

## Last session (2015-10-15)

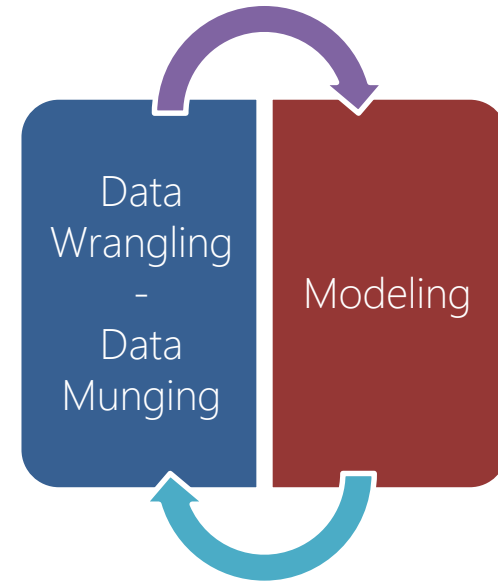
- Big Data Analysis Process
  - Data Wrangling
- Hands-on

## Today's session

- Big Data Analysis Process
  - Modeling
- Hands-on

# Big Data Analysis Process – Main Steps

- Data access
- Data pre-processing / cleaning
- Data transformation / manipulation
- Feature selection
- Feature extraction
- Feature engineering
- Model choice and training
- Model evaluation and tuning
- Model deployment



## Task Types

- Supervised Learning
  - Regression
  - Classification
- Unsupervised Learning
  - Clustering
  - Feature learning
- Reinforcement Learning
  - Control

# Supervised Learning – Teacher <-> Student

Question / Objective

Data / Examples



Learning Algorithm



Predictive Model

	A	B	C	D	E	F	G	H	I	J	K	L
1	DateTime	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11
2	1/1/2004 0.30	14613	126259	126231	484	4829	131658	136213	3132	75382	21319	86700
3	1/1/2004 1.30	14610	123113	133055	457	4596	129909	133055	2956	67368	22100	86609
4	1/1/2004 2.30	14617	119192	128608	450	4525	125717	128608	2953	64050	21376	84243
5	1/1/2004 3.30	14672	117507	125791	448	4654	124462	124791	2944	63861	21315	84435
6	1/1/2004 4.30	17064	118143	127657	484	4977	125120	127657	3221	73452	21564	80887
7	1/1/2004 5.30	17727	122228	130626	490	7330	138158	130626	3361	79969	22241	96210
8	1/1/2004 6.30	18164	126713	136743	523	7795	134526	136743	3506	80239	22937	96470
9	1/1/2004 7.30	19355	132289	142741	581	8303	140593	142741	3904	73452	23666	98218
10	1/1/2004 8.30	20234	139941	150097	622	7090	140431	150097	4254	69552	24120	102024
11	1/1/2004 9.30	18611	141950	153365	633	9516	151466	153365	4210	69888	24116	102875
12	1/1/2004 10.30	17666	145109	156373	683	9414	154523	156373	4215	66549	23718	103584
13	1/1/2004 11.30	16179	144750	156195	599	8915	153479	156195	4201	65625	22770	94846
14	1/1/2004 12.30	15106	143331	152713	519	8318	149489	152713	4080	65608	21866	93564
15	1/1/2004 13.30	14495	133330	151811	546	8127	140720	151811	3990	65877	20771	91604
16	1/1/2004 14.30	13518	127683	137770	526	6858	134460	137770	3774	65667	20222	84515
17	1/1/2004 15.30	13118	126829	136849	507	6635	133465	136849	3753	65625	20073	83058
18	1/1/2004 16.30	14130	131163	140842	560	7220	142383	140842	3920	64932	21130	86113
19	1/1/2004 17.30	14809	155436	167725	629	8658	164093	167725	4348	66129	24225	105060
20	1/1/2004 18.30	18150	157850	170321	683	9179	167029	170321	4608	65221	24627	112328
21	1/1/2004 19.30	18235	154640	169320	683	9105	165945	169320	4532	64911	24641	112727
22	1/1/2004 20.30	17805	153354	165470	637	8771	162126	165470	4272	64911	24200	113163
23	1/1/2004 21.30	16904	149676	158187	605	8222	155197	158187	3964	65352	23578	107922
24	1/1/2004 22.30	16162	137713	148592	590	7805	145018	148592	3527	73605	23826	105111
25	1/1/2004 23.30	14780	128023	138118	522	6336	134360	138118	3388	78482	22526	93204
26	1/1/2004 0.30	14155	121812	133206	481	5903	129356	133206	3003	76545	20544	84643
27	1/1/2004 1.30	14038	122307	133170	438	5678	127965	133170	2847	80830	20121	82206
28	1/1/2004 2.30	14019	122181	133133	447	5719	127900	133133	2852	80830	19564	79742
29	1/1/2004 3.30	14469	121402	133352	439	5735	128137	133352	2953	81165	19643	78462
30	1/1/2004 4.30	14920	125753	135687	488	5961	133714	135687	3036	80819	20647	80051
31	1/1/2004 5.30	16072	136545	146333	495	6400	143025	146333	3252	79882	22066	84517
32	1/1/2004 6.30	17800	148228	159398	534	7069	155296	159398	3527	66804	24602	91493
33	1/1/2004 7.30	19089	162001	174799	549	7713	169714	174799	3774	68229	26422	97086
34	1/1/2004 8.30	19677	170452	183219	573	8200	178433	183219	3790	66801	27499	101162
35	1/1/2004 9.30	20047	175754	185603	616	8143	185717	185603	3910	67095	28473	103712
36	1/1/2004 10.30	19770	177483	185054	596	7996	185441	185054	3900	66586	28774	103706
37	1/1/2004 11.30	18564	175004	188623	556	7648	182452	188623	3926	67730	29102	105206
38	1/1/2004 12.30	17400	168100	178100	500	7016	174000	178100	3810	67730	29102	105206

