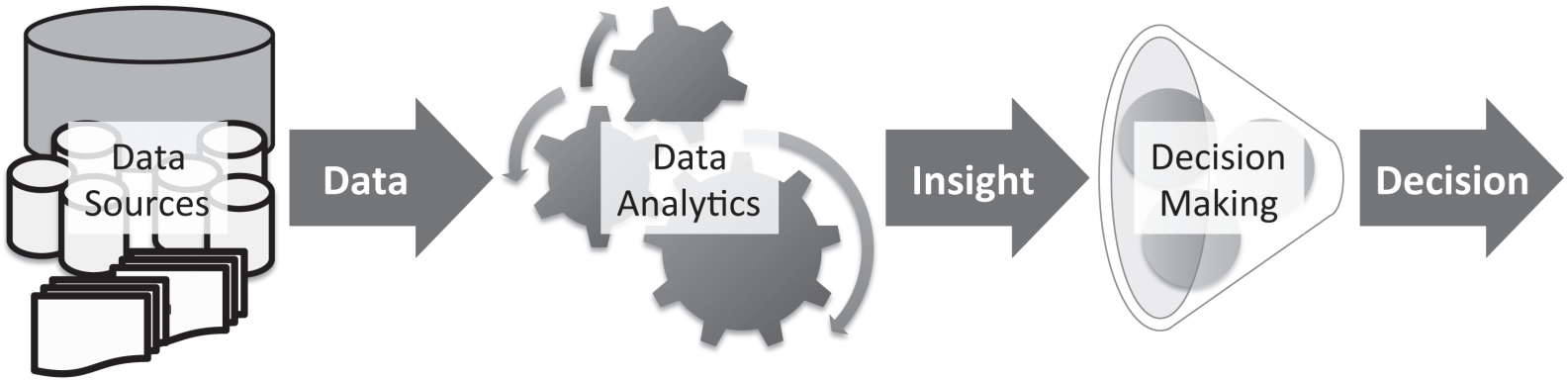
Introduction

Outline

- What are Predictive Data Analytics and Machine Learning?
- How Does Machine Learning Work?
- What Can Go Wrong With ML?
- Bias and Variance
- ML Toolkit

What is Predictive Data Analytics



Predictive data analytics moving from **data** to **insights** to **decisions**.

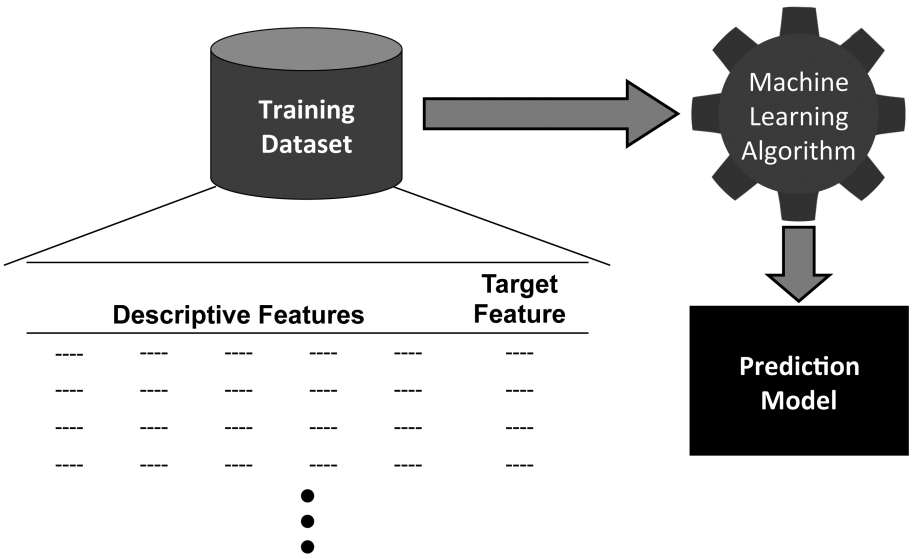
Example Applications:

- Price Prediction
- Fraud Detection
- Diagnosis
- Document Classification
- . . .

What is Machine Learning?

- (Supervised) Machine Learning techniques automatically learn a model of the relationship between a set of **descriptive features** and a **target feature** from a set of historical examples.

Roadmap of ML



Using ML to induce a prediction model from a training dataset.

Using the model to make predictions for new query instances.

A dataset indicates whether the mortgage applicant defaulted on the load

ID	Occupation	Age	Loan-Salary Ratio	Outcome
1	industrial	34	2.96	repaid
2	professional	41	4.64	default
3	professional	36	3.22	default
4	professional	41	3.11	default
5	industrial	48	3.80	default
6	industrial	61	2.52	repaid
7	professional	37	1.50	repaid
8	professional	40	1.93	repaid
9	industrial	33	5.25	default
10	industrial	32	4.15	default

- What is the relationship between the **descriptive features** (Occupation, Age, Loan-Salary Ratio) and the **target feature** (Outcome)?

if LOAN-SALARY RATIO > 3 **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

if LOAN-SALARY RATIO > 3 **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

- This is an example of a **prediction model**

if LOAN-SALARY RATIO > 3 **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

- This is an example of a **prediction model**
- This is also an example of a **consistent** prediction model

if LOAN-SALARY RATIO > 3 **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

- This is an example of a **prediction model**
- This is also an example of a **consistent** prediction model
- Notice that this model does not use all the features and the feature that it uses is a derived feature (in this case a ratio): **feature design** and **feature selection** are two important topics that we will return to again and again.

Relationship between the descriptive features and the target feature (OUTCOME)?

ID	Amount	Salary	Loan-Salary Ratio	Age	Occupation	House	Type	Outcome
1	245,100	66,400	3.69	44	industrial	farm	stb	repaid
2	90,600	75,300	1.2	41	industrial	farm	stb	repaid
3	195,600	52,100	3.75	37	industrial	farm	ftb	default
4	157,800	67,600	2.33	44	industrial	apartment	ftb	repaid
5	150,800	35,800	4.21	39	professional	apartment	stb	default
6	133,000	45,300	2.94	29	industrial	farm	ftb	default
7	193,100	73,200	2.64	38	professional	house	ftb	repaid
8	215,000	77,600	2.77	17	professional	farm	ftb	repaid
9	83,000	62,500	1.33	30	professional	house	ftb	repaid
10	186,100	49,200	3.78	30	industrial	house	ftb	default
11	161,500	53,300	3.03	28	professional	apartment	stb	repaid
12	157,400	63,900	2.46	30	professional	farm	stb	repaid
13	210,000	54,200	3.87	43	professional	apartment	ftb	repaid
14	209,700	53,000	3.96	39	industrial	farm	ftb	default
15	143,200	65,300	2.19	32	industrial	apartment	ftb	default
16	203,000	64,400	3.15	44	industrial	farm	ftb	repaid
17	247,800	63,800	3.88	46	industrial	house	stb	repaid
18	162,700	77,400	2.1	37	professional	house	ftb	repaid
19	213,300	61,100	3.49	21	industrial	apartment	ftb	default
20	284,100	32,300	8.8	51	industrial	farm	ftb	default
21	154,000	48,900	3.15	49	professional	house	stb	repaid
22	112,800	79,700	1.42	41	professional	house	ftb	repaid
23	252,000	59,700	4.22	27	professional	house	stb	default
24	175,200	39,900	4.39	37	professional	apartment	stb	default
25	149,700	58,600	2.55	35	industrial	farm	stb	default

if LOAN-SALARY RATIO < 1.5 **then**
  OUTCOME=*'repay'*
else if LOAN-SALARY RATIO > 4 **then**
  OUTCOME=*'default'*
else if AGE < 40 **and** OCCUPATION $=$*'industrial'* **then**
  OUTCOME=*'default'*
else
  OUTCOME=*'repay'*
end if

- The real value of machine learning becomes apparent in situations like this when we want to build prediction models from large datasets with multiple features.

How Does ML Work?

- Machine learning algorithms work by searching through a set of possible prediction models for the model that best captures the relationship between the descriptive features and the target feature.
- An obvious search criteria to drive this search is to look for models that are consistent with the data.
- However, because a training dataset is only a sample ML is an ill-posed problem.

A simple retail dataset

ID	BBY	ALC	ORG	GRP
1	no	no	no	couple
2	yes	no	yes	family
3	yes	yes	no	family
4	no	no	yes	couple
5	no	yes	yes	single

A full set of potential prediction models before any training data becomes available.

BBY	ALC	ORG	GRP	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...	$\mathbb{M}_{6\,561}$
no	no	no	?	couple	couple	single	couple	couple		couple
no	no	yes	?	single	couple	single	couple	couple		single
no	yes	no	?	family	family	single	single	single		family
no	yes	yes	?	single	single	single	single	single		couple
yes	no	no	?	couple	couple	family	family	family	...	family
yes	no	yes	?	couple	family	family	family	family		couple
yes	yes	no	?	single	family	family	family	family		single
yes	yes	yes	?	single	single	family	family	couple		family

- How to calculate the possible combinations?
- What is the number of descriptive features? - 2^3
- What is the number of target features? - 3
- So in total $3^8 = 6561$

A sample of the models that are consistent with the training data

BBY	ALC	ORG	GRP	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...	$\mathbb{M}_{6\,561}$
no	no	no	couple	couple	couple	single	couple	couple		couple
no	no	yes	couple	single	couple	single	couple	couple		single
no	yes	no	?	family	family	single	single	single		family
no	yes	yes	single	single	single	single	single	single		couple
yes	no	no	?	couple	couple	family	family	family	...	family
yes	no	yes	family	couple	family	family	family	family		couple
yes	yes	no	family	single	family	family	family	family		single
yes	yes	yes	?	single	single	family	family	couple		family

- How many potential models remain consistent left?
  - $3^3 = 27$
- It is because a single consistent model cannot be found based on a sample training dataset that ML is ill-posed.
- Why multiple models are not good?
  - Contradicting predictions: check results of M2, M4, M5, while BBY = Yes, ALC= Yes, ORG = Yes

- Consistency $\approx$ memorizing the dataset.
  - consistency with the training data doesn't provide guidance
  - Doesn't generalize for queries outside training data!
  - Consistency with noise in the data isn't desirable
- Objective of ML:
  - To learn a model which well-adapt it on "new samples", not just on training set.
  - Generalization: The ability of a model to adapt on new data set.
- In general, our samples obey Independent Identically Distributed (IID). More training samples may result in better generalization ability.

