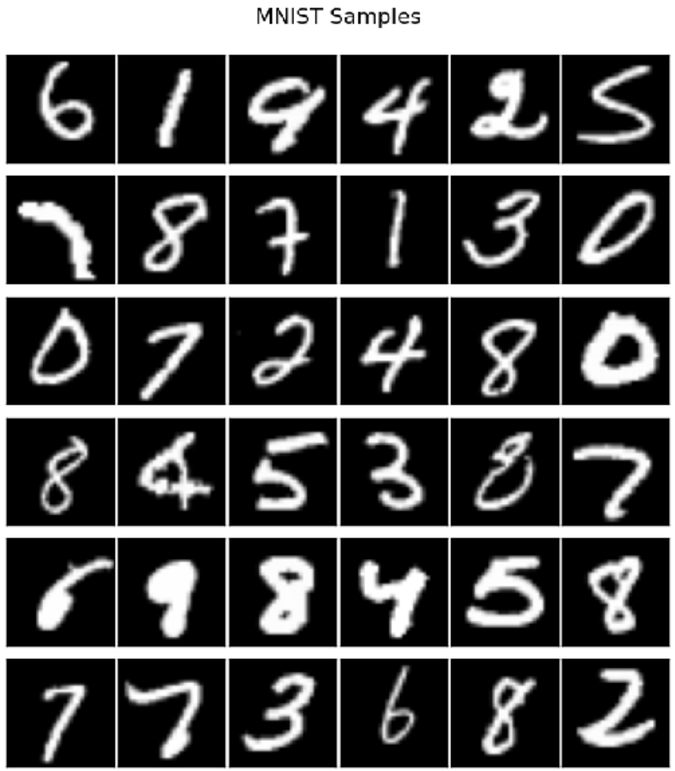
Data

- Numerical Values (Age, Salary, Blood pressure,…)
- Categories (blood group, city,..)
- Images
- Videos
- Text
- Genes
- ….

Data

- What is data? Which types of data exist?
  - **Quantitative** (numerical): e.g. price, age,..
  - **Categorical** (discrete, often binary): Cancer/no cancer





$x \in \chi \subset \{0, 1, ..., 255\}^{400}$ is a vector of pixel intensities in a $20x20$ images.

Statistic Concepts
(a review)

Review: Some statistic concepts

- **Mean** (or arithmetic mean):

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1 + x_1 + \ldots + x_n}{n}$$

  sometimes is called μ

- **Median**:

$$\text{If } n \text{ is odd then } M_e = x^*_{((n+1)/2)}$$
$$\text{If } n \text{ is even then } M_e = \frac{x^*_{n/2} + x^*_{(n/2+1)}}{2}$$
$$\text{where } x^* \text{ is the sorted version of } x.$$

Review: Some statistic concepts

- **Variance** (Var(X)): measures how far a set of numbers are spread out

$$Var(X) = \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$std(X) = \sigma = \sqrt{Var(X)}$$

Review: Some statistic concepts

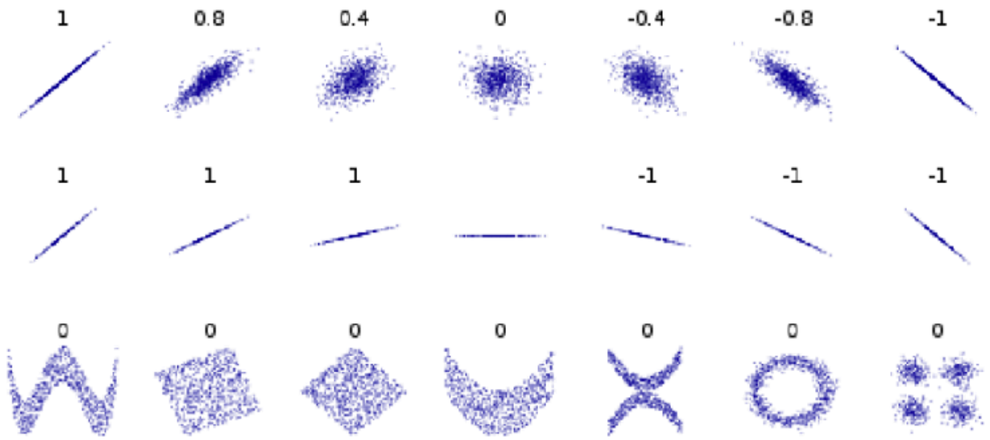
- **Covariance** (Cov(X,Y)) : measures how much two random variables change together:

$$Cov(X, Y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

Review: Some statistic concepts

- **Correlation** (p): is a measure of the linear correlation between two variables X and Y, giving a value between +1 and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation.

$$\rho(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y}$$

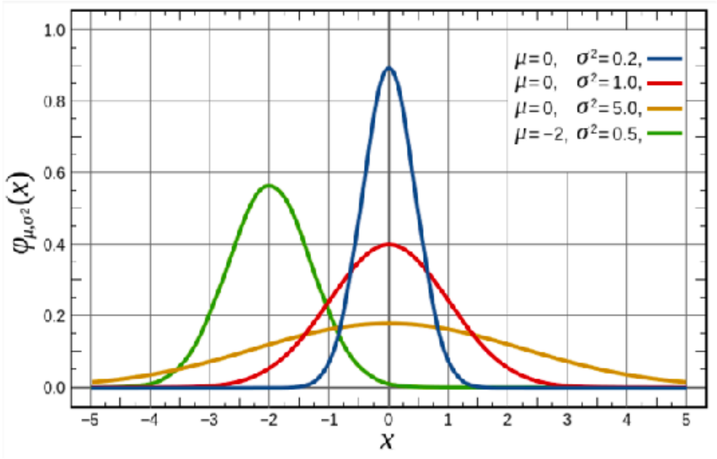


Review: Some statistic concepts

- **Normal** (or **gaussian**) **distribution** ($\mathcal{N}(\mu, \sigma^2)$)

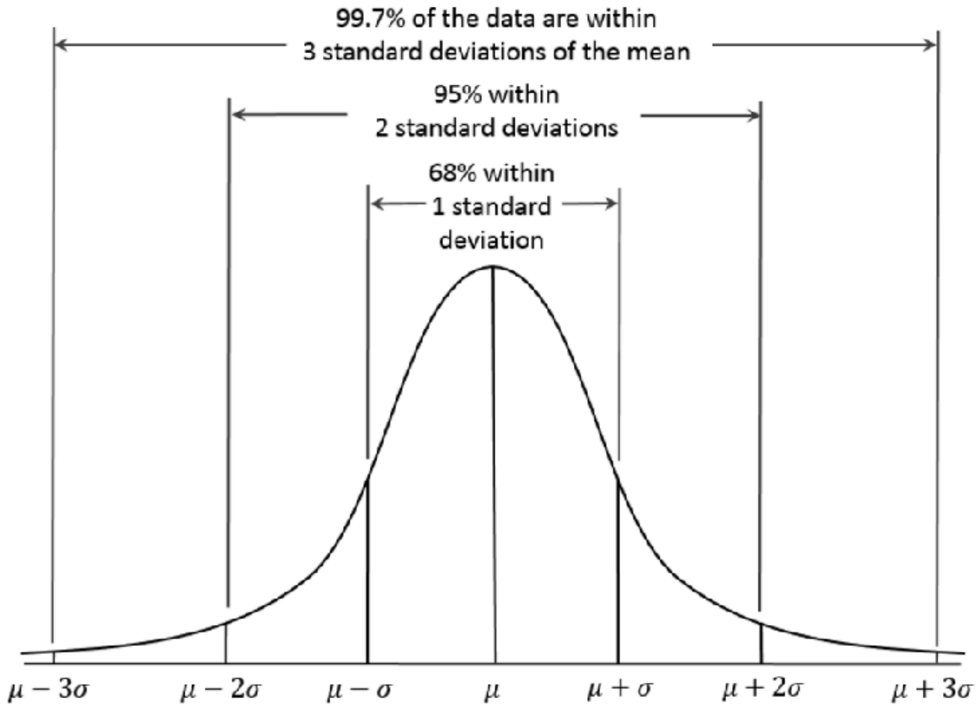
- The probability function of a normal distribution is:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$