... containing answers to the question



... answers the question on new data

# Unsupervised Learning - Explorer

Question / Objective

Data / Examples

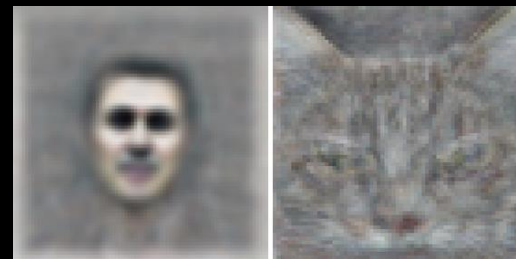


Learning Algorithm



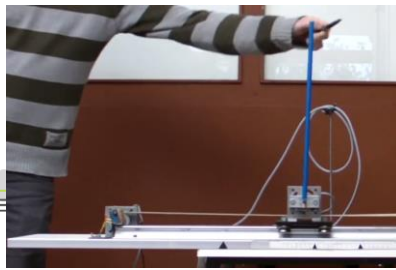
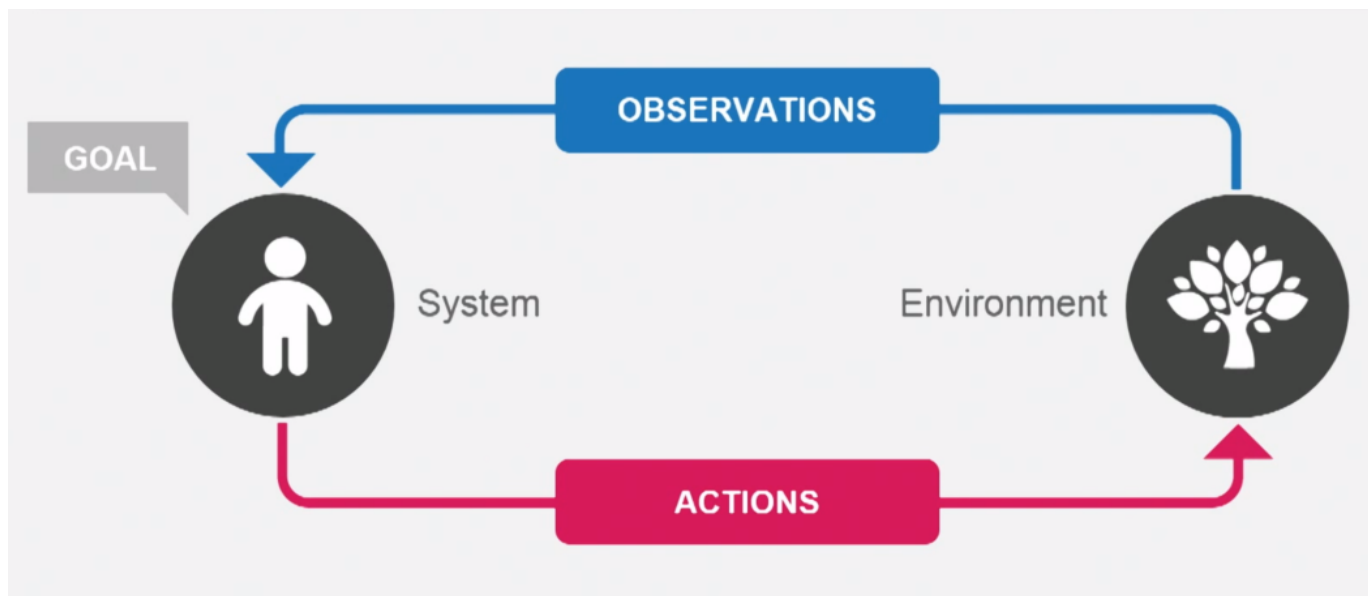
Structure Model