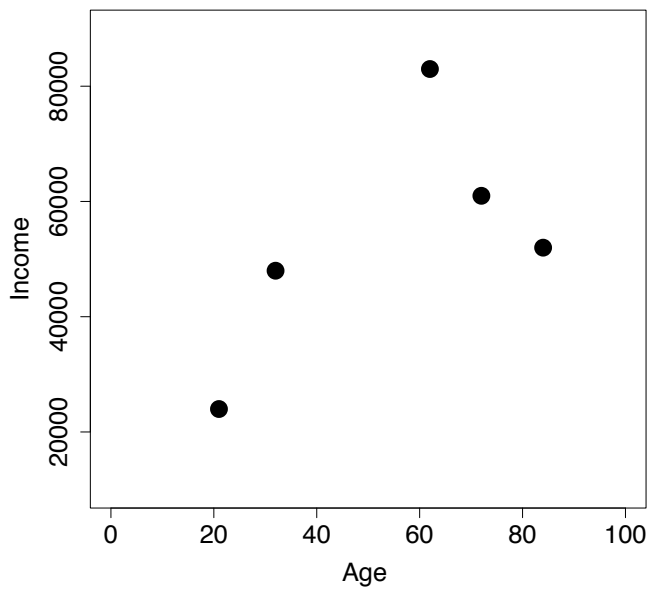
- Inductive bias the set of assumptions that define the model selection criteria of an ML algorithm.
- There are two types of bias that we can use:
  1. restriction bias
  2. preference bias
- Inductive bias is necessary for learning (beyond the dataset).

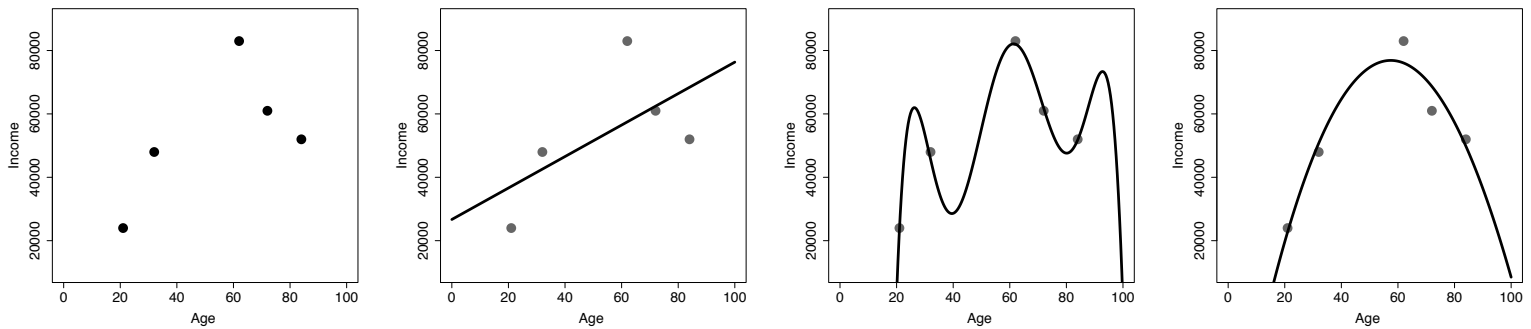
What Can Go Wrong With ML?

- No free lunch!
- What happens if we choose the wrong inductive bias:
  1. **underfitting**
  2. **overfitting**

The age-income dataset.

ID	AGE	INCOME
1	21	24,000
2	32	48,000
3	62	83,000
4	72	61,000
5	84	52,000





(a) Dataset (b) Underfitting (c) Overfitting (d) Just right

Striking a balance between overfitting and underfitting when trying to predict age from income.

Where Does the Error Come From?

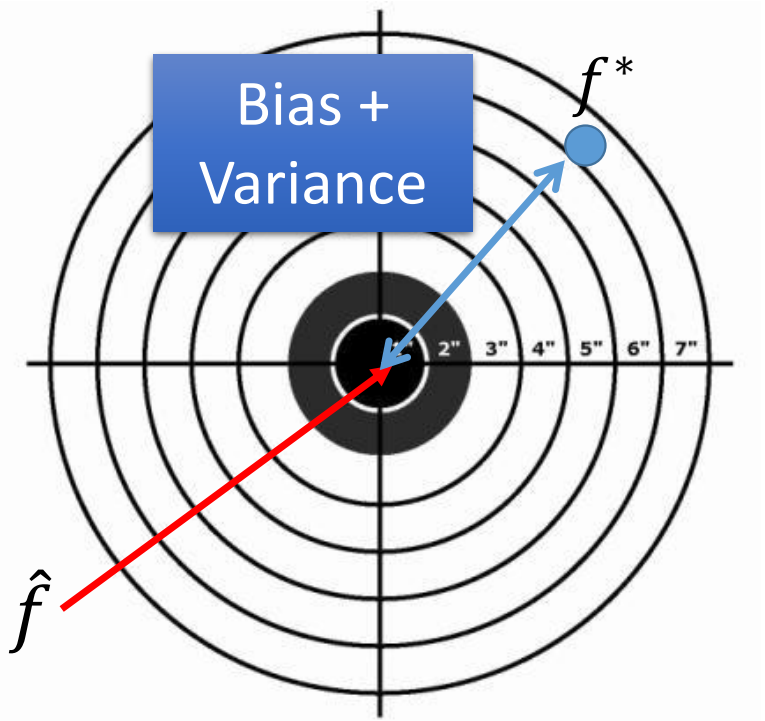
Estimator

$$\hat{y} = \hat{f}(\quad)$$

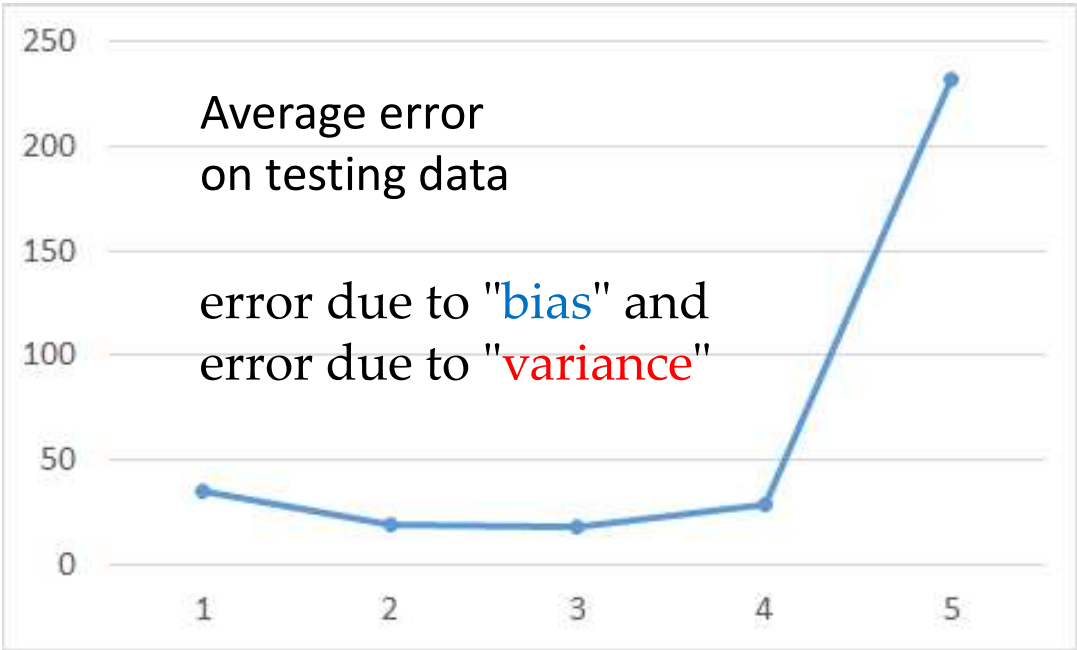
Only Niantic knows $\hat{f}$

From training data, we find f^*

f^* is an estimator of $\hat{f}$



$$y = \sum_n w_n x^n$$



A more complex model does not always lead to better performance on ***testing data***.

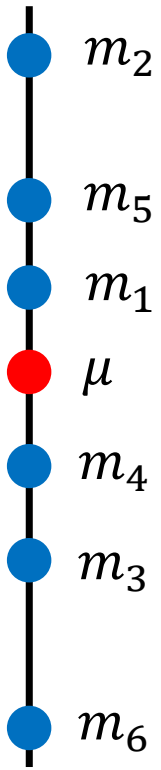
Bias and Variance of Estimator

- Estimate the mean of a variable x
  - assume the mean of x is μ
  - assume the variance of x is σ^2
- Estimator of mean μ
  - Sample N points: $\{x^1, x^2, \ldots, x^N\}$

$$m = \frac{1}{N}\sum_n x^n \neq \mu$$

$$E[m] = E\left[\frac{1}{N}\sum_n x^n\right] = \frac{1}{N}\sum_n E[x^n] = \mu$$

unbiased



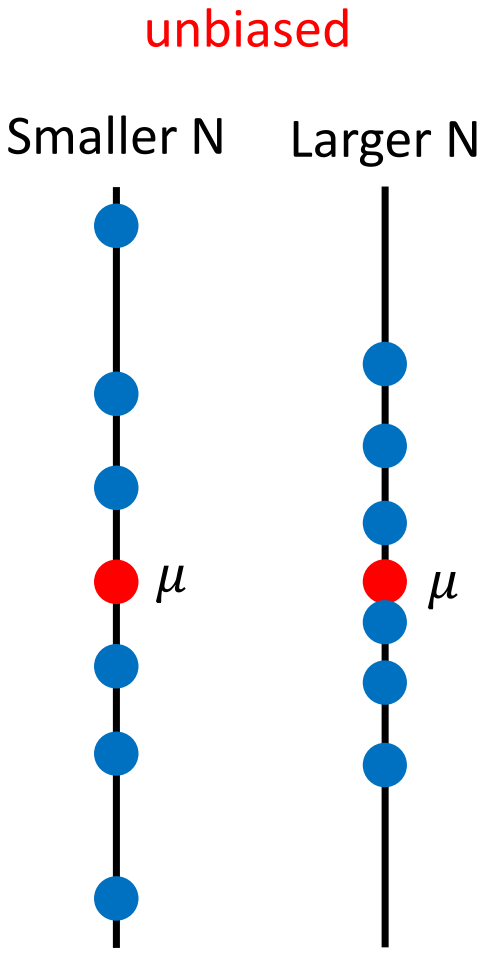
Bias and Variance of Estimator

- Estimate the mean of a variable x
  - assume the mean of x is μ
  - assume the variance of x is σ^2
- Estimator of mean μ
  - Sample N points: $\{x^1, x^2, \ldots, x^N\}$