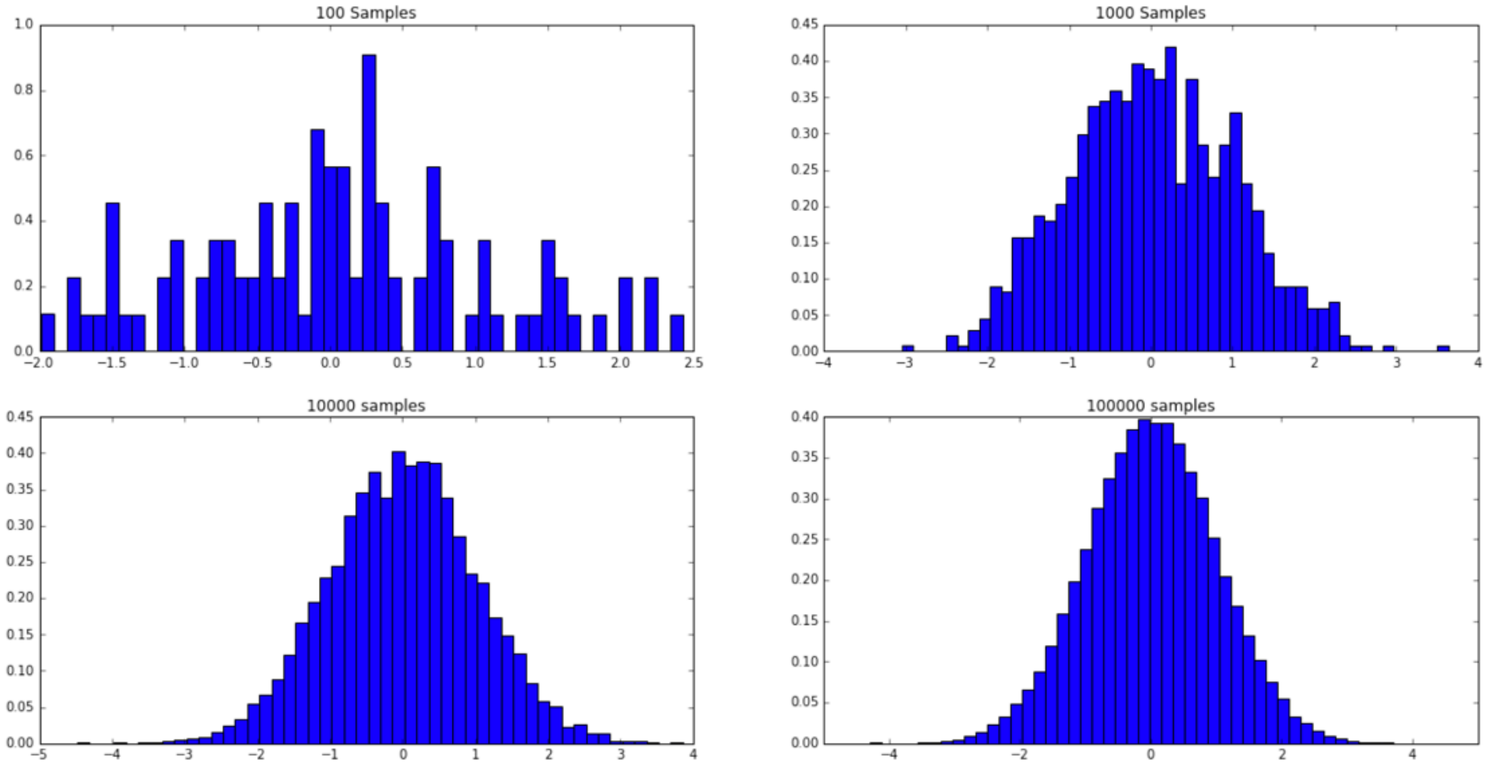
Review: Some statistic concepts

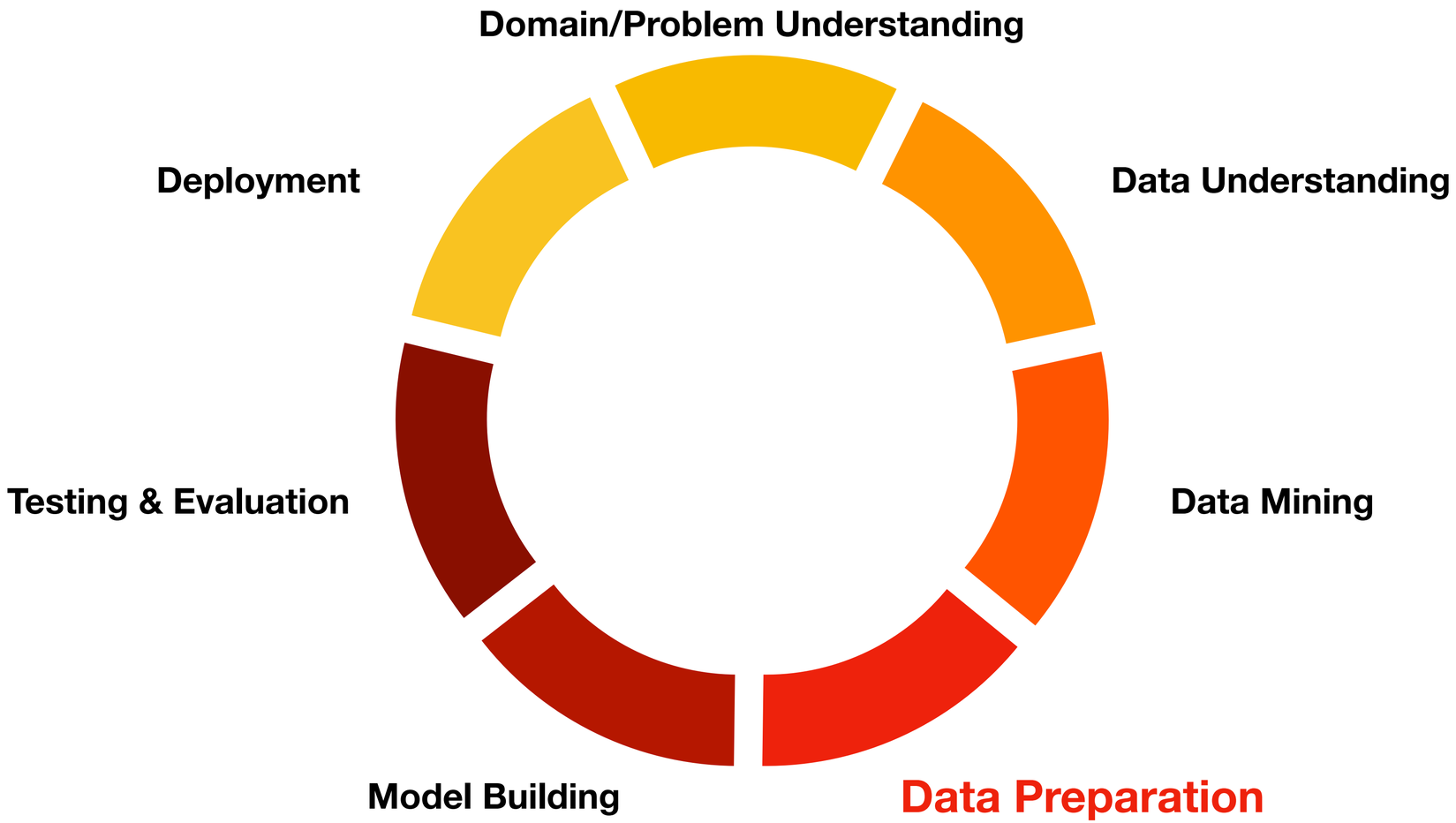
- Normal (or gaussian) distribution ($\mathcal{N}(\mu, \sigma^2)$)

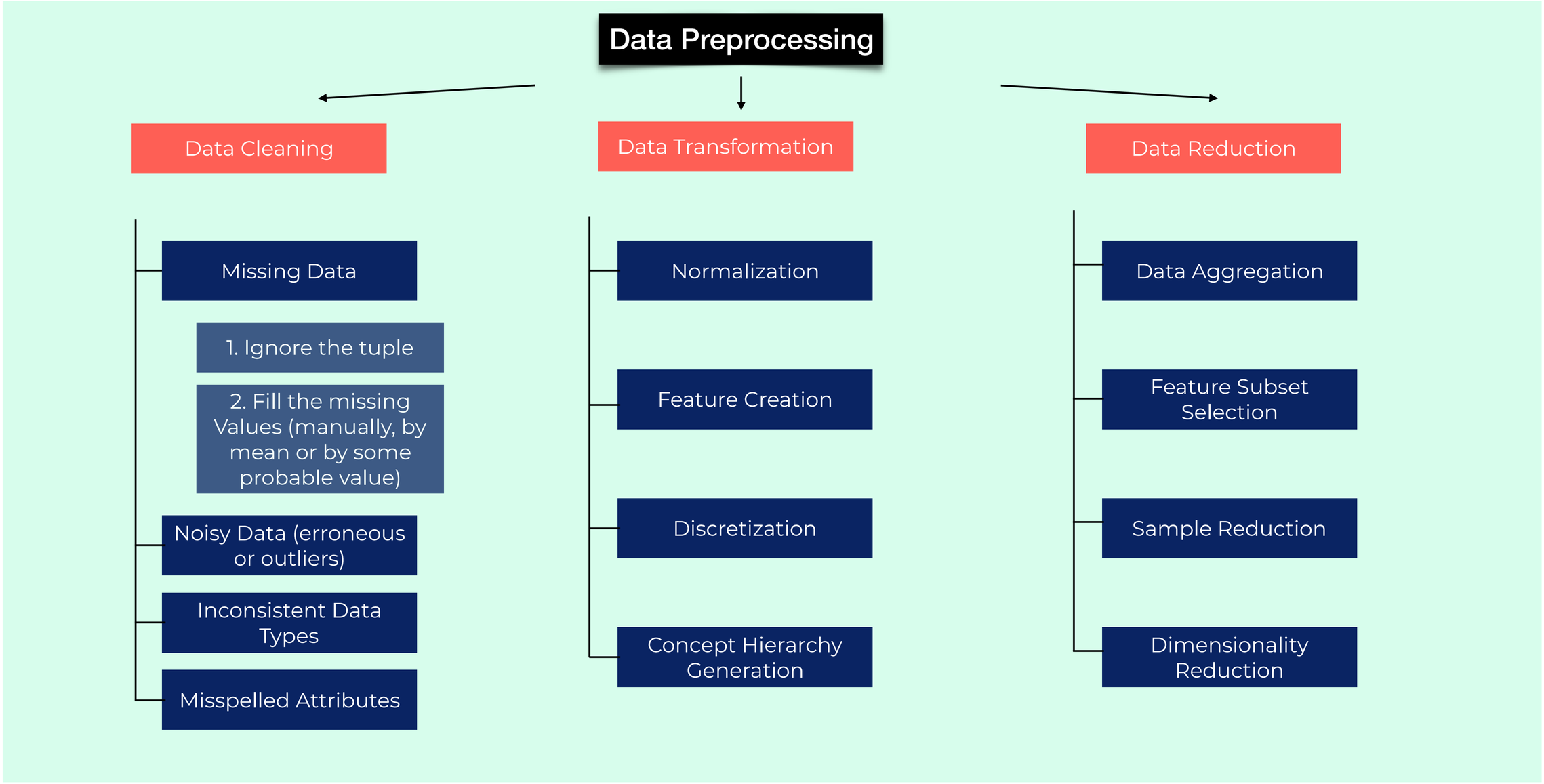


Review: Some statistic concepts

- Normal (or gaussian) distribution ($\mathcal{N}(\mu, \sigma^2)$)




Domain/Problem Understanding
Data Understanding
Data Mining
Data Preparation
Model Building
Testing & Evaluation
Deployment

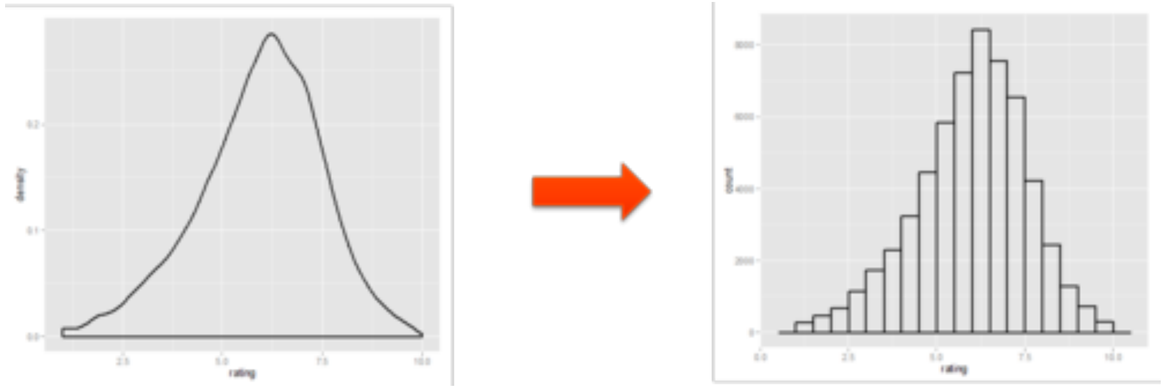

Data Preprocessing
Data Cleaning
Data Transformation
Data Reduction
Missing Data
1. Ignore the tuple
2. Fill the missing Values (manually, by mean or by some probable value)
Noisy Data (erroneous or outliers)
Inconsistent Data Types
Misspelled Attributes
Normalization
Feature Creation
Discretization
Concept Hierarchy Generation
Data Aggregation
Feature Subset Selection
Sample Reduction
Dimensionality Reduction

Data Transformation/Normalization

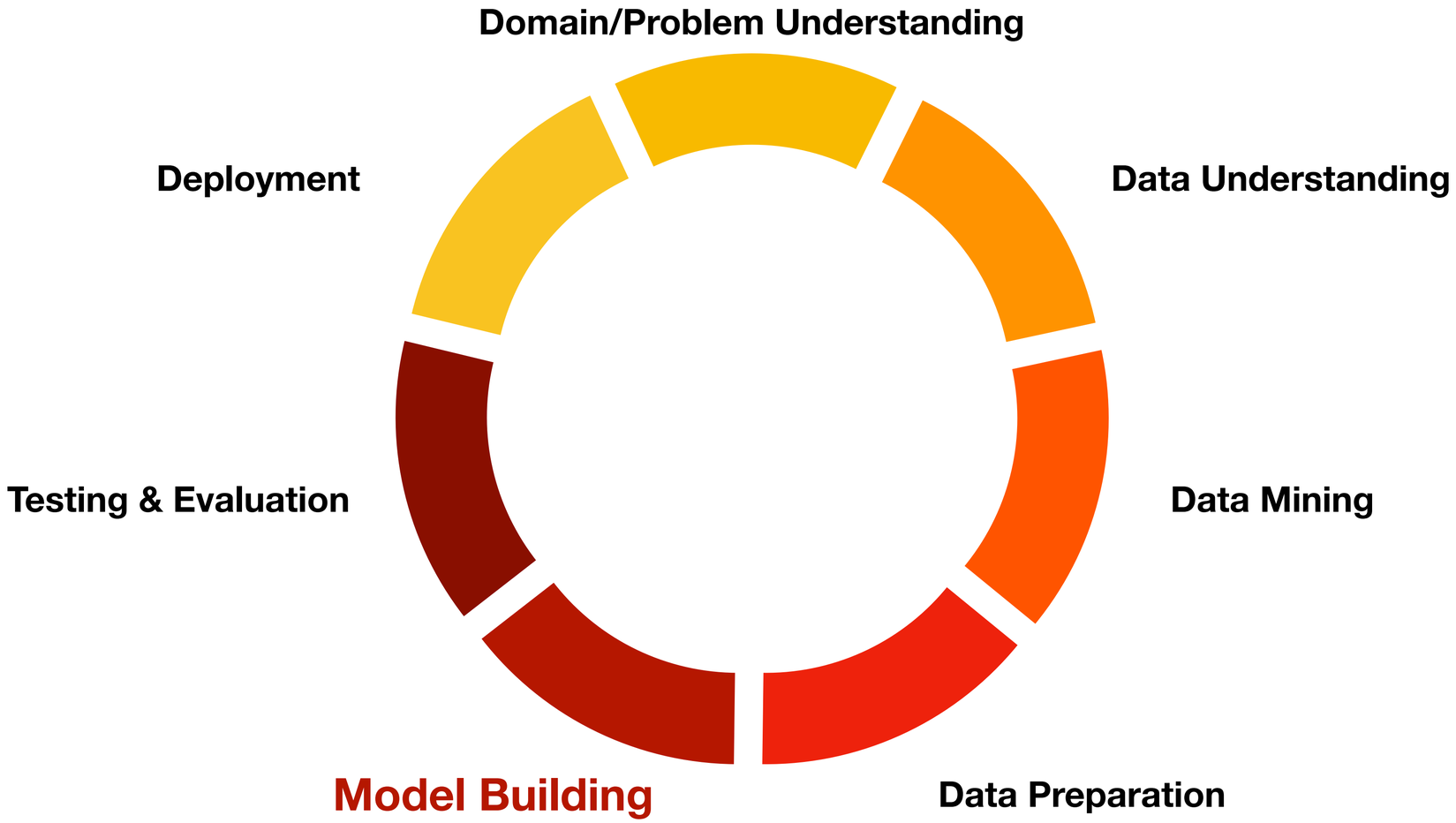
- In order to handle noise in the data we can transform it by normalization or discretization.
- Normalization/Discretization can be different from data type to data type
  - Normalization:
    - *Min-Max* normalization (for example into [0,1] range):
      - $x' = \frac{x - min}{max - m}$
    - Z-score normalization (also know as standardization)
      - $x' = \frac{x - mean}{std}$

Data Discretization

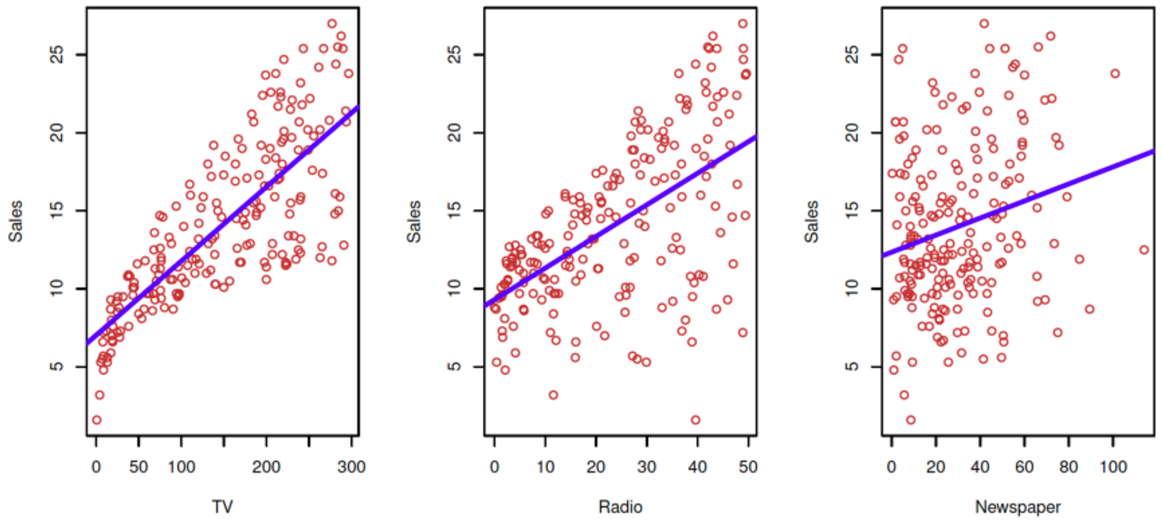
- Binning (i.e histograms)
  - Equal-width binning (more simple but can be affected by outliers)
  - Equal-depth binning (maintains data distribution shape)
- Regression analysis



Supervised Learning


Domain/Problem Understanding
Data Understanding
Data Mining
Data Preparation
Model Building
Testing & Evaluation
Deployment

Supervised learning



The graphics show the Sales vs. the marketing spend on TV, Radio i Newspaper, with blue the linear regresion line is shown for each of the graphs. The question is: Can we predict Sales using the above three variables? Perhaps we can create model to predict it:

$$Sales \approx f(TV, Radio, Newspaper)$$

Notation

Sales is the *response* or *target* that we wish to predict. TV is a *feature*, or *input variable*, we name it X_1. Likewise Radio is named as X_2 and Newspaper as X_3.
We can refer to the *input vector* or *feature vector* collectively as :

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

And the model can be written as:

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

where ϵ captures the measurement error and other discrepancies.