	A	B	C	D	E	F	G	H	I	J	K	L
1	DateTime	21	22	23	24	25	26	27	28	29	30	31
2	1/2/2004 0:00	14815	126258	136231	484	4429	133088	136231	3124	75243	21899	90700
3	1/2/2004 1:00	14846	121112	133051	417	4596	129009	133051	2916	67468	21100	86699
4	1/2/2004 2:00	14517	138182	128608	450	4525	125717	128608	2913	64050	21176	86419
5	1/2/2004 3:00	14873	137507	122791	448	4614	124052	123791	2914	63861	21135	86485
6	1/2/2004 4:00	17064	138143	127802	448	4977	126120	127802	3121	74652	21164	86827
7	1/2/2004 5:00	17727	131218	130805	490	7130	128158	130805	3161	79989	22241	90119
8	1/2/2004 6:00	18174	126781	134743	515	7795	134826	134743	3506	80075	22937	90570
9	1/2/2004 7:00	18411	141900	125165	431	9516	151446	125165	4210	69008	24126	102475
10	1/2/2004 8:00	19134	139941	130997	612	9130	149131	130997	4114	69152	24110	102024
11	1/2/2004 9:00	18411	141900	125165	431	9516	151446	125165	4210	69008	24126	102475
12	1/2/2004 10:00	17666	141009	130373	611	9414	134523	130373	4225	66549	23718	101544
13	1/2/2004 11:00	16374	144758	136195	599	8911	151673	136195	4203	65625	22770	98486
14	1/2/2004 12:00	14231	142173	137173	519	818	148489	137173	4000	61004	21864	87044
15	1/2/2004 13:00	14455	131088	143618	546	7432	140720	143618	3950	61877	20771	88504
16	1/2/2004 14:00	13118	127683	137770	526	6818	134040	137770	3774	61067	20122	86515
17	1/2/2004 15:00	13118	127683	137770	526	6818	134040	137770	3774	61067	20122	86515
18	1/2/2004 16:00	14130	131053	145842	560	7220	142183	145842	3903	64912	21130	88219
19	1/2/2004 17:00	14809	134548	147715	619	8018	146293	147715	4348	64129	20425	100960
20	1/2/2004 18:00	18130	137850	137811	693	9179	147029	137811	4608	63121	20412	111208
21	1/2/2004 19:00	18235	136840	148240	663	9105	145845	148240	4512	64911	20461	112727
22	1/2/2004 20:00	17945	133104	145470	637	8771	142126	145470	4212	64511	20300	111611
23	1/2/2004 21:00	19064	136879	138587	615	8222	131097	138587	3904	63152	21178	107622
24	1/2/2004 22:00	16182	137713	148182	550	7805	140218	148182	3127	73005	21826	101511
25	1/2/2004 23:00	14790	138012	148118	522	8186	134960	148118	3188	74642	22026	102006
26	1/2/2004 0:00	14115	128452	133206	461	9903	129104	133206	3003	76545	20564	88663
27	1/2/2004 1:00	14018	122307	133370	418	9478	127985	133370	2847	80850	20121	82206
28	1/2/2004 2:00	14019	121211	133813	447	5719	127900	133813	2812	80850	20164	78742
29	1/2/2004 3:00	14489	123402	133152	439	5785	126137	133152	2903	81105	19863	78642
30	1/2/2004 4:00	14900	125753	131687	448	5961	131714	131687	3016	80619	20647	80051
31	1/2/2004 5:00	14012	130445	147311	415	6480	144025	147311	3121	78482	21006	86117
32	1/2/2004 6:00	17800	148218	139188	534	7069	151296	139188	3127	68604	24062	91493
33	1/2/2004 7:00	18089	148205	137499	549	7113	148154	137499	3716	68129	24421	100406
34	1/2/2004 8:00	19177	137642	148113	511	8100	139015	148113	3740	64041	21709	101161
35	1/2/2004 9:00	20047	137374	139403	614	8143	148117	139403	3910	61095	20473	101712
36	1/2/2004 10:00	19719	137499	139403	599	7998	147442	139403	3910	61095	20473	101712
37	1/2/2004 11:00	18564	137506	148829	554	7468	147052	148829	3910	61790	20102	101208
38	1/2/2004 12:00	14115	128452	133206	461	9903	129104	133206	3003	76545	20564	88663