$$m = \frac{1}{N}\sum_n x^n \neq \mu$$

$$Var[m] = \frac{\sigma^2}{N}$$

Variance depends on the number of samples



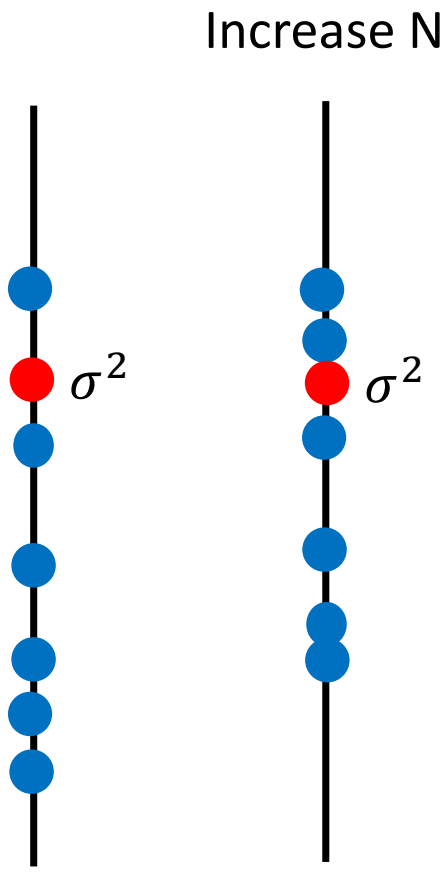
Bias and Variance of Estimator

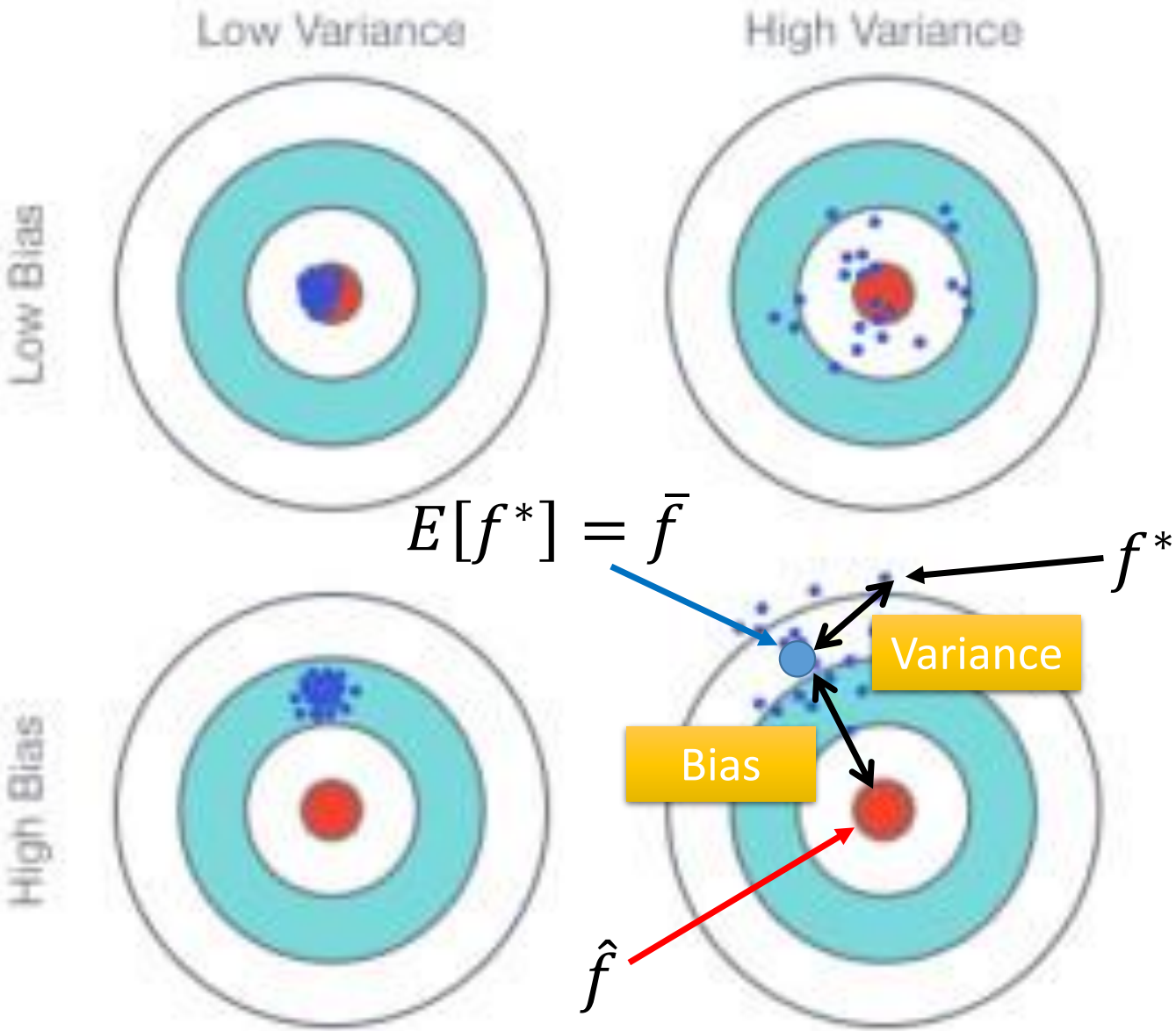
- Estimate the mean of a variable x
  - assume the mean of x is μ
  - assume the variance of x is σ^2
- Estimator of variance σ^2
  - Sample N points: $\{x^1, x^2, \ldots, x^N\}$

$$m = \frac{1}{N}\sum_n x^n \quad s^2 = \frac{1}{N}\sum_n (x^n - m)^2$$

Biased estimator

$$E[s^2] = \frac{N-1}{N}\sigma^2 \neq \sigma^2$$

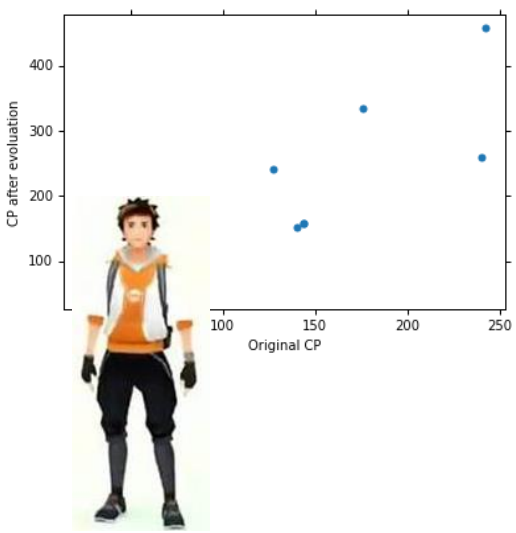



Low Variance
High Variance
Low Bias
High Bias
$E[f^*] = \bar{f}$
f^*
Variance
Bias
$\hat{f}$

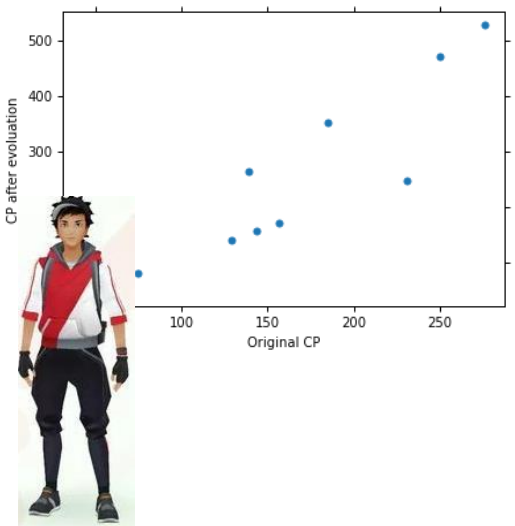
Parallel Universes

- In all the universes, we are collecting (catching) 10 Pokémons as training data to find f^*

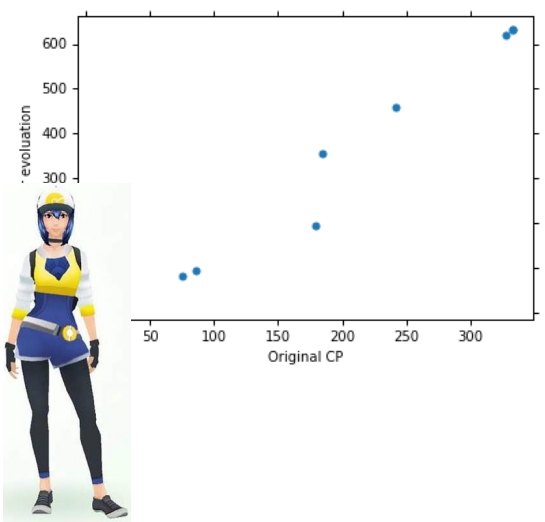
Universe 1



Universe 2



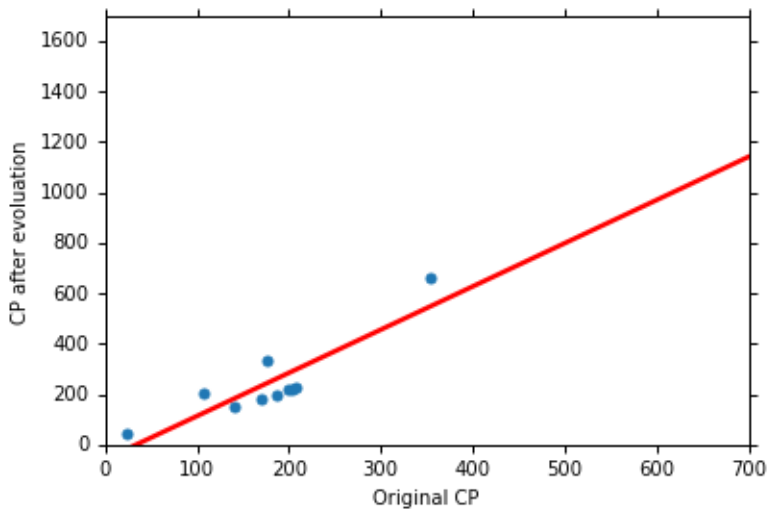
Universe 3



Parallel Universes

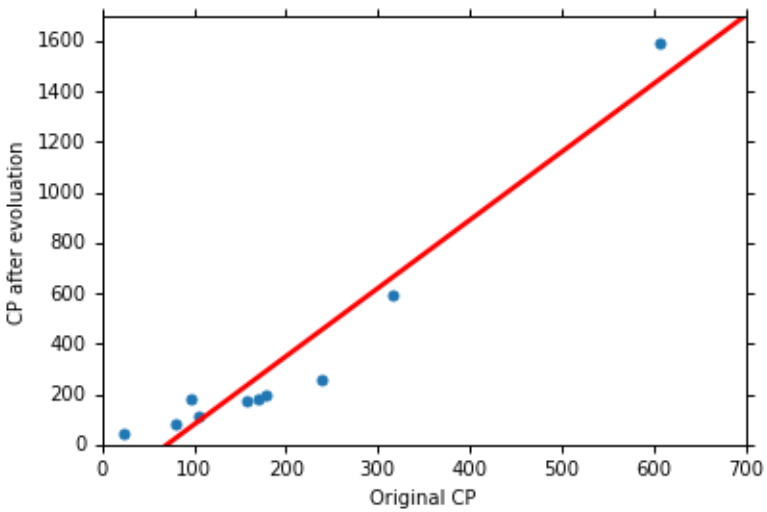
- In different universes, we use the same model, but obtain different f^*