In other words...

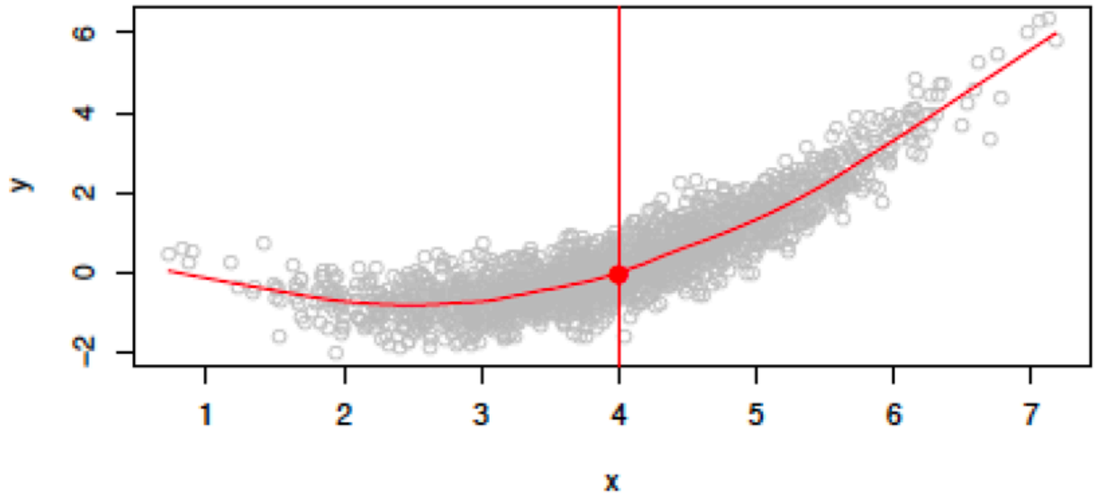
We want to learn the mapping f: $X \rightarrow Y$, where some $x \in X$ is some object and $y \in Y$ is the class label.
where in our case $x \in \mathbb{R}^3$, $\{y \in \mathbb{R} : y \geq 0\}$

What is a ***f(x)*** *good for?*

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

- We can understand which components of $X = (X_1, X_2, \ldots, X_p)$ 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.



Is there an **ideal** $f(X)$? In particular, what is a good value for $f(X)$ at any selected value of X, say $X = 4$? There can be many Y values at $X = 4$. A good value is:

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

$E(Y|X = 4)$ means **expected value** (average) of Y given $X = 4$.
This ideal f(x) = E(Y|X = x) is called **regression function**.

The Regression Function

Regression Function

- The regression function for a vector X defined in a :

$$f(\mathbf{x}) = f(x_1, x_2, x_3) = E(Y \,|\, X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

- The **ideal** or **optimal** predictor of Y with regard to mean-squared prediction error is the function that minimizes $E[(Y - g(x))^2 \,|\, X = x]$ over all function 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 Yvalues.

- For any estimate $\hat{f}(x)$ of $f(x)$, we have

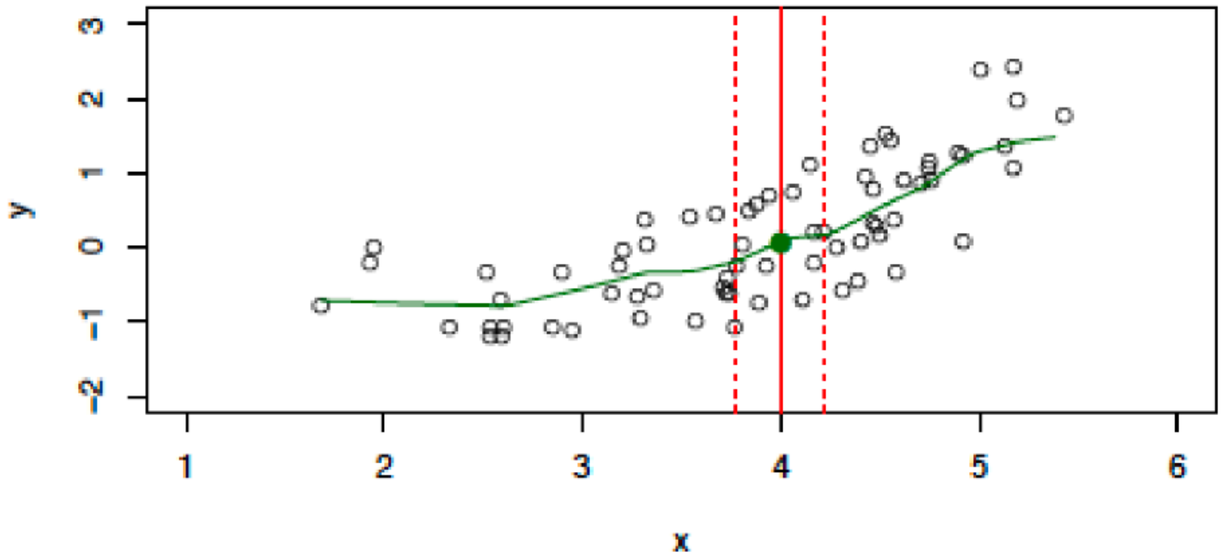
$$E[(Y - \hat{f}(X))^2 \,|\, X = x] = \underbrace{[f(x) - \hat{f}(x)]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

How to estimate the regression function *f*

Regression Function

- Typically we our dataset is small, or at least it is finite.
- If we want to estimate the predictor for X=4, it is normal that we do not have sample with this values. So, we cannot compute $E[Y|X=4]$
- We have to relax the definition. Let's denote a nearest neighbor regression model as follows:
  - $\hat{y} = Ave(Y|X \in N(X))$

Where $N(X)$ is some **neighborhood** of X



Regression Function

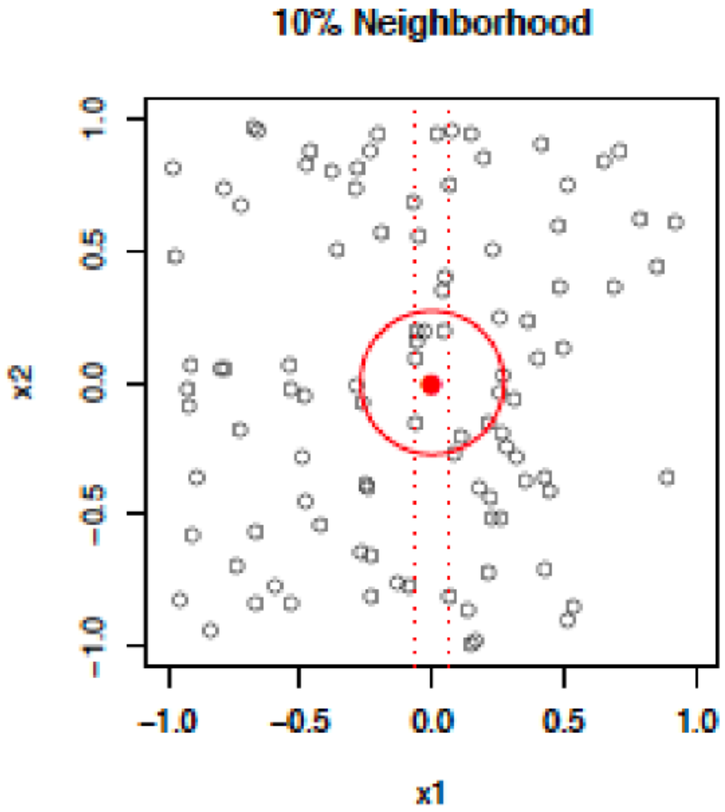
- Nearest neighbor averaging can be pretty good for small p - i.e. $p \leq 4$ and large N.

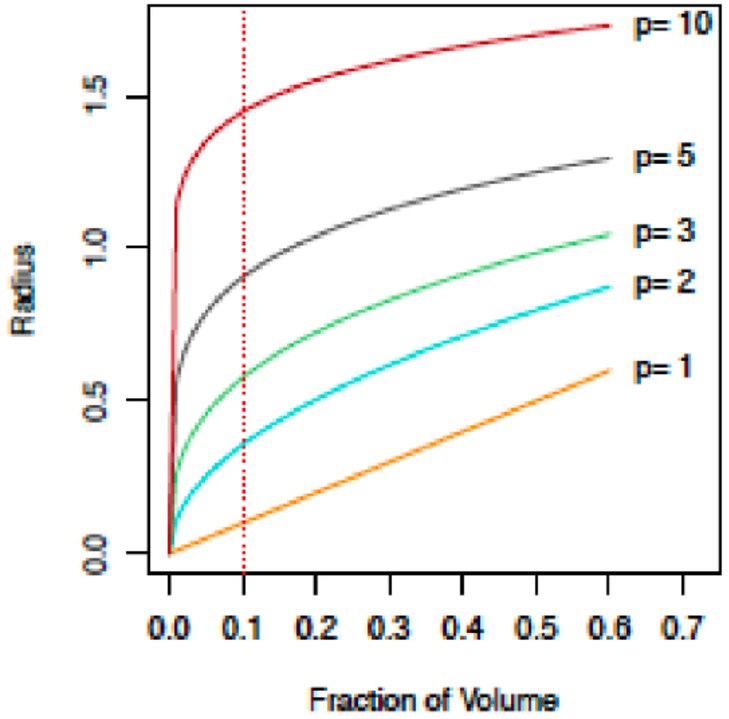
- Nearest neighbor methods can be **lousy** when p is large. This is because the effect cause **curse of dimensionality**. Nearest neighbors tent to be far away in high dimensions

  - We need to get a reasonable fraction of the N values of y_i to average to bring the variance down - e.g. 10%.

  - At 10% neighborhood in high dimension need no longer to be local, so we lose the spirit of estimating $E(Y|X = x)$ by local averaging.