... answers the question on new data

# Reinforcement Learning - Experimenter



# We shall focus on Supervised Learning

By far the most used type of machine learning

# Supervised Learning – Important Concepts

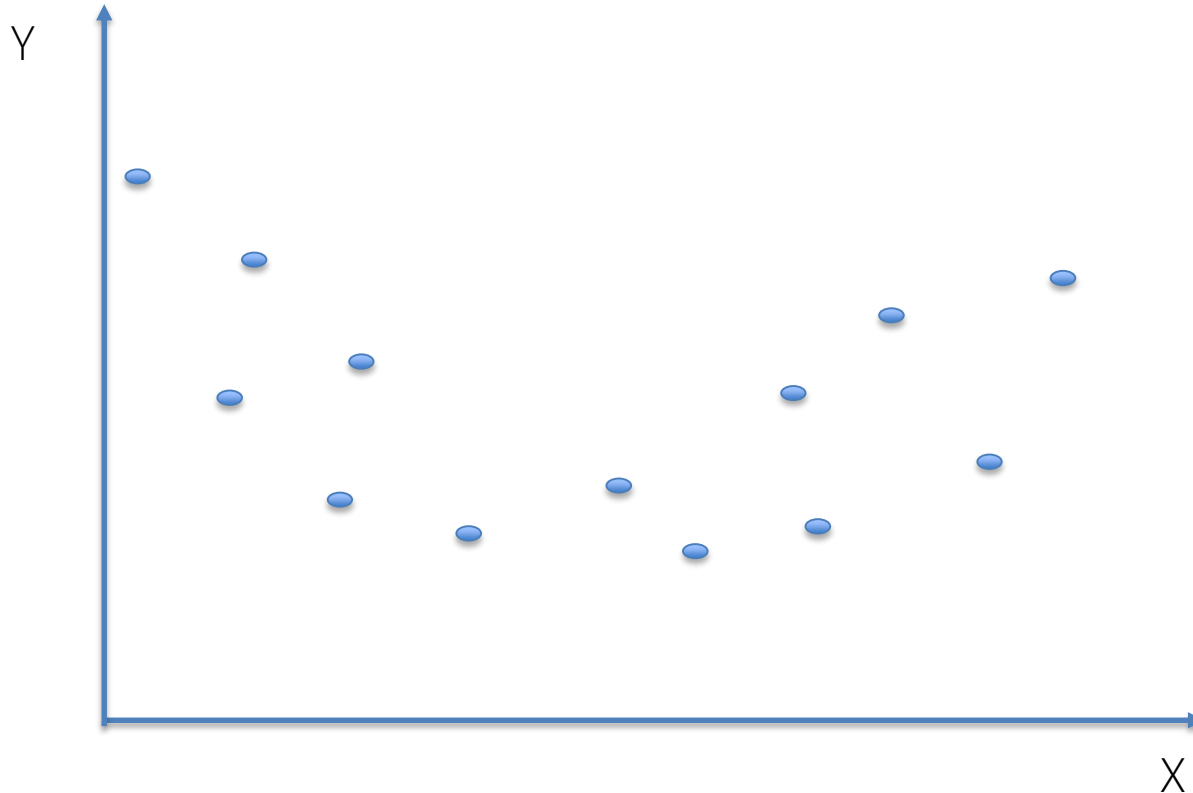
- Generalization Error
- Cross Validation
- Overfitting problem
- Class imbalance problem
- Bias & Variance tradeoff
- Ensemble modeling