Universe 123



$y = b + w \cdot x_{cp}$

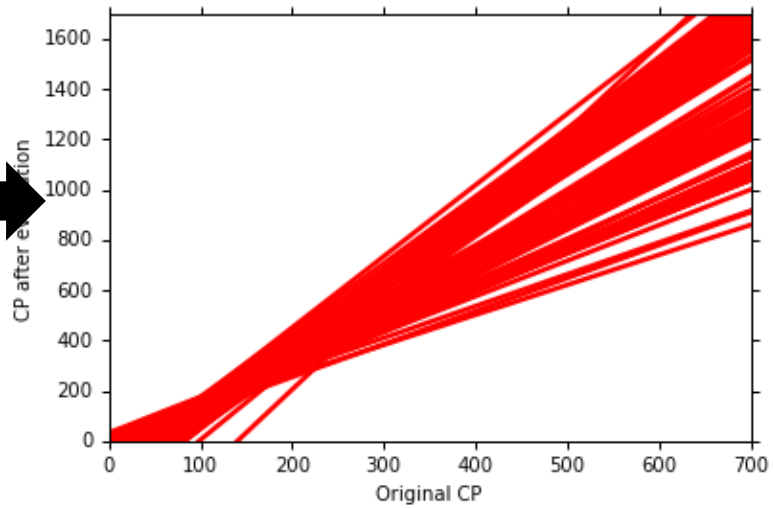
Universe 345

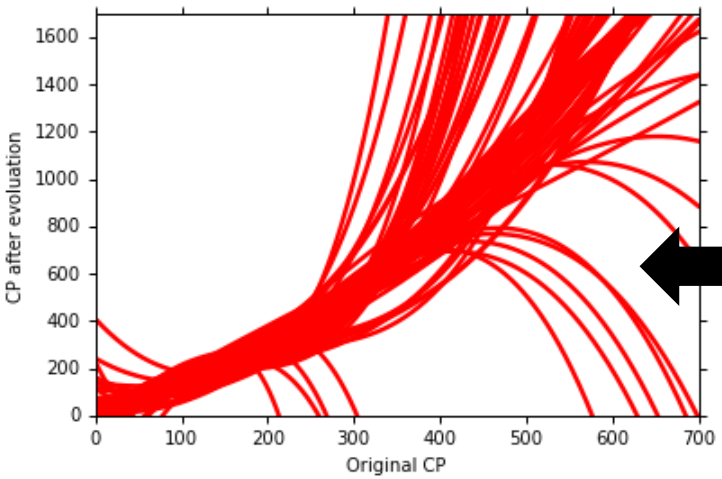


$y = b + w \cdot x_{cp}$

f^* in 100 Universes

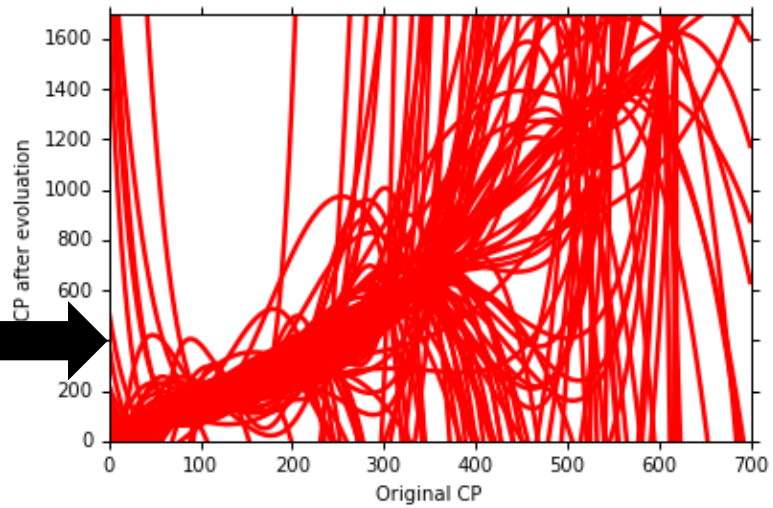
$y = b + w \cdot x_{cp}$





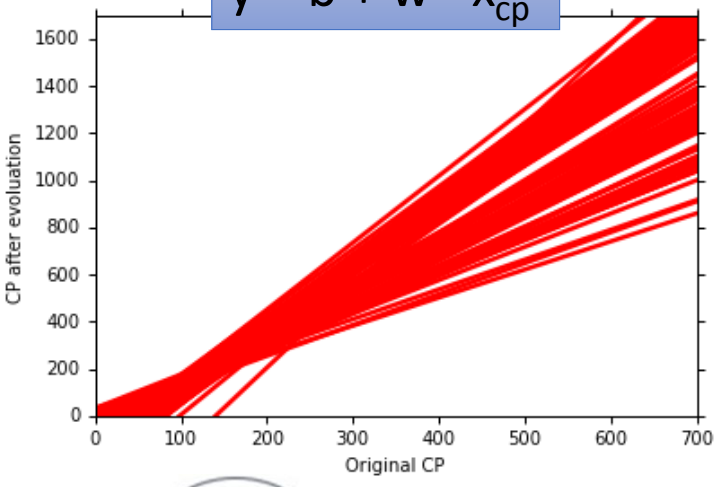
$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3$

$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4 + w_5 \cdot (x_{cp})^5$



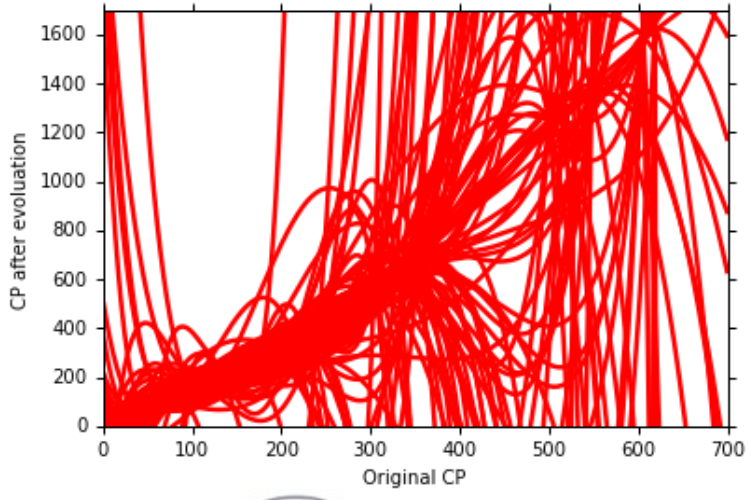
Variance

$y = b + w \cdot x_{cp}$



Small Variance

$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4 + w_5 \cdot (x_{cp})^5$



Large Variance

Simpler model is less influenced by the sampled data

Consider the extreme case f(x) = c

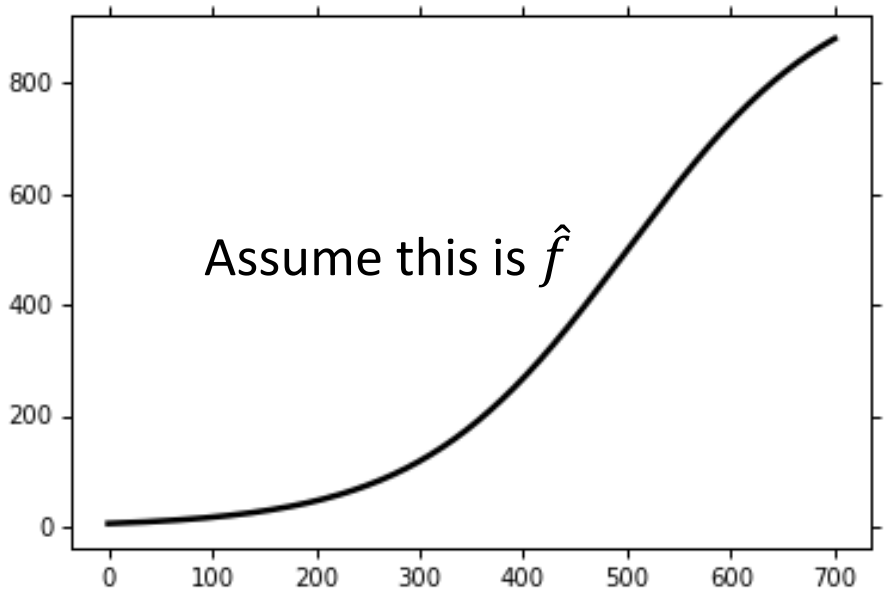
Bias

$$E[f^*] = \bar{f}$$

- Bias: If we average all the f^*, is it close to $\hat{f}$

Large Bias

Small Bias

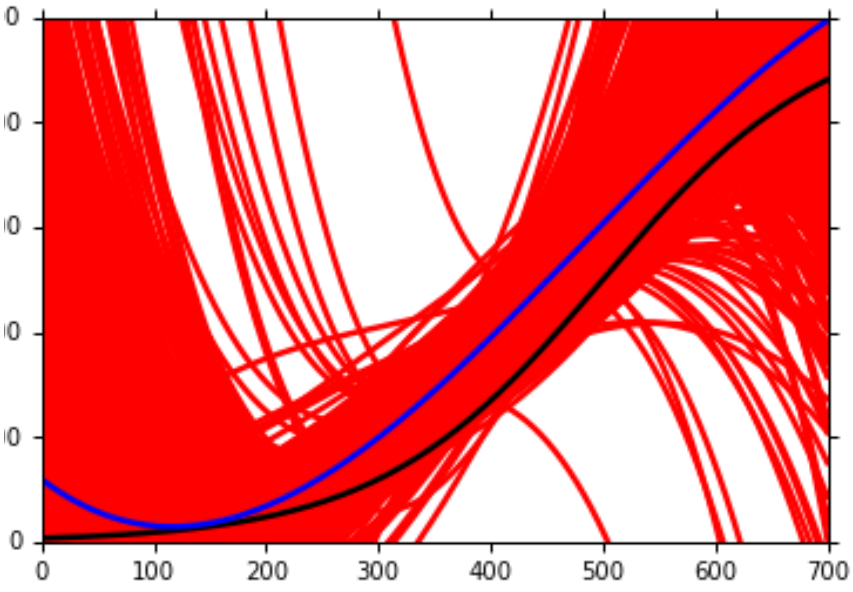


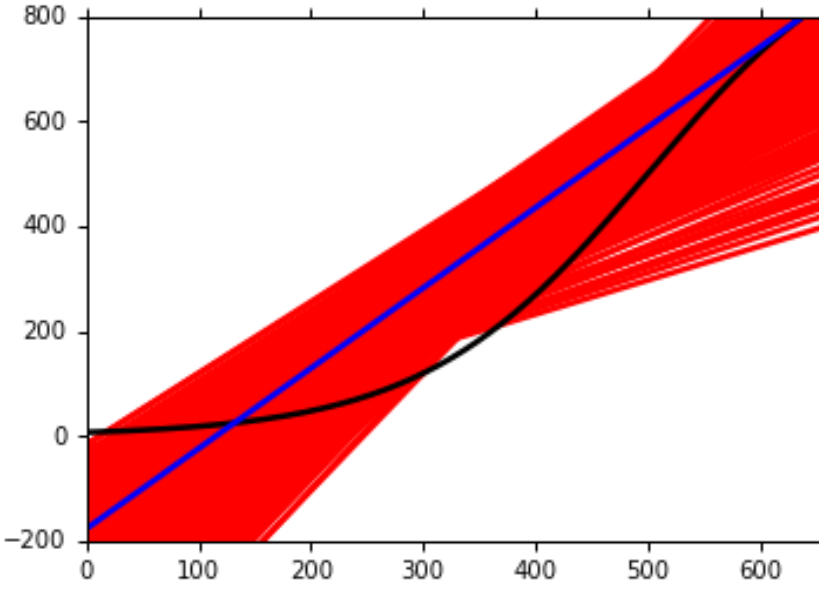
Black curve: the true function $\hat{f}$