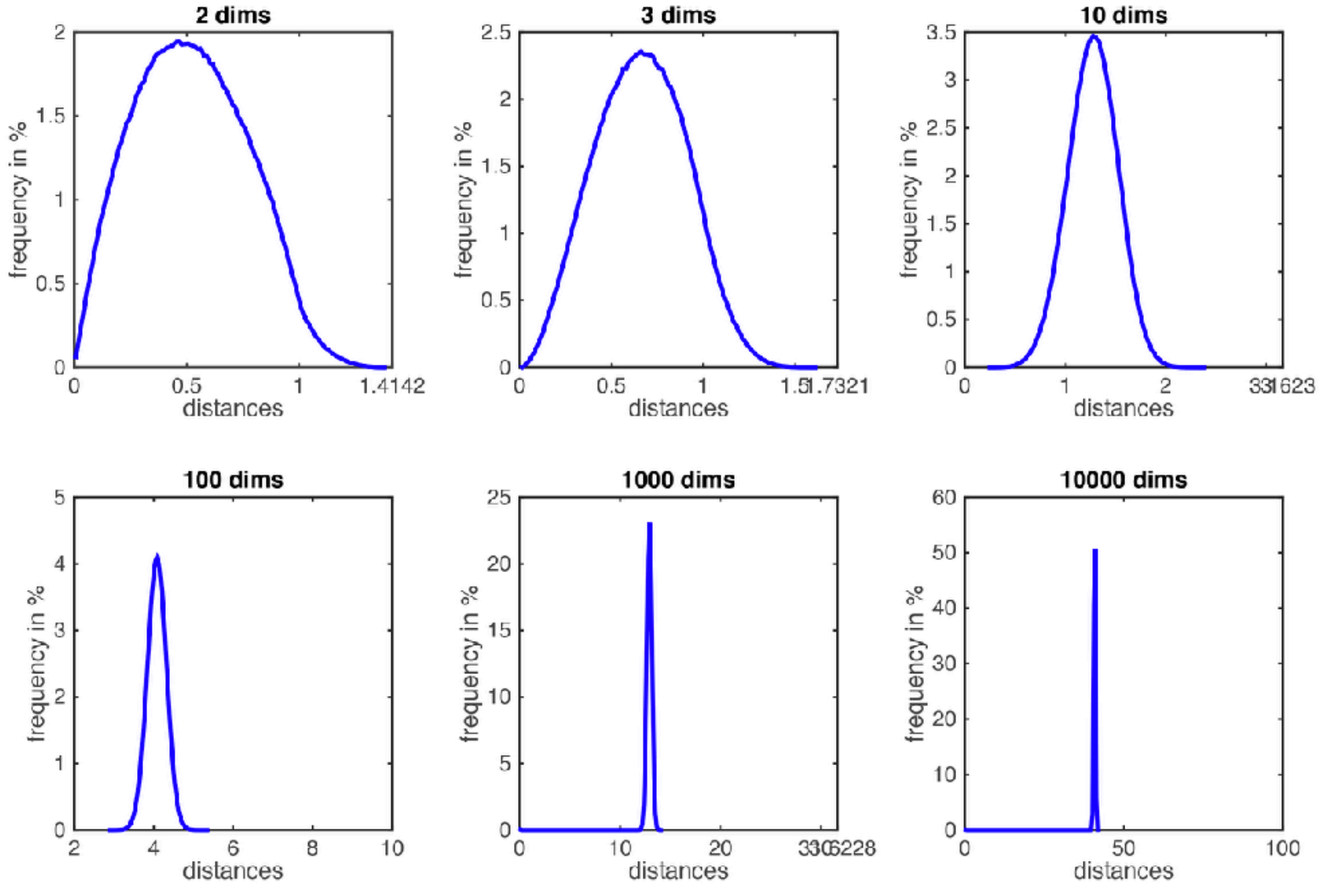
Regression Function





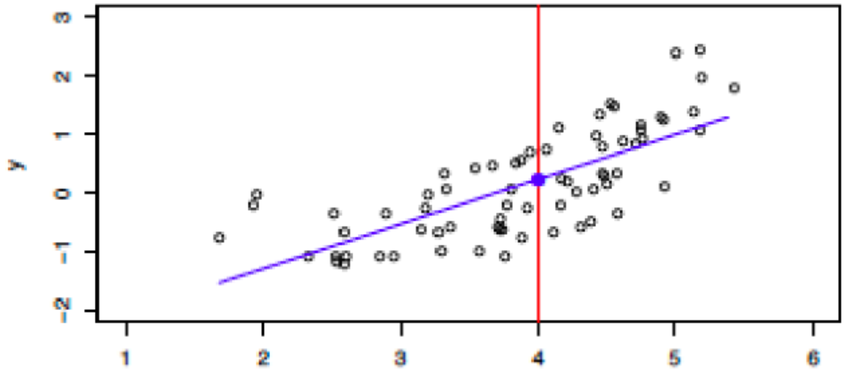
Curse of Dimensionality

K-NN classifier makes the assumption that similar points share similar labels.
Unfortunately, **in high dimensional spaces**, points that are drawn from a probability distribution, tend to **never** be **close** together.

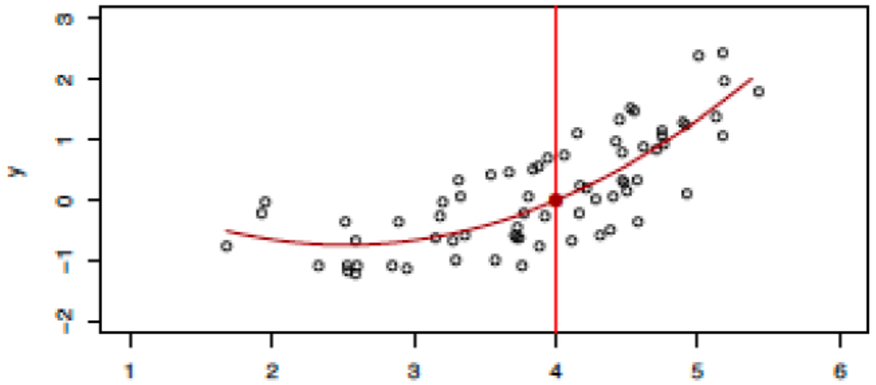


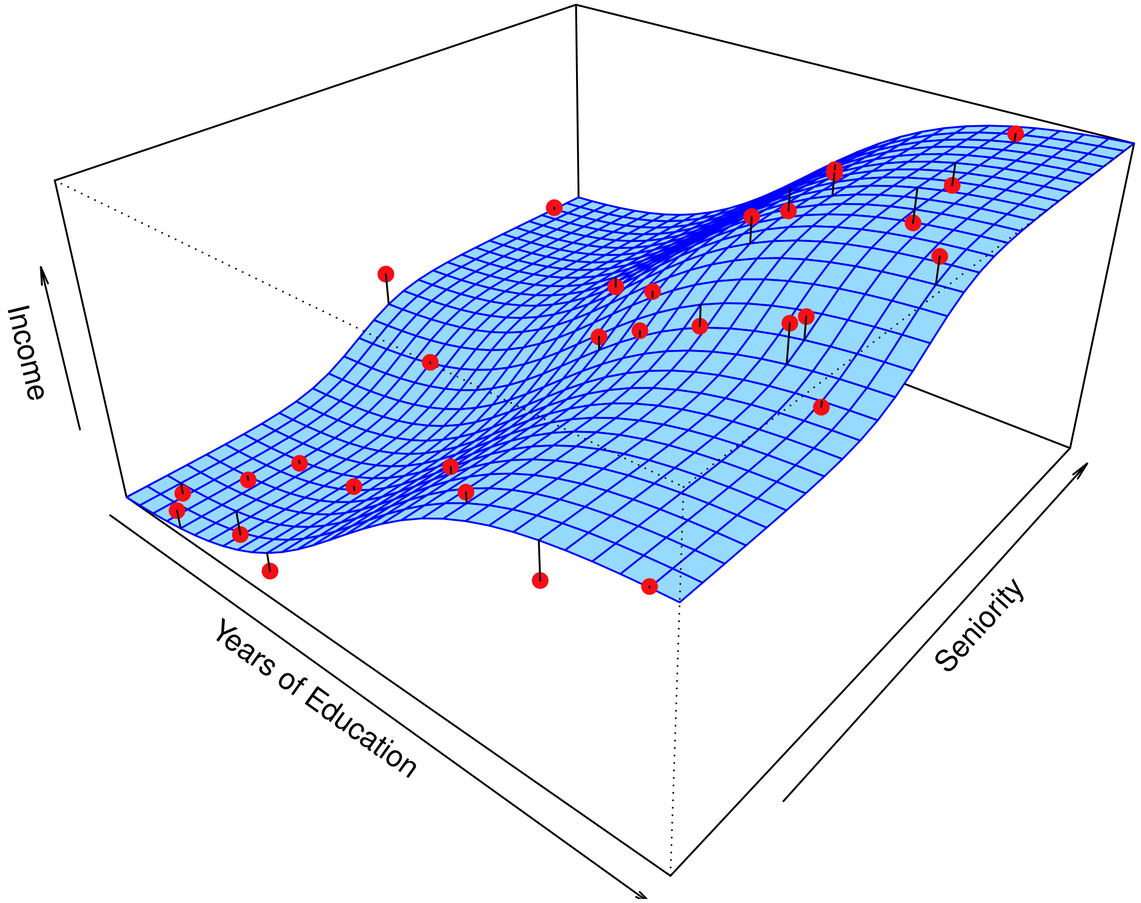
If this problem didn't exist, we would use the near neighbor averaging as the sole basis for doing estimation

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.

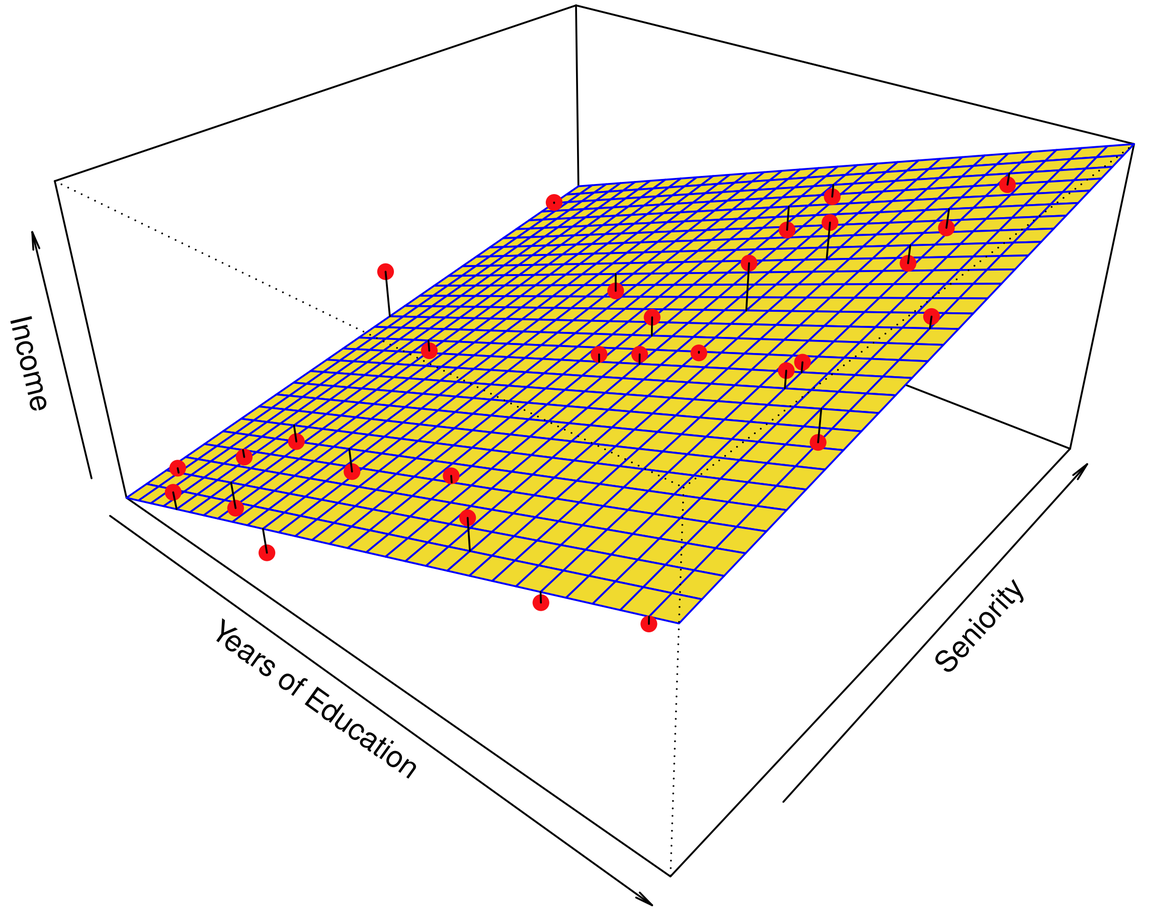




Simulated example. Red points are simulated values for income from the model

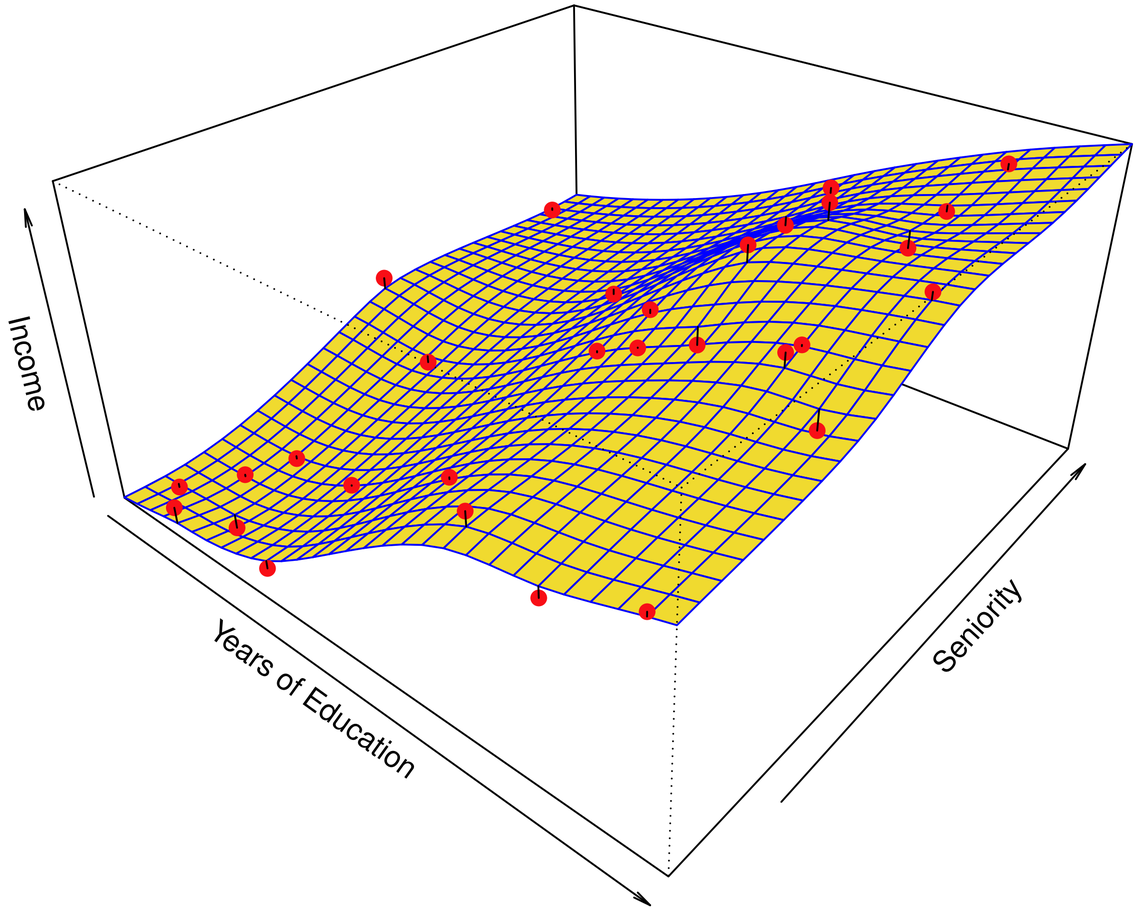
$$income = f(education, seniority) + \epsilon$$