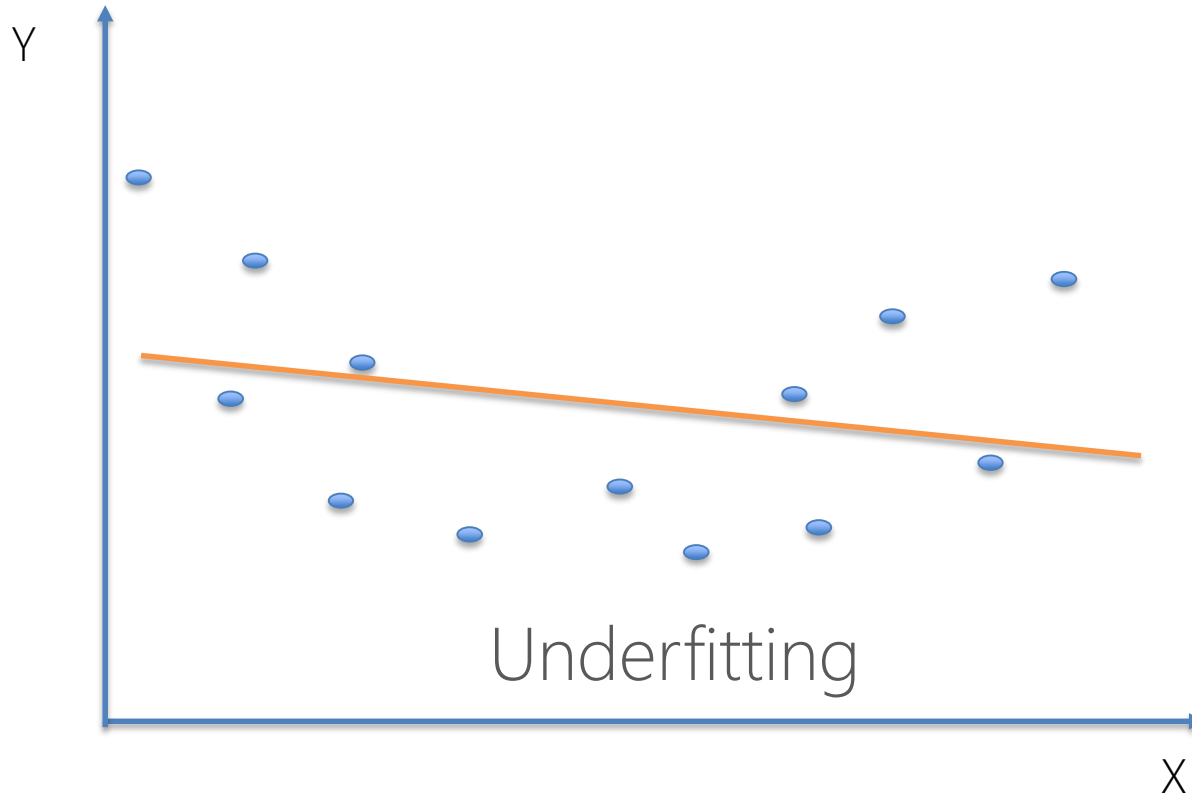
# Generalization Error

- Measures how well a machine learning model "generalizes"
- Measures how well it will perform on data it has never seen before
- Can be measured specifically for a new data set (test error)
- Can be estimated for unseen data  
(see Cross-Validation)

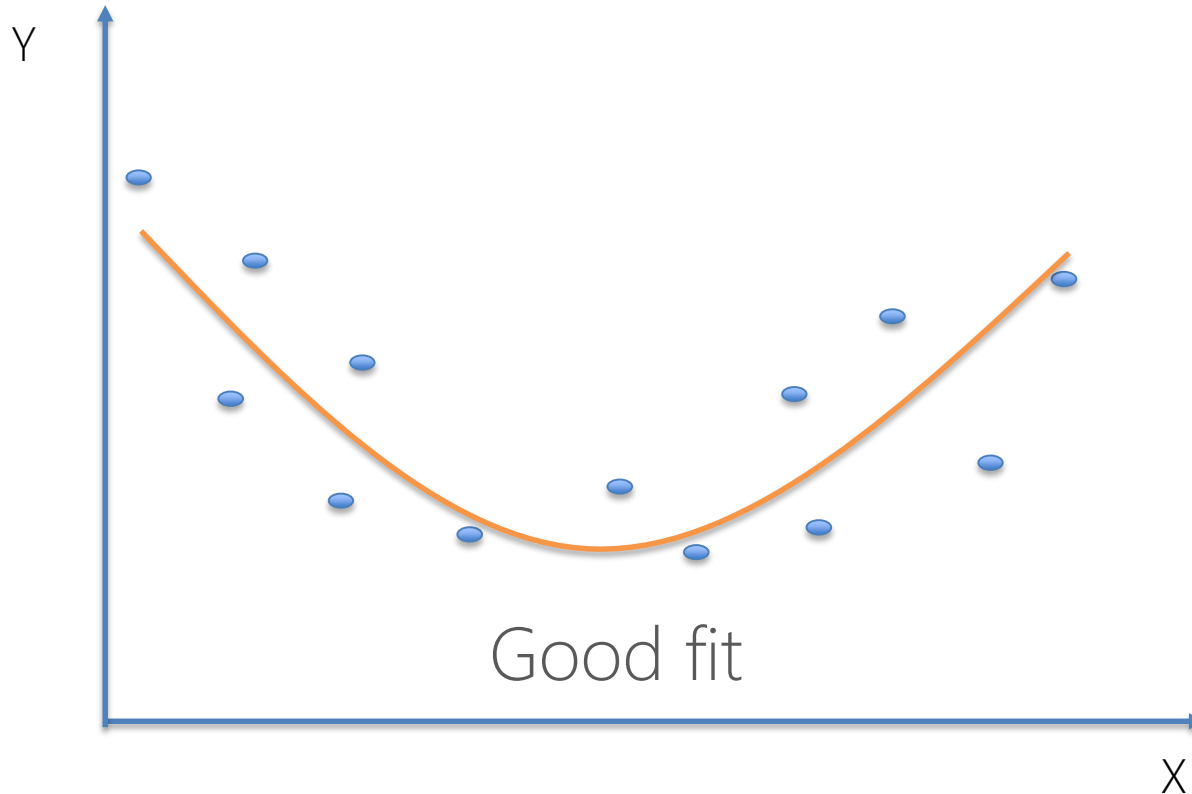
# Overfitting in Regression / Estimation