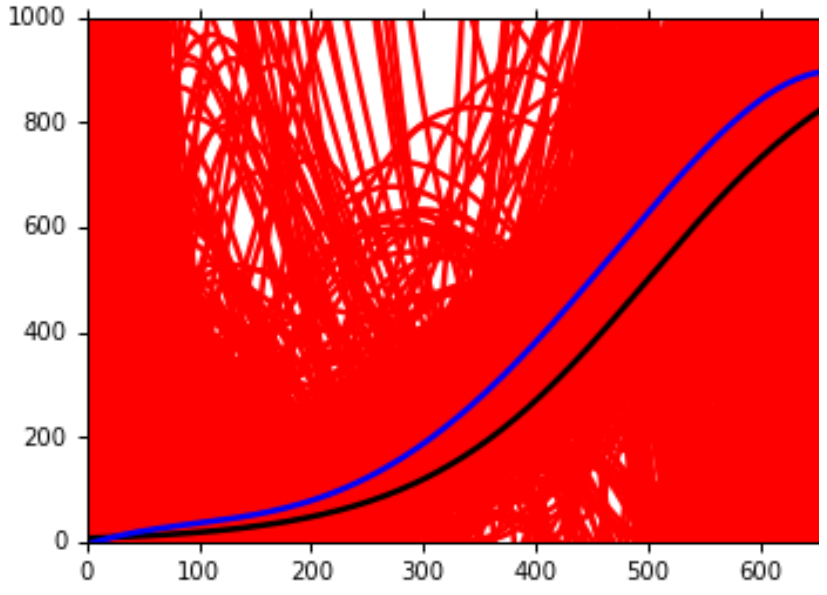
Red curves: 5000 f^*

Blue curve: the average of 5000 f^*

$= \bar{f}$



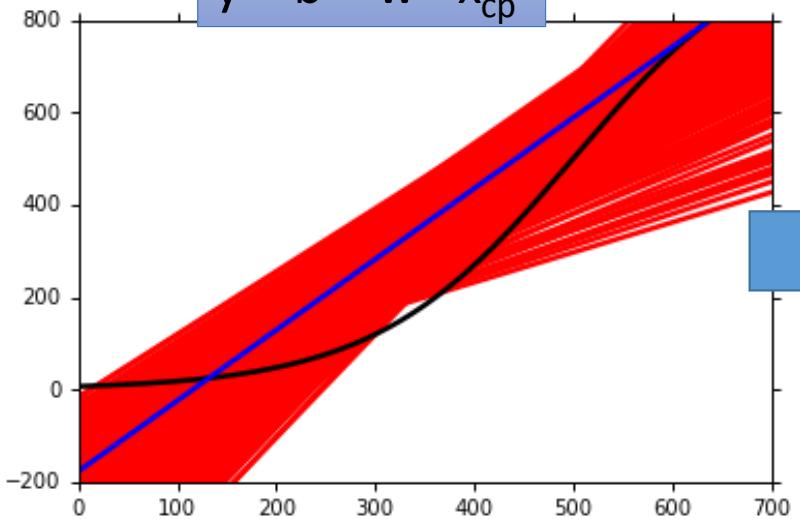


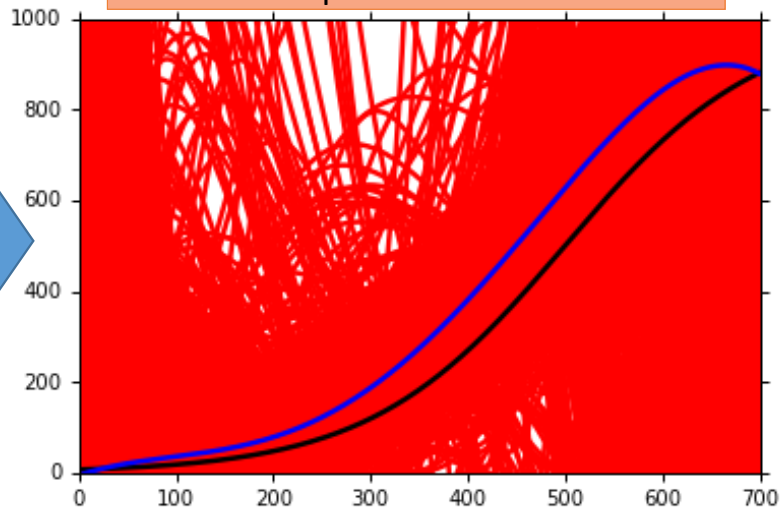


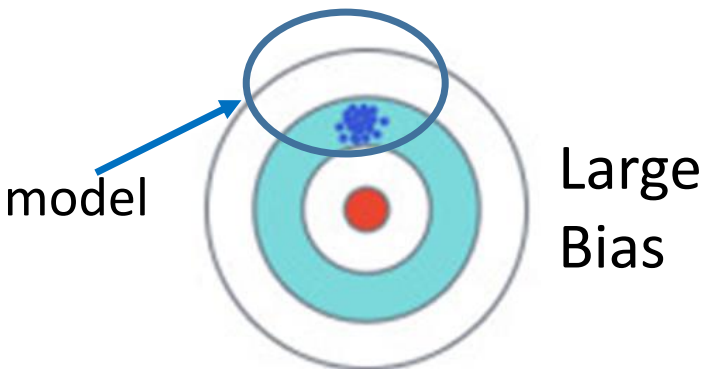
Bias

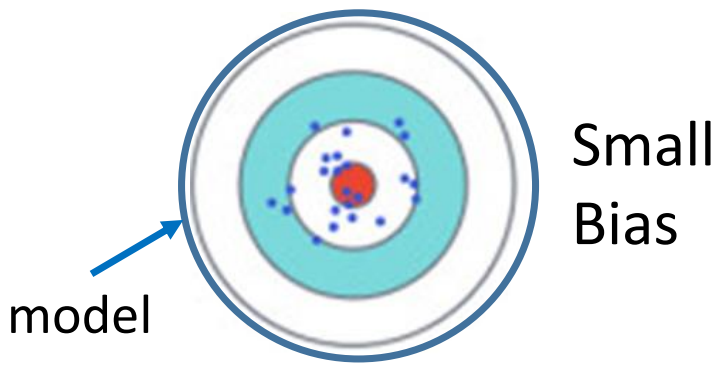
$y = b + w \cdot x_{cp}$

$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4 + w_5 \cdot (x_{cp})^5$

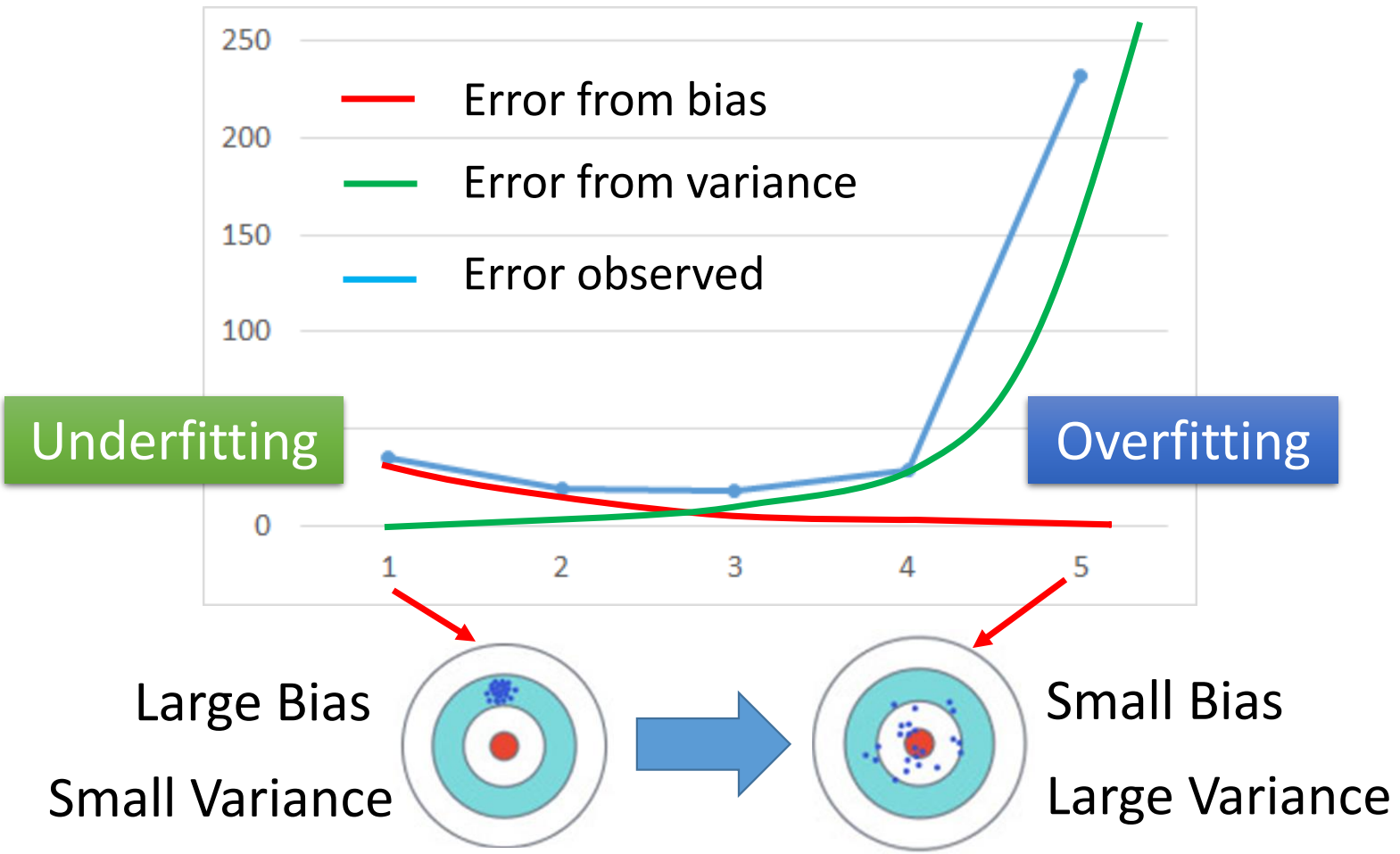






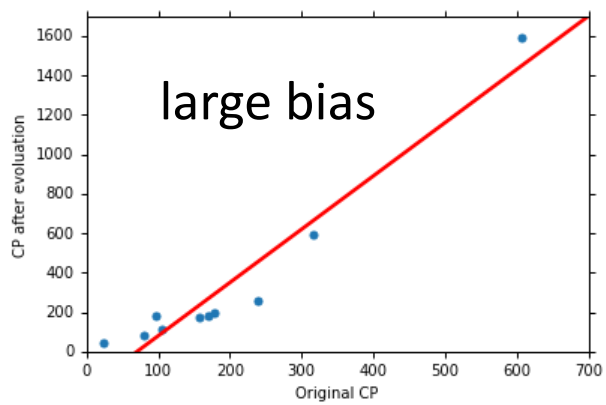


Bias v.s. Variance



What to do with large bias?