f is the blue surface

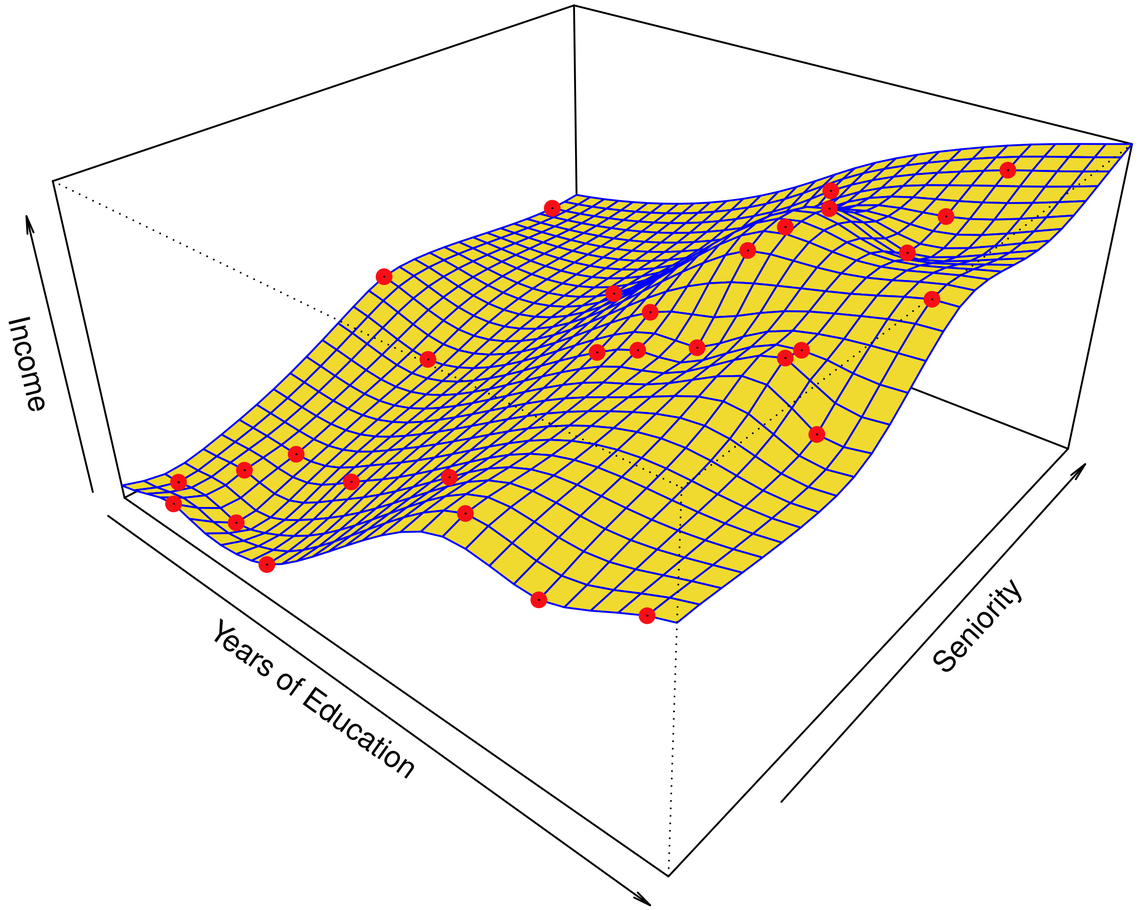


Linear regression model fit to the simulated data

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



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

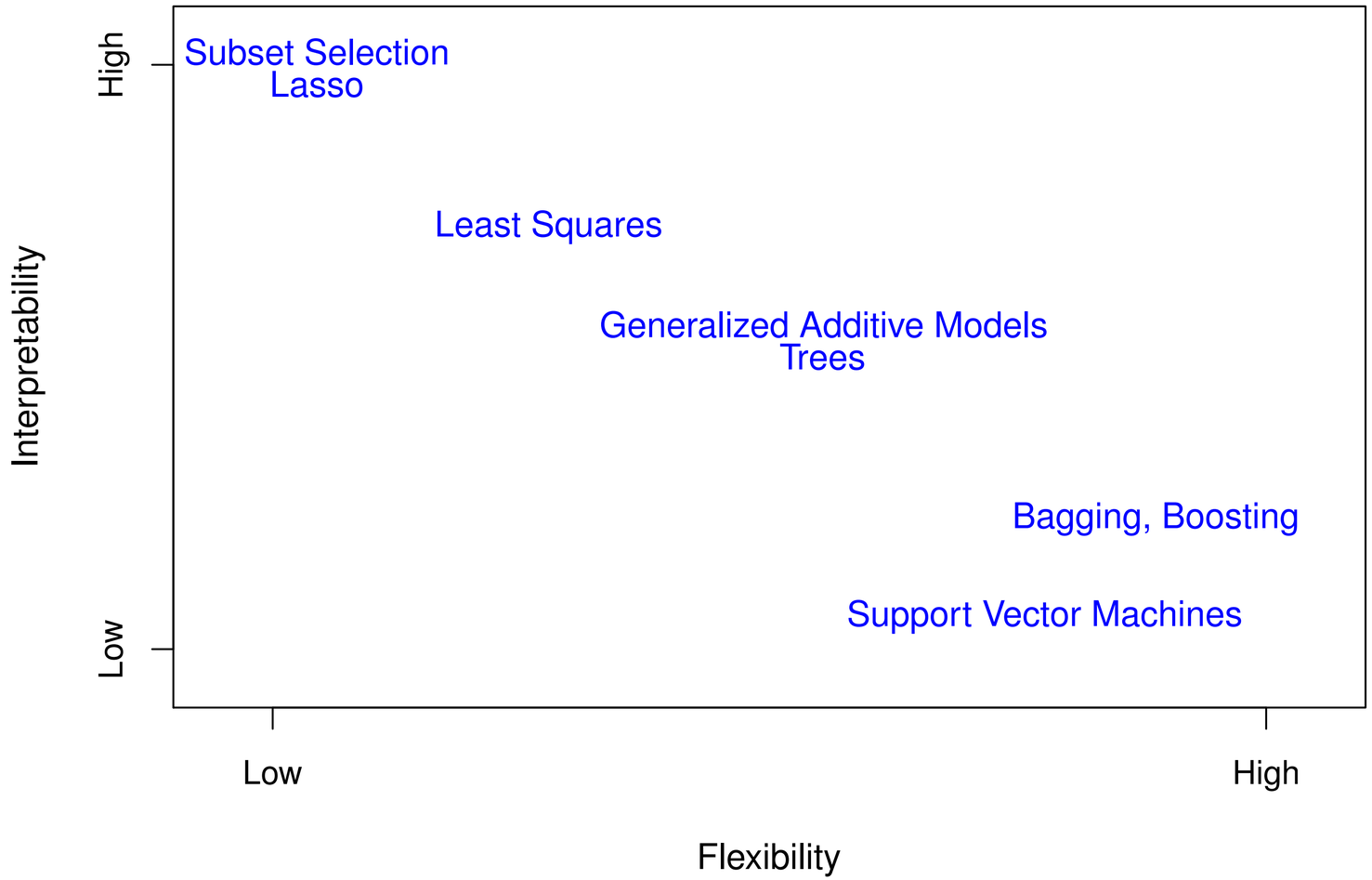


Even more flexible spline regression model $\hat{f}_S(\textit{education}, \textit{seniority})$ fit to the simulated data. Here the fitted model makes no errors on the training data! Also known as **overfitting**.

Trade-off

- Prediction Accuracy **vs** Interpretability
  - Linear models are easy to interpret, thin-plane spline models are not.
- **Good fit** vs **over-fit** or **under-fit.**
  - **How do we know it??**
- Parsimony versus black-box
  - We often prefer a simpler model involving fewer variables over a black-box predictor involving them all.

Interpretability vs. Flexibility/Complexity



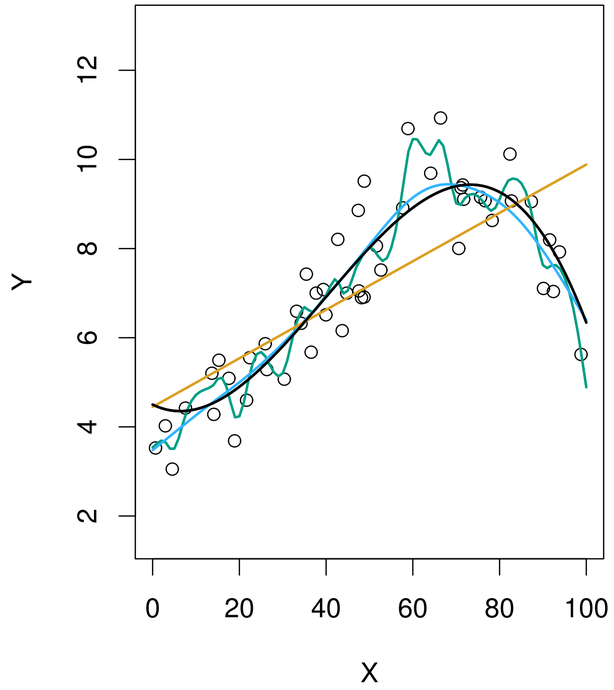
Assessing the Model Accuracy

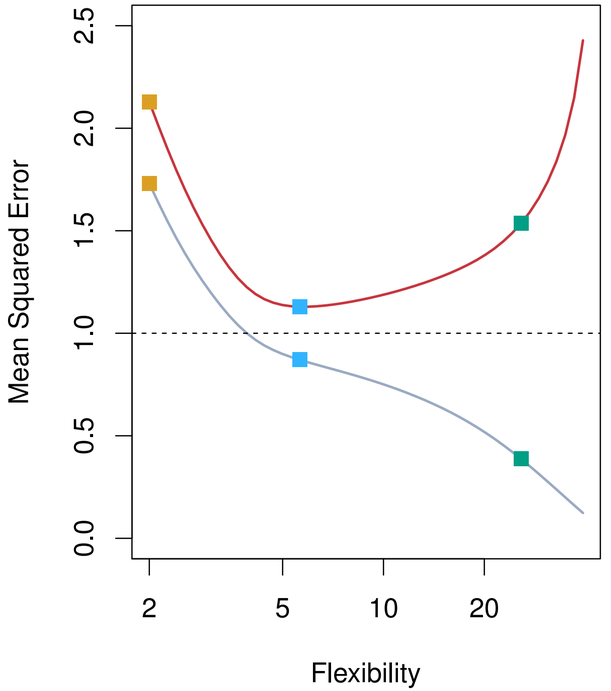
- Suppose we fit a model $\hat{f}(x)$ to some training data $X_{Tr} = \{x_i, y_i\}_1^N$, and we wish to see how well it performs.
  - We could compute the average squared prediction error over X_{Tr}

$$MSE_{Tr} = Ave_{i \in X_{Tr}}[y_i - \hat{f}(x_i)]^2$$

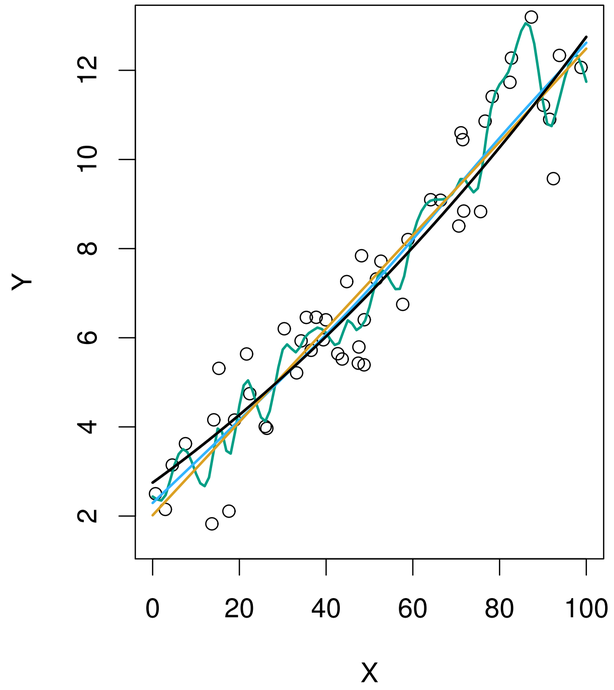
    - This may be biased toward more overfit models.
  - Instead we should, if possible, compute it using fresh **test** data $X_{Te} = \{x_i, y_i\}_1^m$:

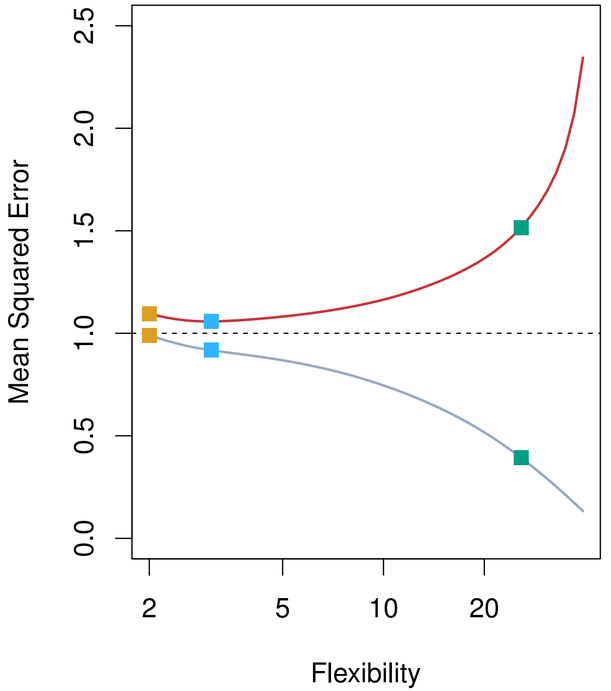
$$MSE_{Te} = Ave_{i \in X_{Te}}[y_i - \hat{f}(x_i)]^2$$



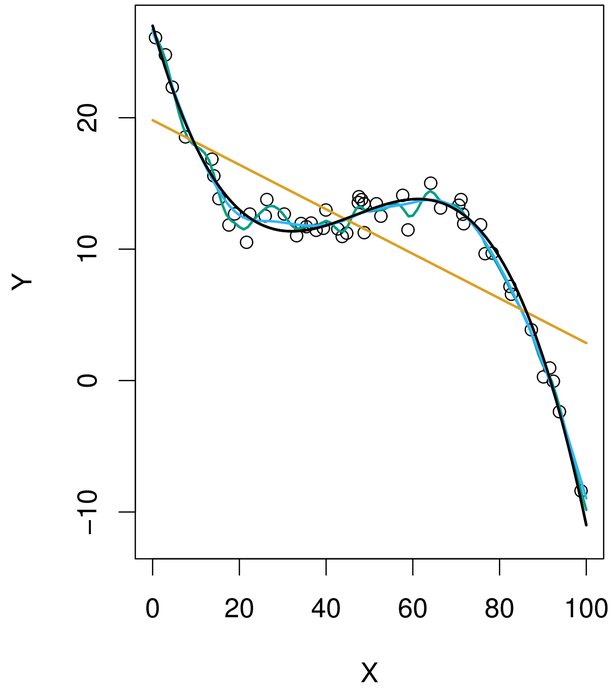


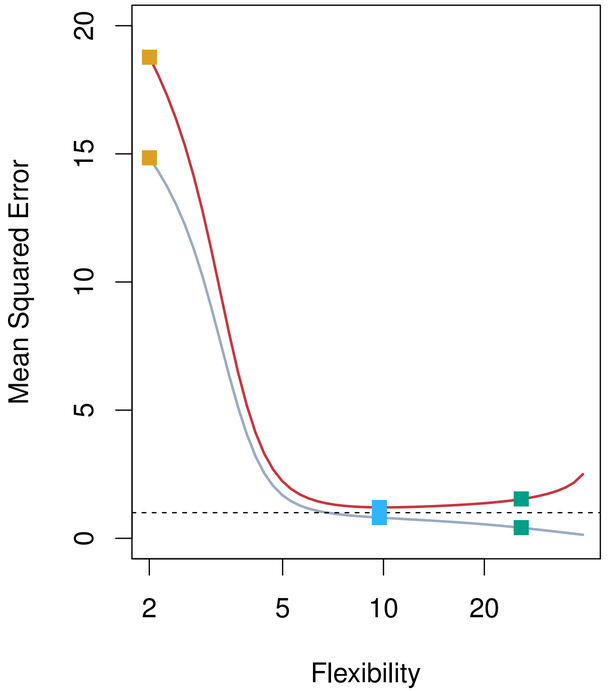
Black curve is truth. **Red** curve on the right is MSE_{Te}, **grey** curve is MSE_{Tr}. **Orange**, **blue** and **green** curves/squares correspond to fits of different flexibility models.





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





truth is wiggly and the noise is low, the more flexible methods fits do well

Bias-Variance Trade-off

- Suppose we have fit a model $\hat{f}(x)$ to some training data X_{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

$$E(y_0 - \hat{f}(x_0))^2 = \underbrace{Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

- So, $E(y_0 - \hat{f}(x_0))^2$ defines the **expected test MSE**, and refers to the average test MSE that would obtain if we repeatedly estimated f using a large set of training sets, and tested each at x_0. **Then expectation averages over the variability of y_0 as well as the variability in X_{Tr}.**

- The equation tells us that in order to minimize the expected test error, we need to select a method that simultaneously achieves low variance and low bias.

Bias-Variance Trade-off

- **Error due to Bias**: The error due to bias is taken as the difference between the expected (or average) prediction of our model and the correct value which we are trying to predict. Of course you only have one model so talking about expected or average prediction values might seem a little strange. However, imagine you could repeat the whole model building process more than once: each time you gather new data and run a new analysis creating a new model. Due to randomness in the underlying data sets, the resulting models will have a range of predictions. Bias measures how far off in general these models' predictions are from the correct value.

$$Bias(\hat{f}(x_0)) = E[\hat{f}(x_0) - f(x_0)]$$

- and refers to the error that is introduced by approximating a real-life problem.

Bias-Variance Trade-off

- **Error due to Variance:** The error due to variance is taken as the variability of a model prediction for a given data point. Again, imagine you can repeat the entire model building process multiple times. The variance is how much the predictions for a given point vary between different realizations of the model.

$$Var(\hat{f}(x_0)) = E[\hat{f}(x_0)^2] - (E[\hat{f}(x_0)])^2$$

- and refers to the amount by which $\hat{f}$ would change if we estimated it using a **different training data set**.

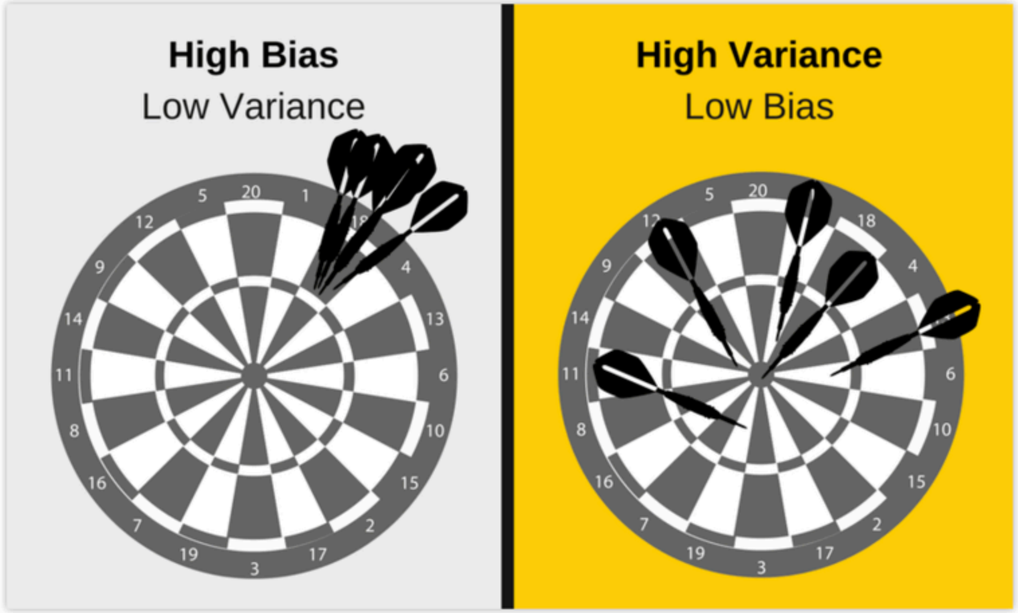
Bias-Variance Trade-off

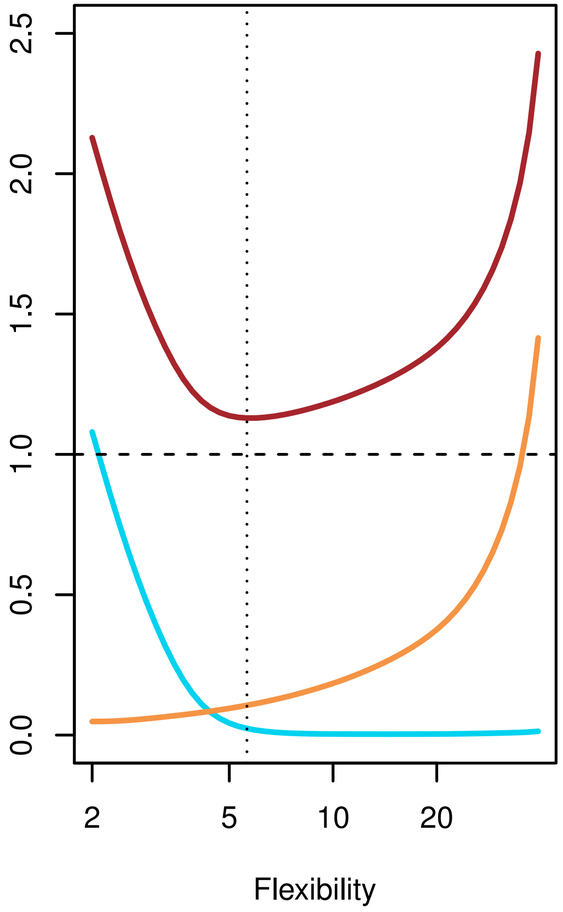
- Let’s do an experiment:

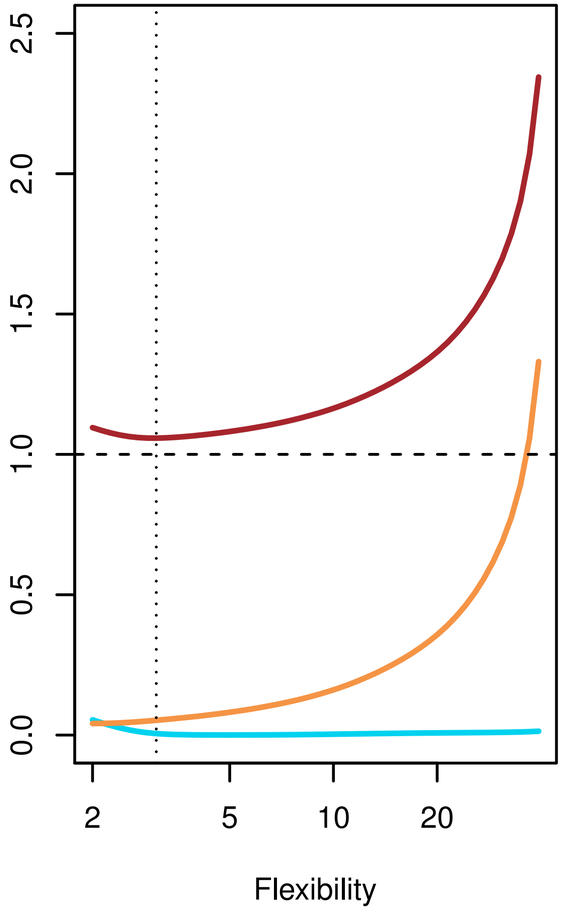
  1. Imagine you’ve collected 5 different training sets for the same problem.

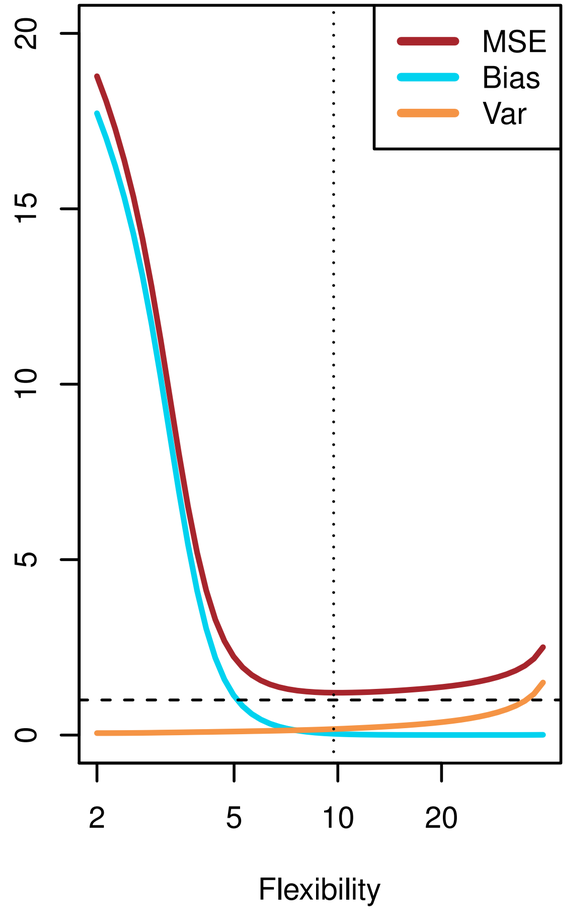
  2. Now imagine using one algorithm to train 5 models, one for each of your training sets.

  3. Bias vs. variance refers to the accuracy vs. consistency of the models trained by your algorithm.




0.0
0.5
1.0
1.5
2.0
2.5
2
5
10
20
Flexibility


0.0
0.5
1.0
1.5
2.0
2.5
2
5
10
20
Flexibility


MSE
Bias
Var
0
5
10
15
20
2
5
10
20
Flexibility

The Classification Function

Classification Problem

- Here the response variable Y is **qualitative** - e.g. email is one of $\mathscr{C} = (\text{spam}, \text{ham})(\text{ham} = goodemail)$, digit class is on of $\mathscr{C} = (0, 1 \ldots, 9)$. Our goals are to:
  - Build a classifier $C(X)$ that assigns a class label from $\mathscr{C}$ to a future unlabeled observation X.
  - Assess the uncertainty in each classification.
  - Understand the roles of the different predictors among $X = (X_1, X_2, \ldots, X_p)$.