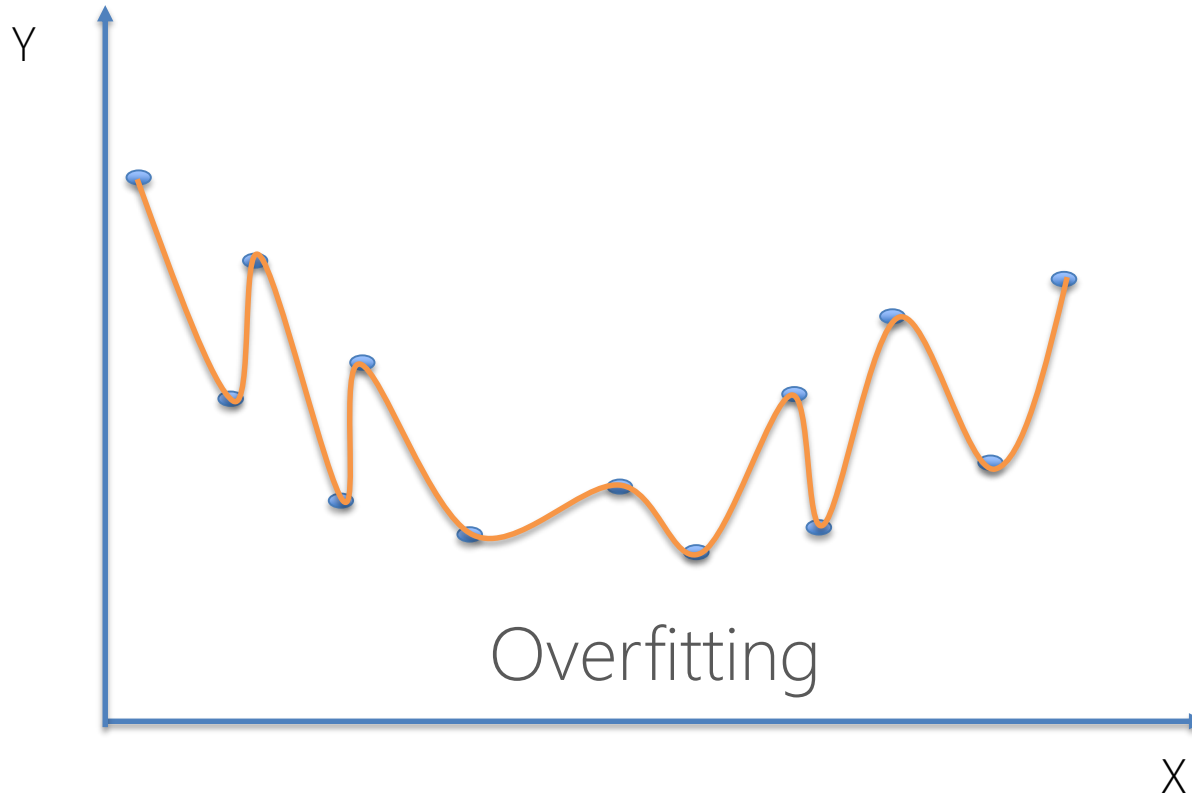
# Overfitting



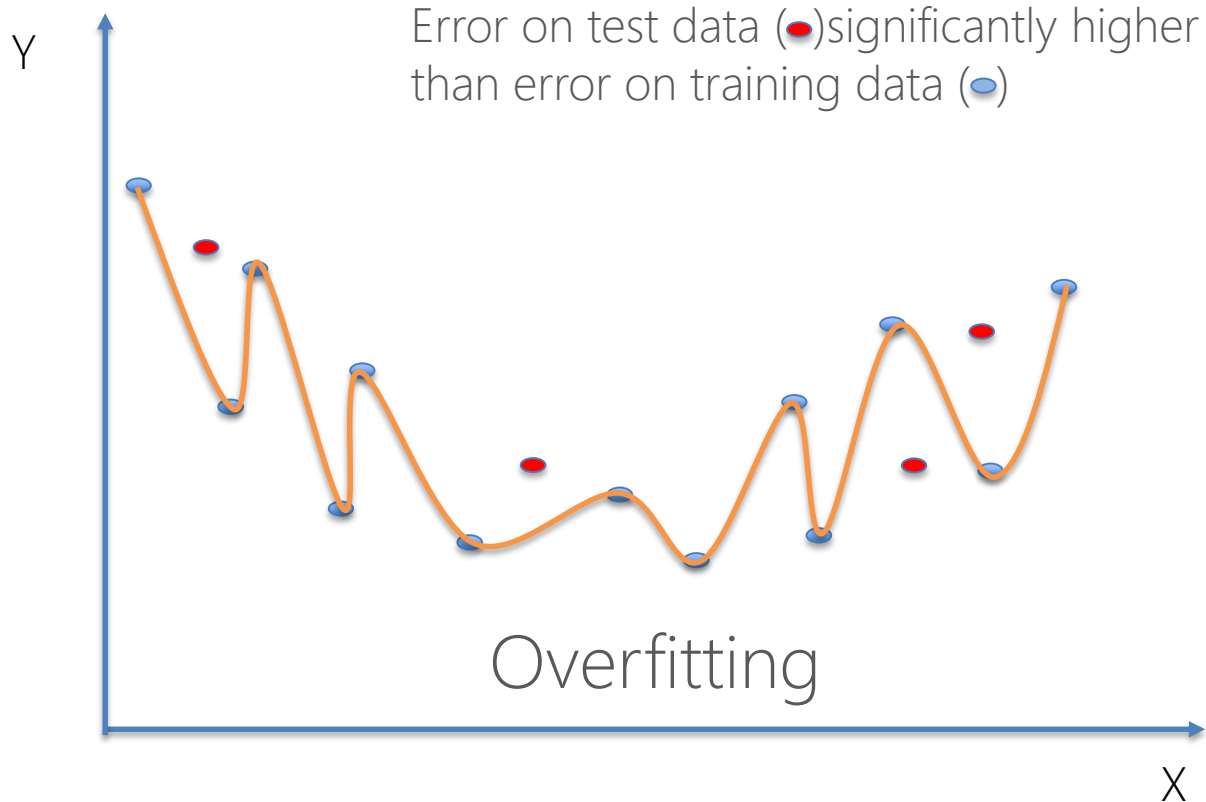
# Overfitting



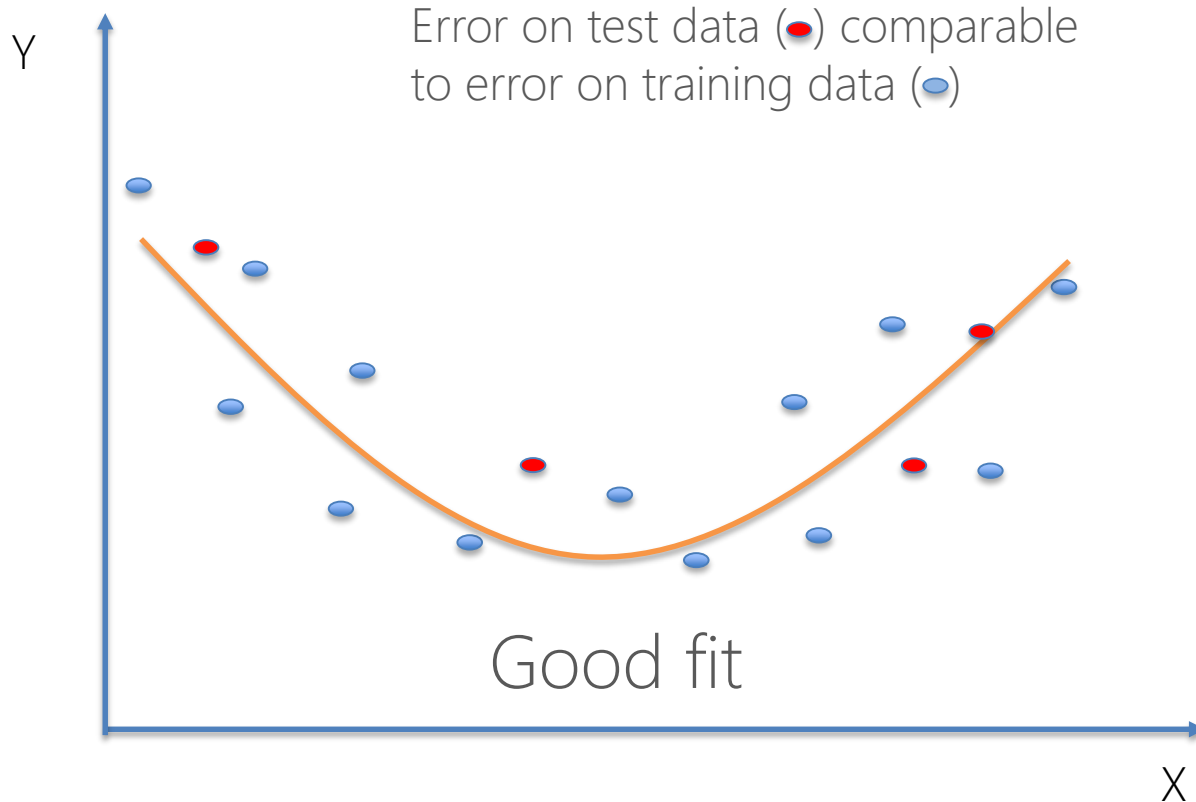