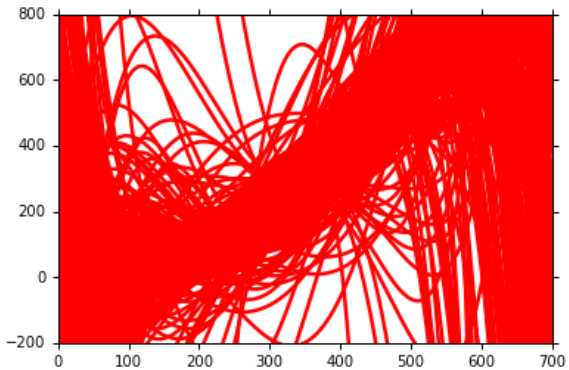
- Diagnosis:
  - If your model cannot even fit the training examples, then you have large bias **Underfitting**
  - If you can fit the training data, but large error on testing data, then you probably have large variance **Overfitting**
- For bias, redesign your model:
  - Add more features as input
  - A more complex model



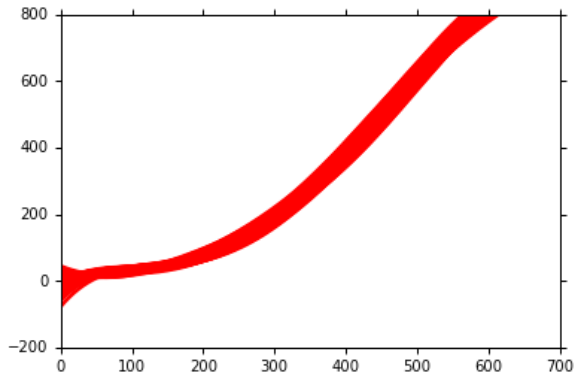
What to do with large variance?

- More data

Very effective, but not always practical

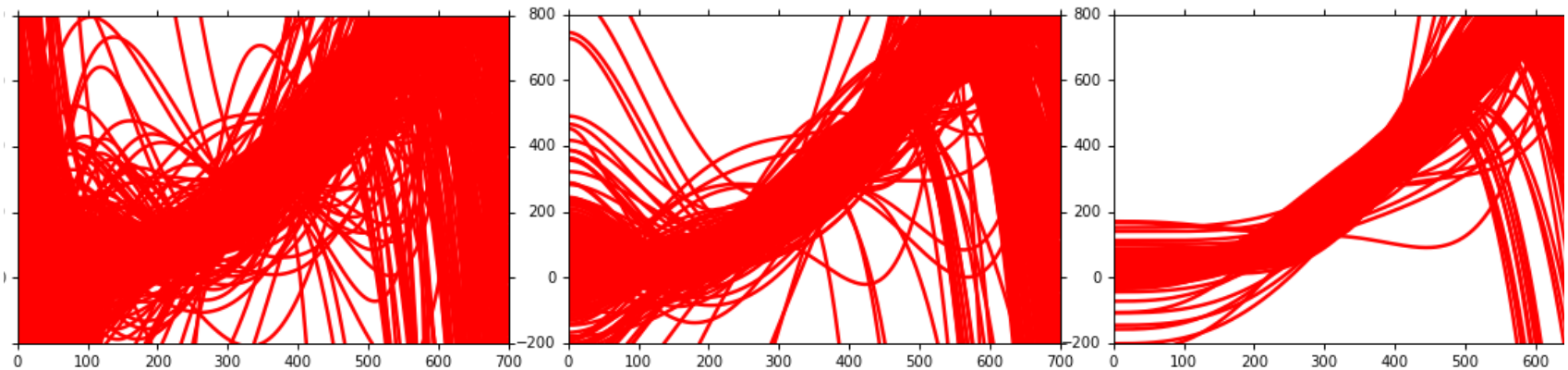


10 examples



100 examples

- Regularization



Regularization in Machine Learning

- **L1 Regularization (Lasso):**
  - Adds a penalty equal to the absolute value of the magnitude of coefficients.
  - Encourages sparsity, i.e., many coefficients become zero.
  - Objective function:

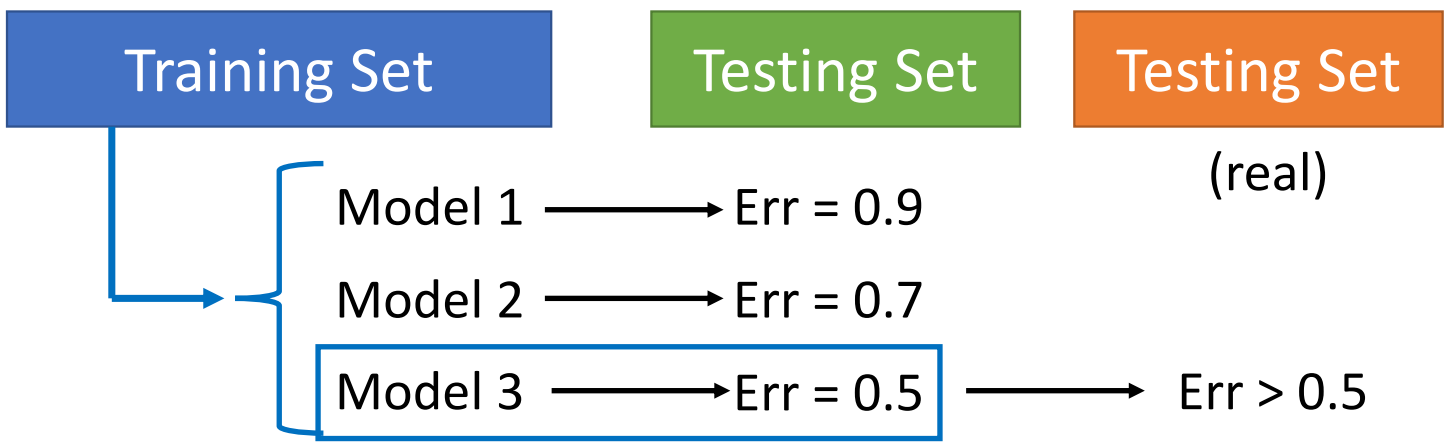
$$\min_{w} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \mathbf{w})^2 + \lambda \sum_{j=1}^{p} |w_j| \right\}$$

- **L2 Regularization (Ridge):**
  - Adds a penalty equal to the square of the magnitude of coefficients.
  - Does not encourage sparsity.
  - Objective function:

$$\min_{w} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \mathbf{w})^2 + \lambda \sum_{j=1}^{p} w_j^2 \right\}$$

Model Selection

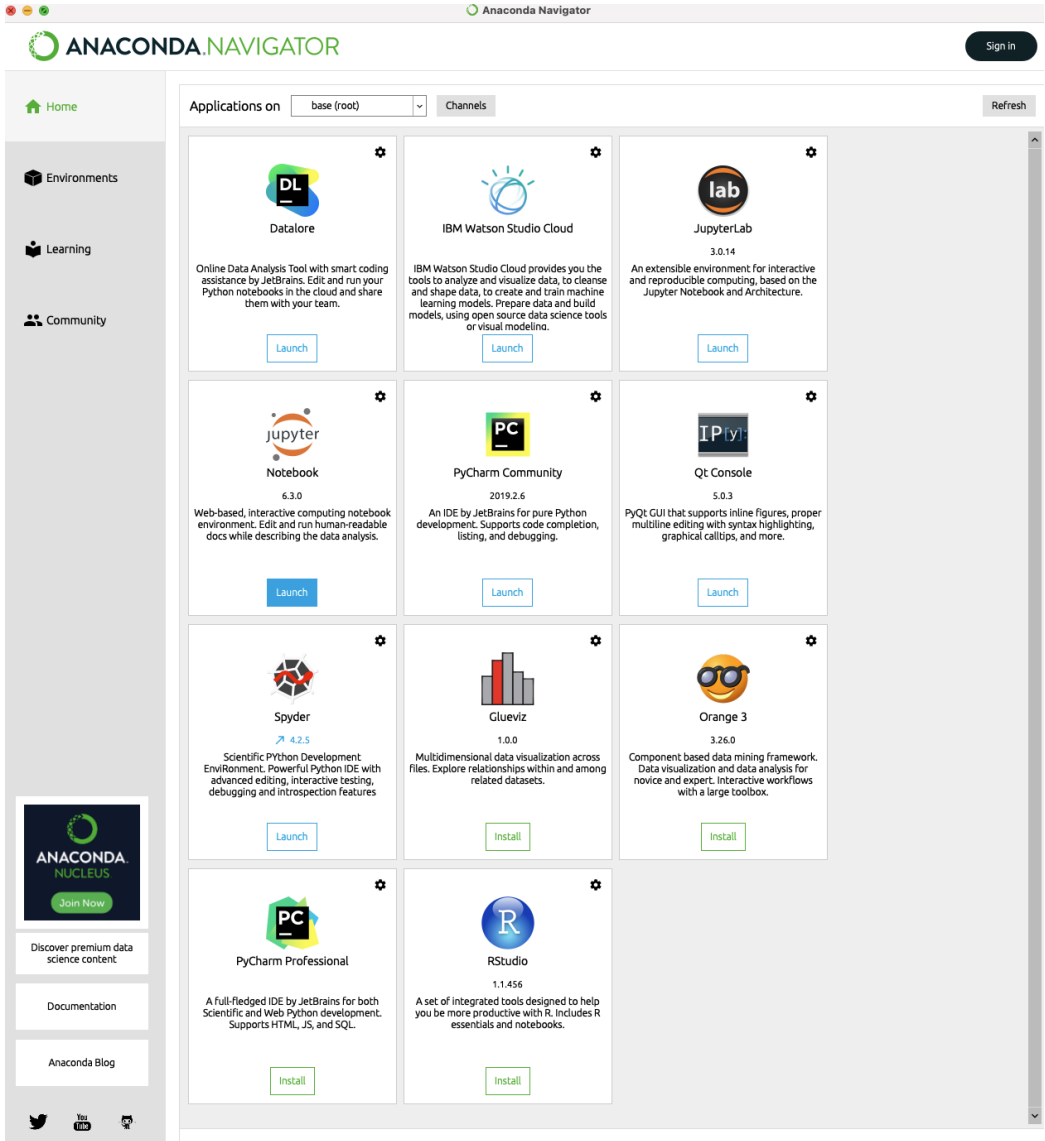
- There is usually a trade-off between bias and variance.
- Select a model that balances two kinds of error to minimize total error
- What you should NOT do:



Toolkit: Introduction to Anaconda

- Comprehensive Data Science Platform: Anaconda is an open-source distribution of Python and R, designed for scientific computing and data science.
- Includes 1,500+ Packages: Provides a vast library of data science packages, including NumPy, pandas, and scikit-learn.
- Simplifies Package Management: Offers a user-friendly way to manage packages and environments with conda.