The Bayes Classifier

- The error rate is minimized, on average, by very simple classifier that assigns each observation to the most likely class, given its predictor values.

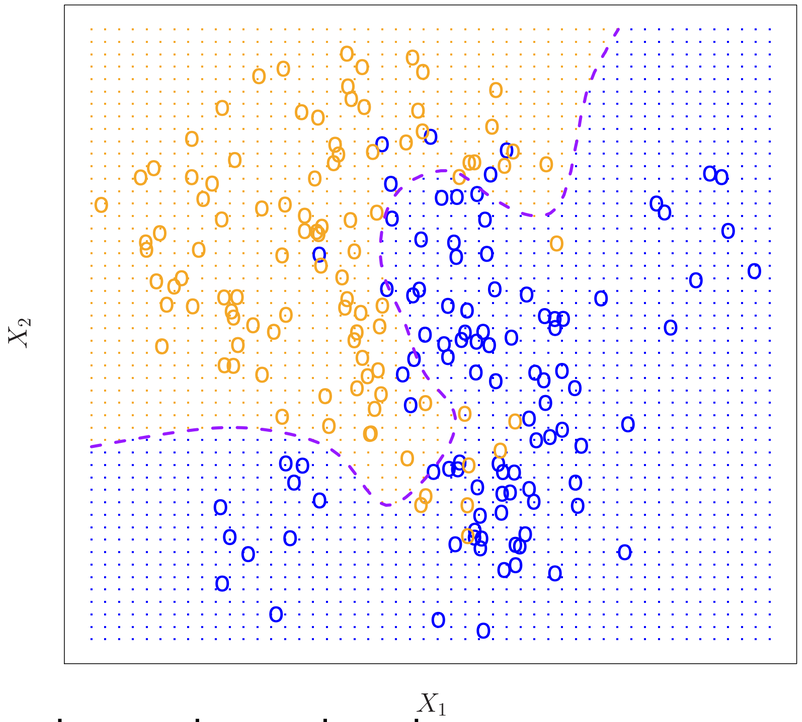
- Assigns a test observation with a feature vector x_0 to the class j for which:

$$P_r(Y = j \mid X = x_0)$$ is maximum.

- This is call the **conditional Probability**: it is the probability that $Y = j$, given the observed feature vector x_0 .

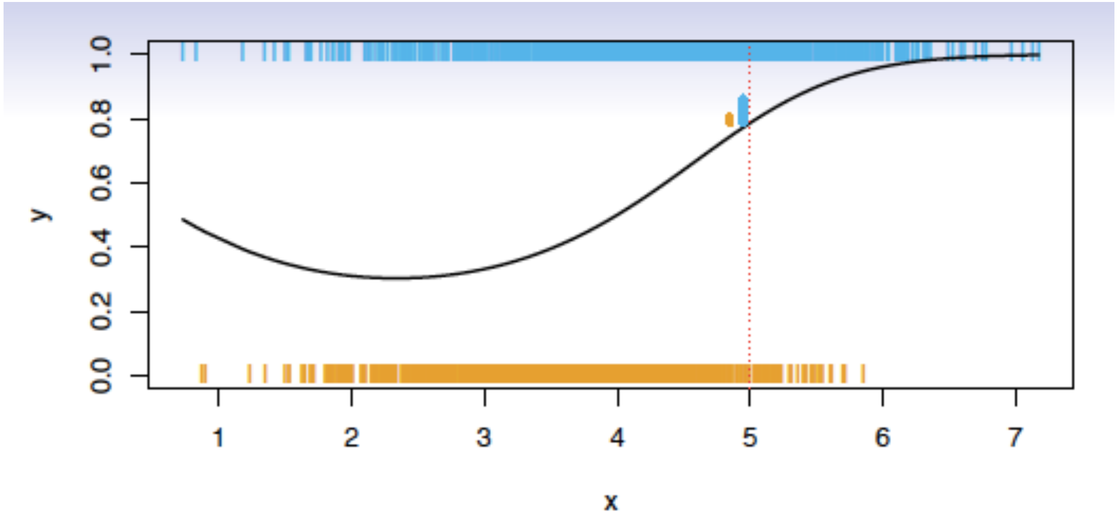
This is one of the most simple classifier called **Bayes Classifier**.

Example: The Bayes Classifier in two dimensions



- Since it is a simulated data, we know how the data was generated, and so, we can calculate the conditional probabilities for each value of X_1 and X_2 . The **orange** shaded region reflects the set of points for which $P_r(Y = organge \,|\, X)$ is greater than 50% while the **blue** shaded region indicates the set of points for which the probability is below 50%.

Example: The Bayes Classifier in two dimensions



- Suppose the K elements in $\mathscr{C}$ are numbered $1,2,...,K$. Let

$$p_k(x) = Pr(Y = k \mid X = x), k = 1,2,\ldots, K$$

- These are the **conditional class probabilities** at x; e.g. see little bar-plot at x = 5. Then the **Bayes optimal** classifier at x is

$$C(x) = j \text{ if } p_j(x) = max\{p_1(x), p_2(x), \ldots, p_K(x)\}$$

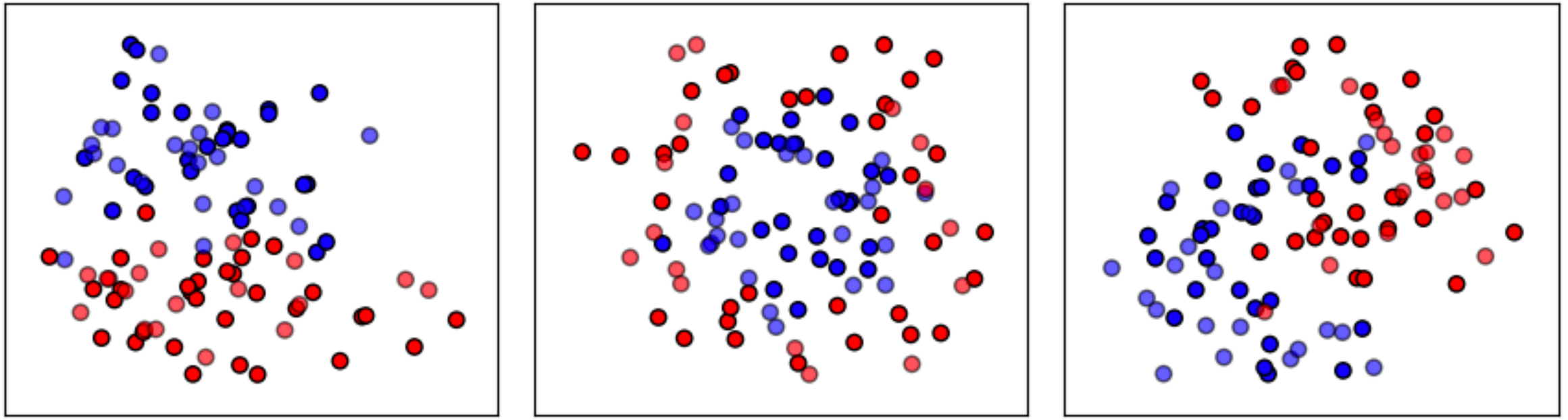
Example: The Bayes Classifier in two dimensions

- In **theory** we would like to predict qualitative responses using the Bayes Classifier. But for **real data we do not know the conditional distribution of Y given X**.

- Many approaches attempt to estimate the conditional distribution of X given Y, and then classify a given observation to the class with the highest estimated probability.

Classification Methods

Which is the optimal boundary for these datasets?



Classification Methods

K-NN: Distance based

Logistic Regression

Naïve Bayes

SVM

Tree Based Method

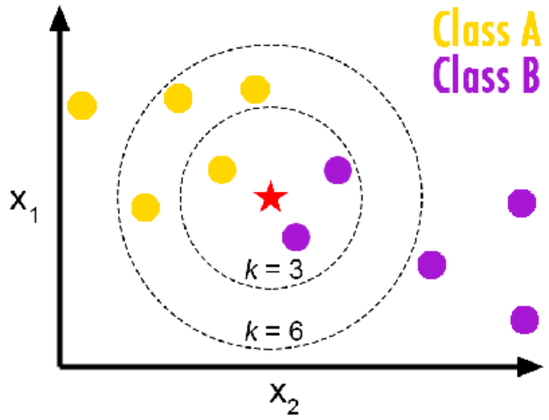
Neural Networks

K-Nearest Neighbor

- K-Nearest Neighbor is considered a lazy learning algorithm that classifies data sets based on their similarity with neighbors

- **Assumption**: Similar Inputs have similar outputs

- **Classification rule**: For a test input **x**, assign the most common label amongst its k most similar training inputs



★ is a new Example to be classified

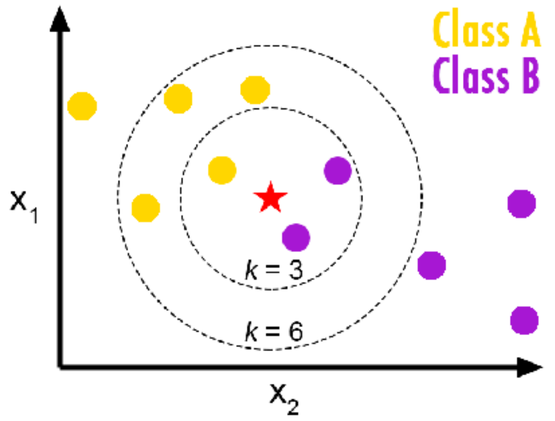
K stands for the number of neighbors to consider for the classification

K-Nearest Neighbor

- K-nearest neighbors (**KNN**) classifier is one of these methods.
  - Given a positive integer K and a test sample x_0 , KNN classifier first identifies the K points in the training data that are closest to x_0, represented by $\mathcal{N}_0$
  - Estimates the conditional probability for each j in $\mathscr{C}$

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

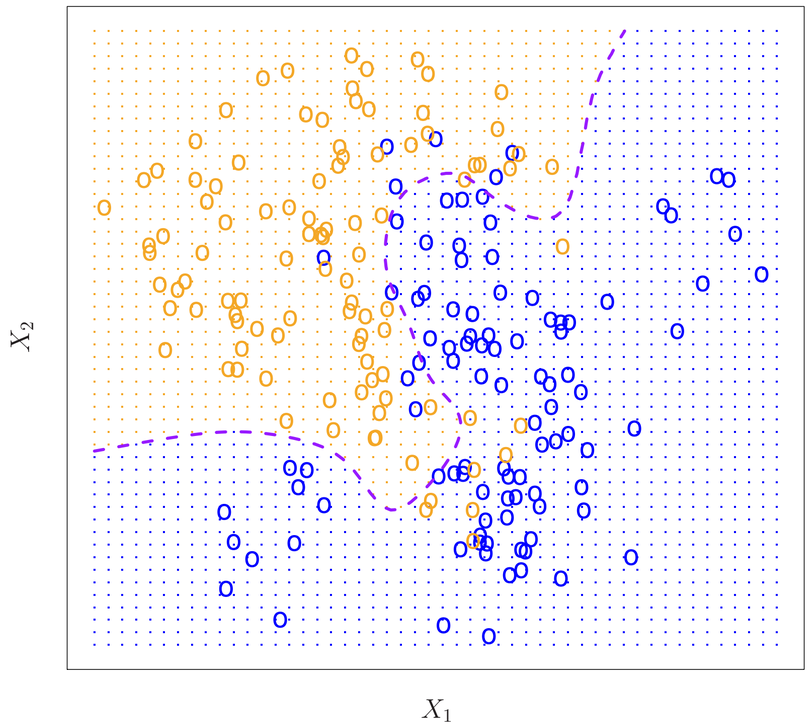
  - Classifies the test observation x_0 to the class with the maximum probability.