# Overfitting



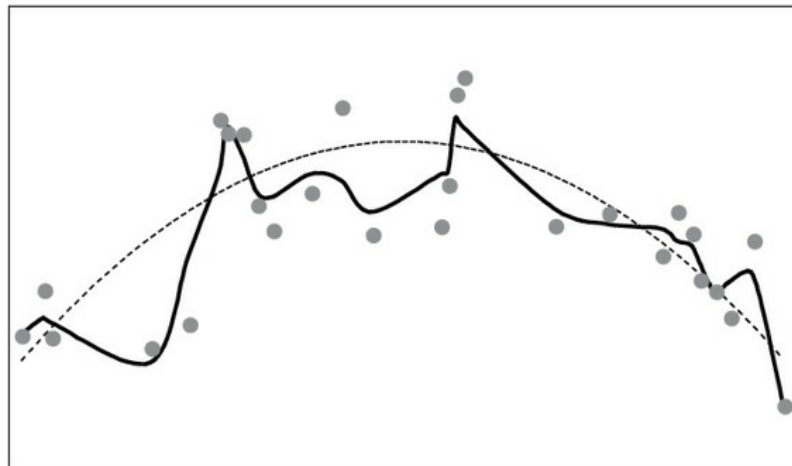
# Detecting Overfitting



# Overfitting

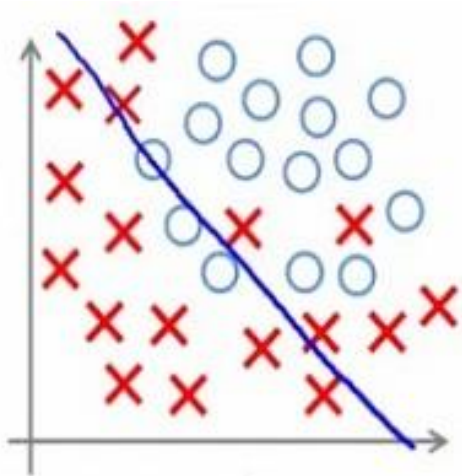


Overfitting  $\approx$  Modeling the noise in the data