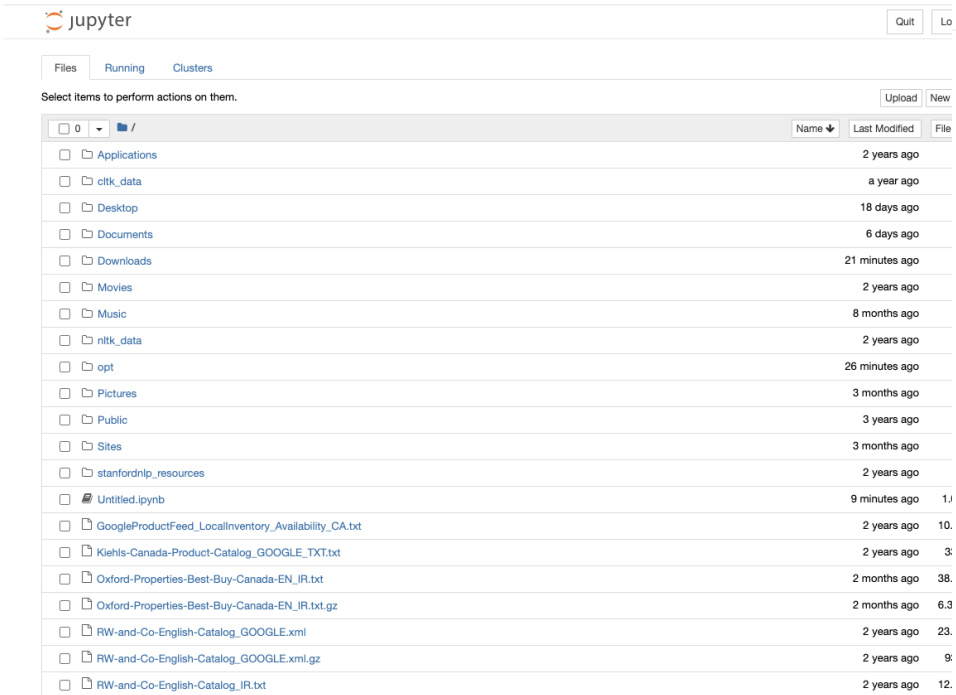
Navigating Anaconda Navigator

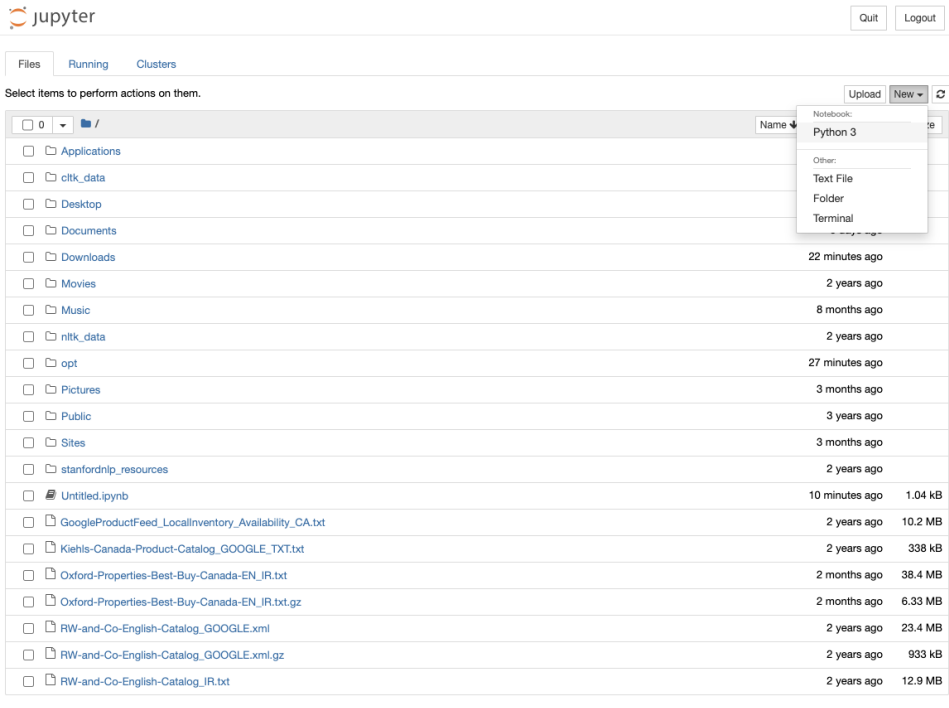


Installation

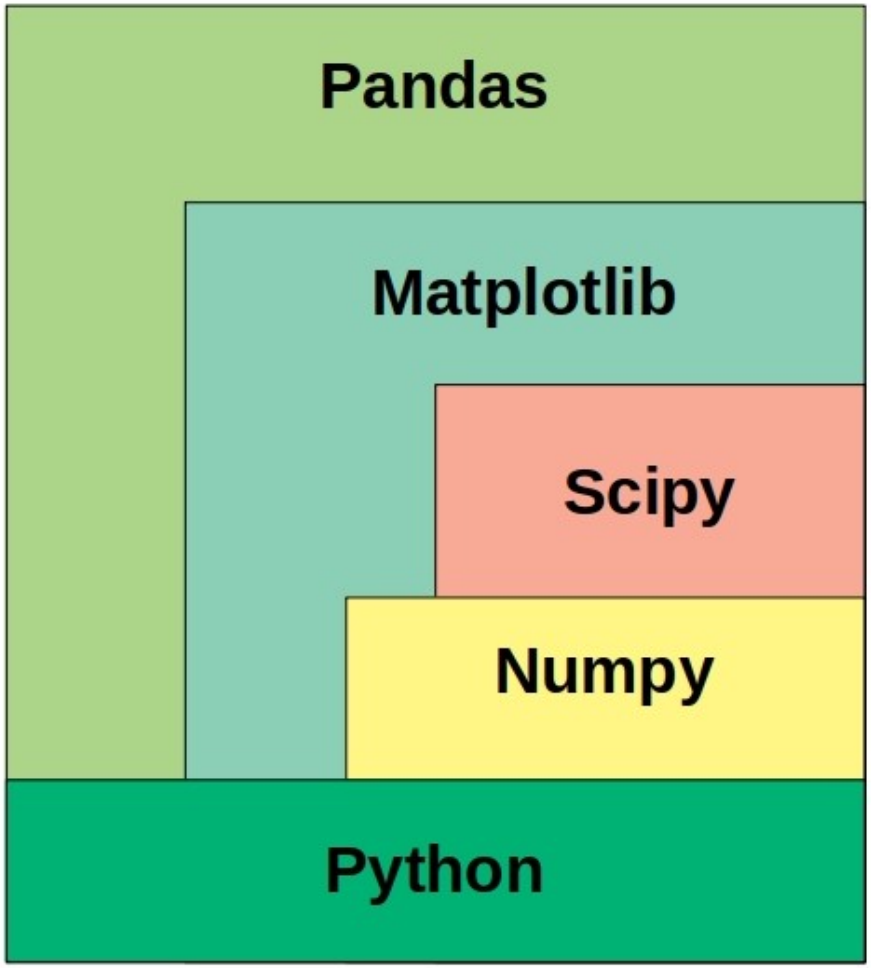
- Download the Installer: Visit the Anaconda website and download the installer for your operating system.
- Run the Installer: Follow the on-screen instructions to complete the installation.
- Verify Installation: Open a terminal or command prompt and type ‘conda –version‘ to ensure Anaconda is installed correctly.

Jupyter Notebook



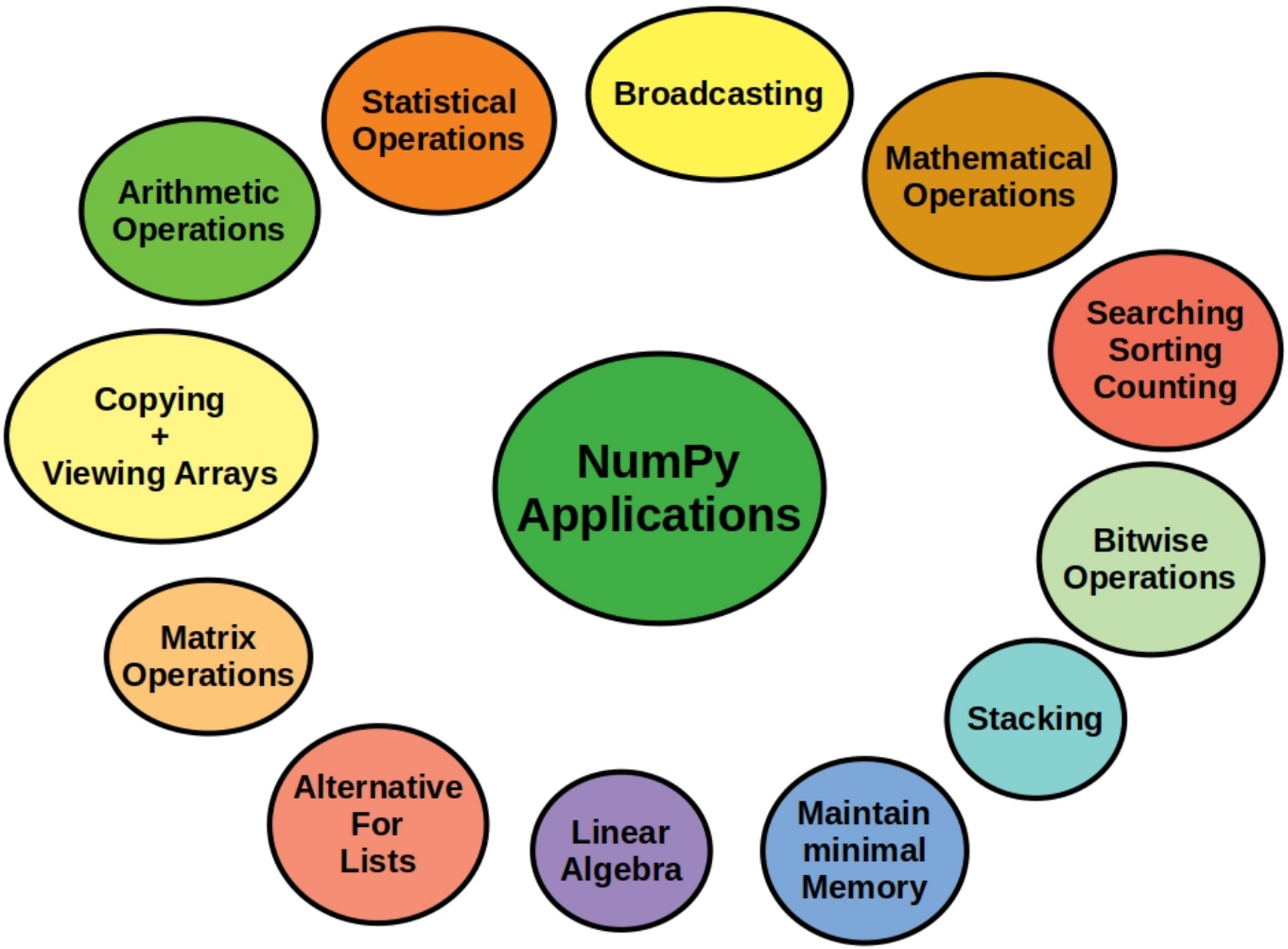


NumPy vs Pandas - Which is used When?