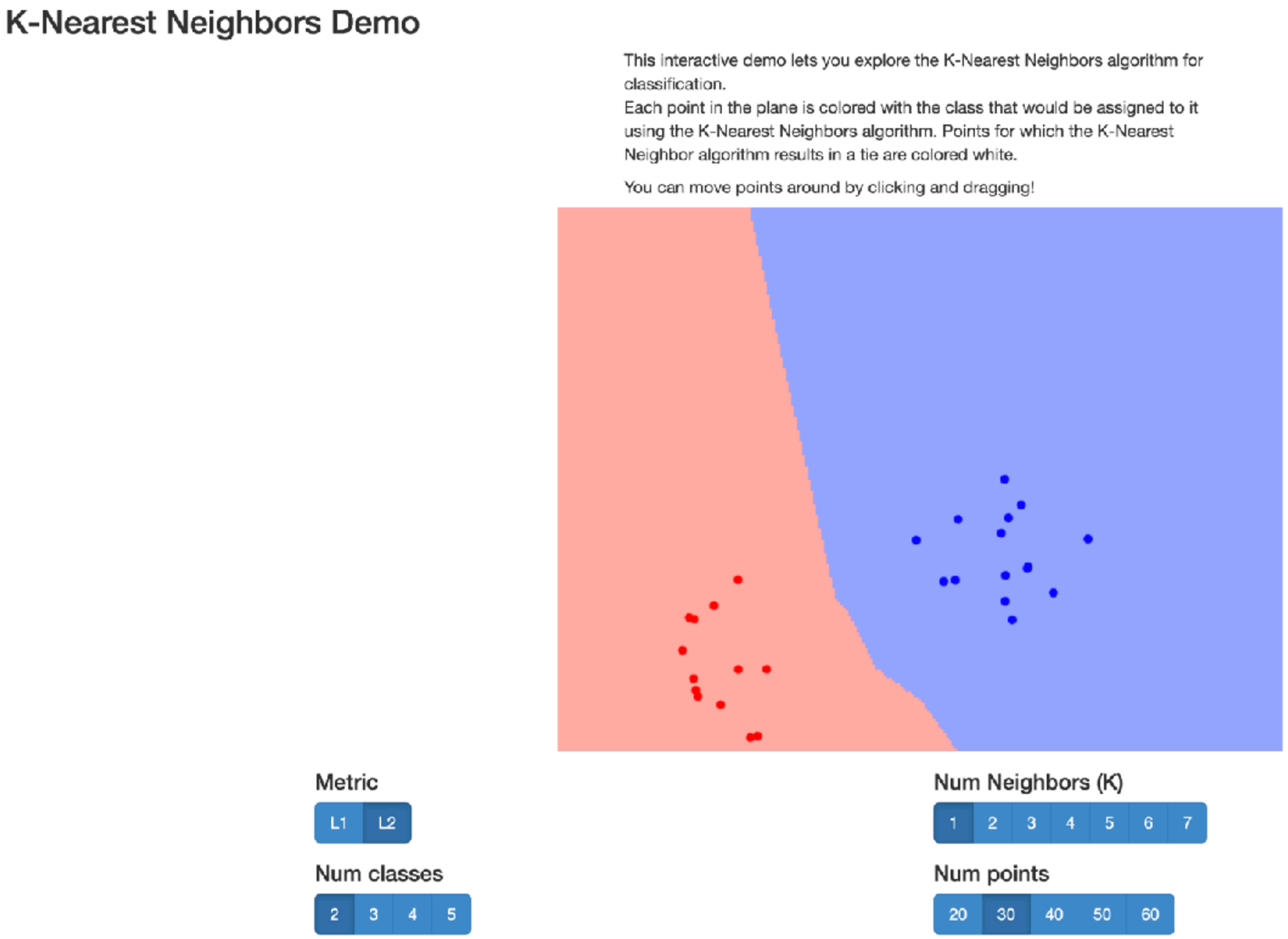
★ is a new Example to be classified

Example: K-nearest neighbors in two dimensions

What is the effect with different K values? **Let's think on it.**

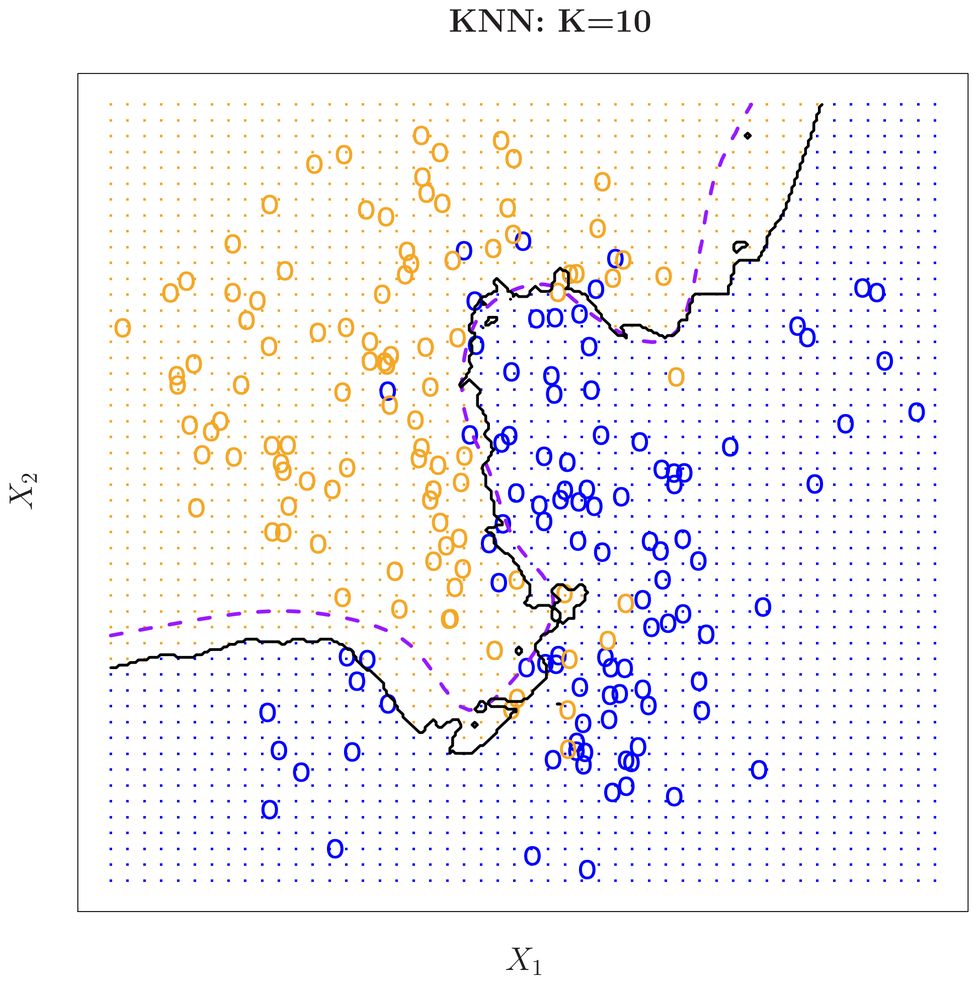


K-Nearest Neighbor

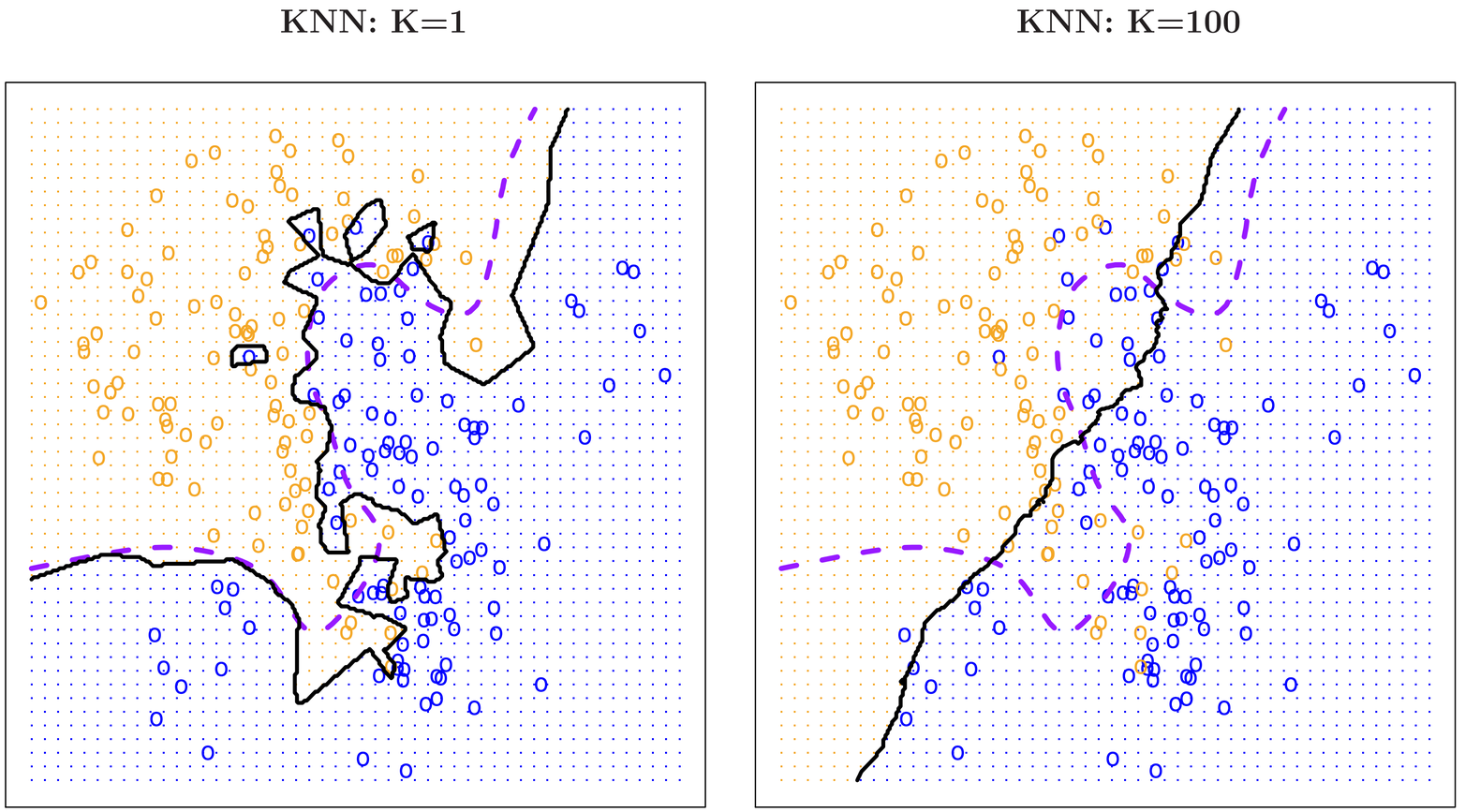


http://vision.stanford.edu/teaching/cs231n-demos/knn/

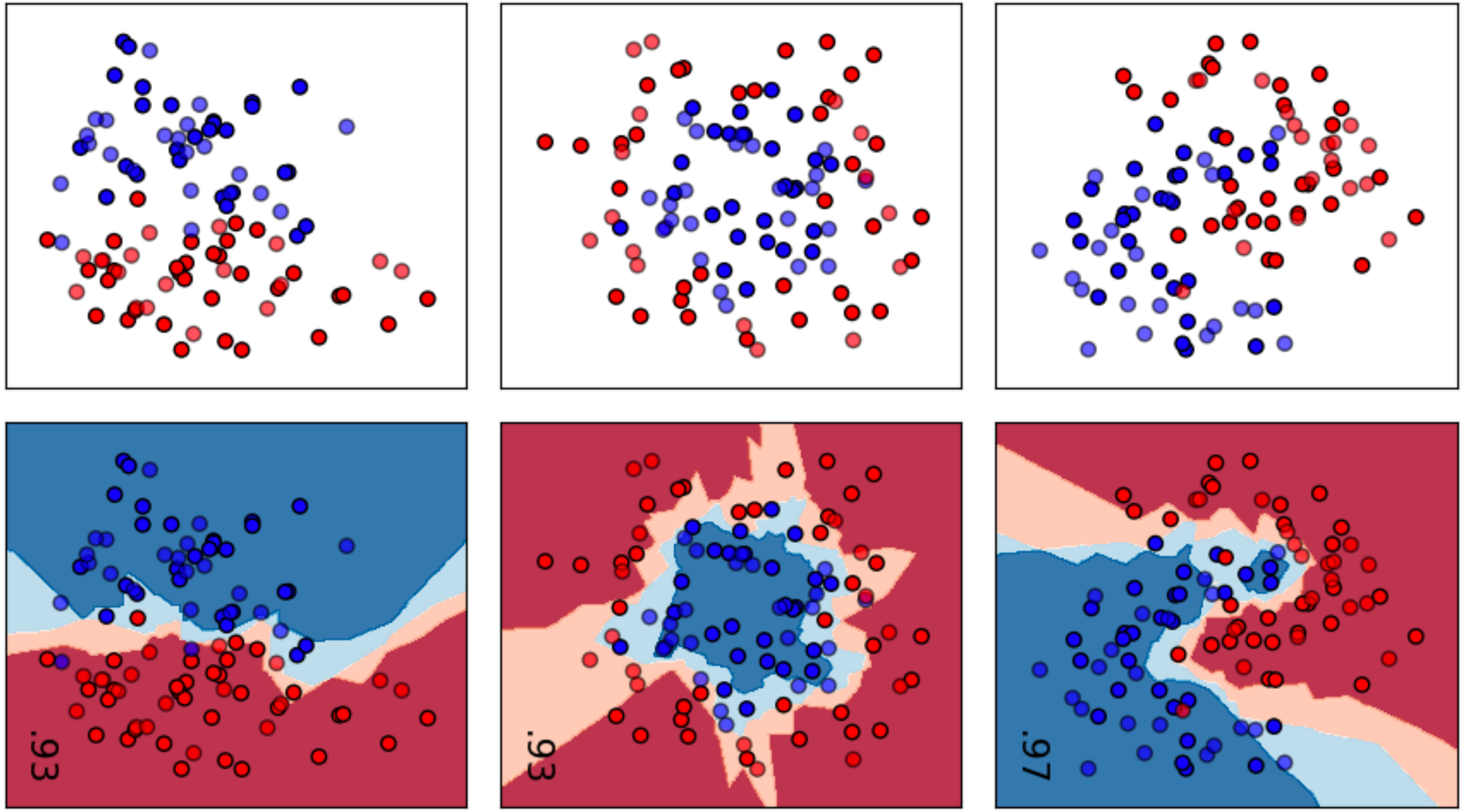
Example: K-nearest neighbors in two dimensions



Example: K-nearest neighbors in two dimensions



K-Nearest Neighbor



K-Nearest Neighbor - **Choosing the right K value?**

- **Choosing the right K value?**
  - To select the K that's right for your data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before.
- Some tips:
  - Small K-value = unstable/Overfitting
  - Large K-value = stable/small accuracy
  - Odd K-value to have a tiebreaker
- **Problems**:
  - Very **slow method** when the number of samples is large
  - ***Curse of dimensionality***

K-Nearest Neighbor - **Choosing the right K value?**