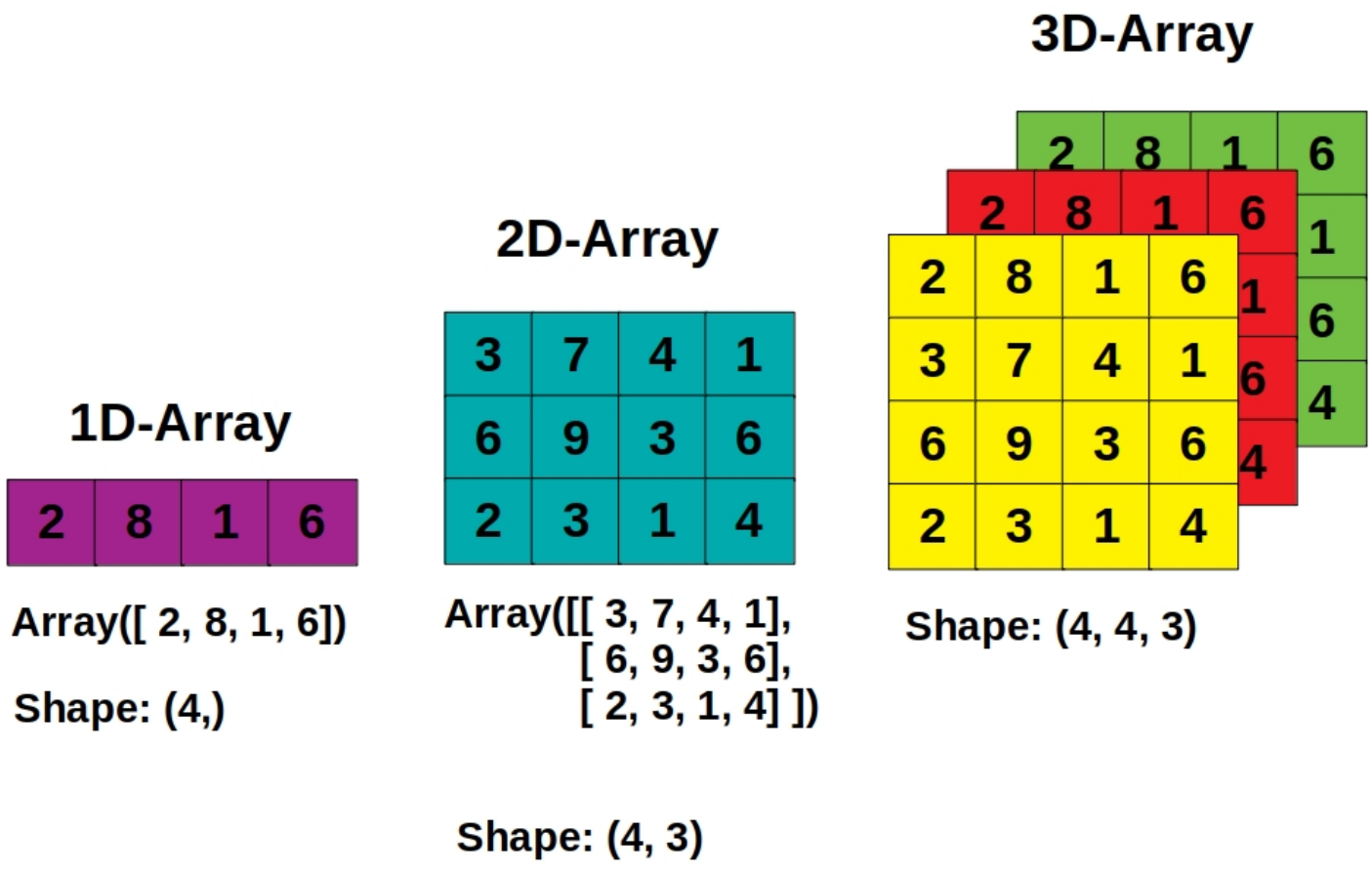
NumPy and Pandas are perhaps two of the best-known python libraries

Numpys Major Applications



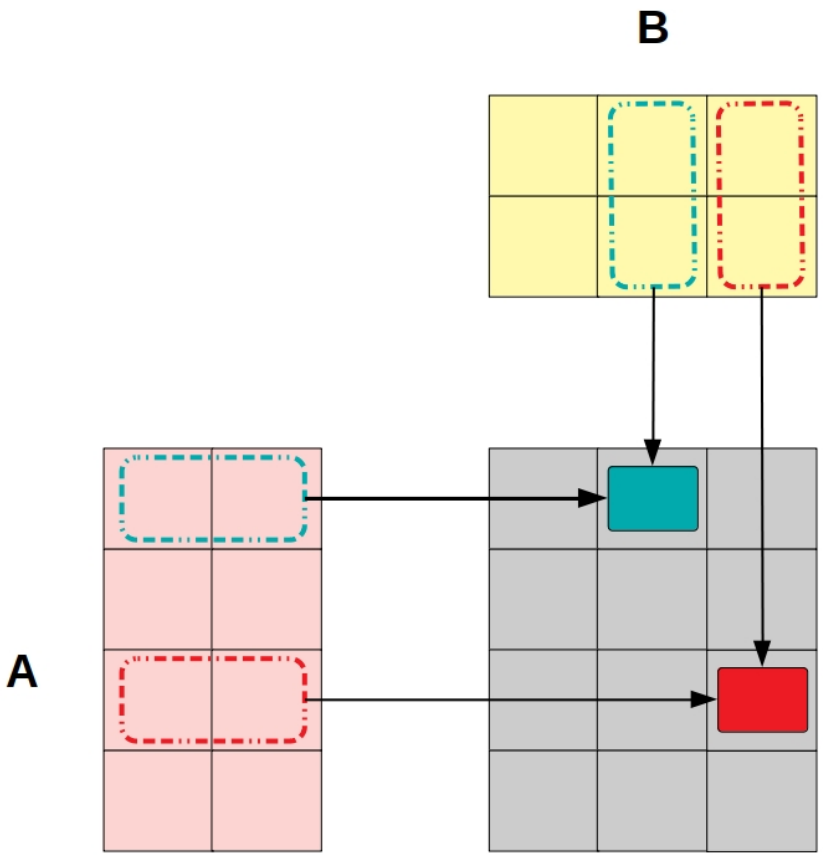
NumPy's fundamental data structure - ndarray

Allow easy handling of vectors, matrices, or large multidimensional arrays in general



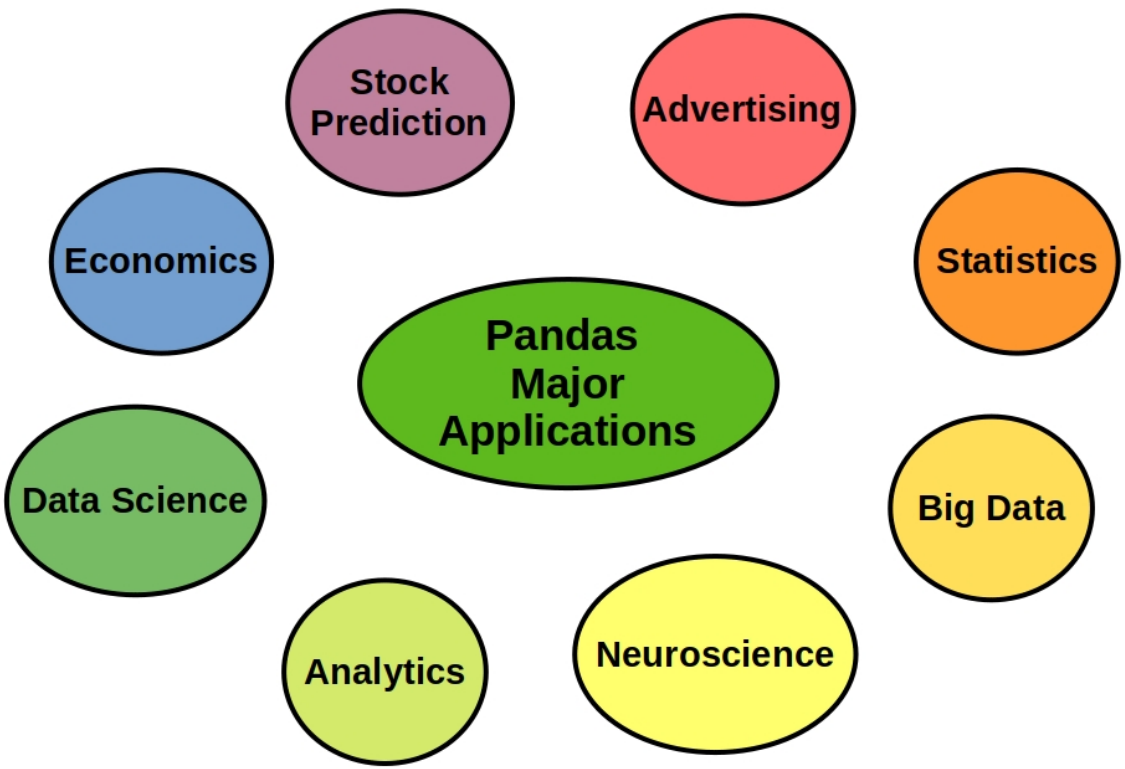
NumPys data structure is manipulable

Hold elements of the same data type and always consists of a pointer to a contiguous memory area together with the metadata



What is Pandas?

Open source library for data analysis and manipulation in Python. The name is derived from Panel Data



Pandas Features

Strength lies in the processing and analysis of tabular data and time series