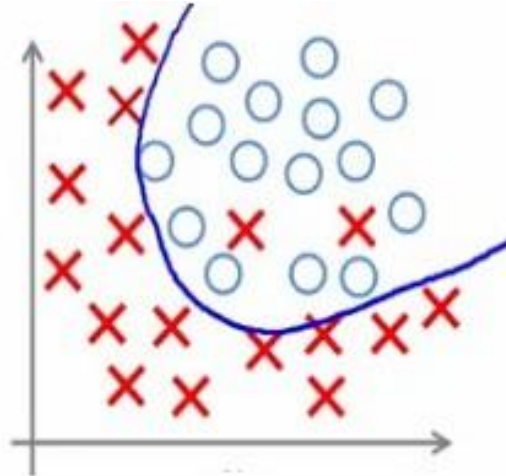




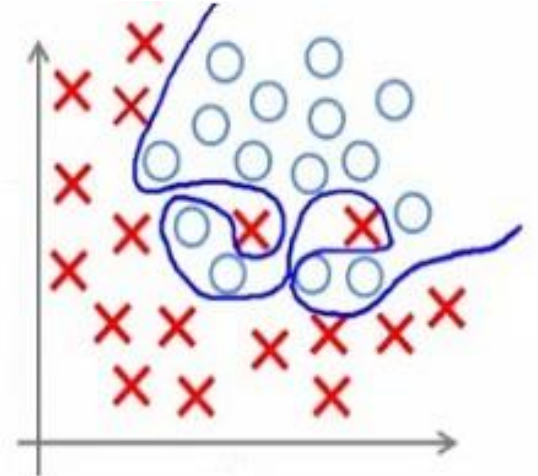
# Overfitting in Classification



**Under-fitting**



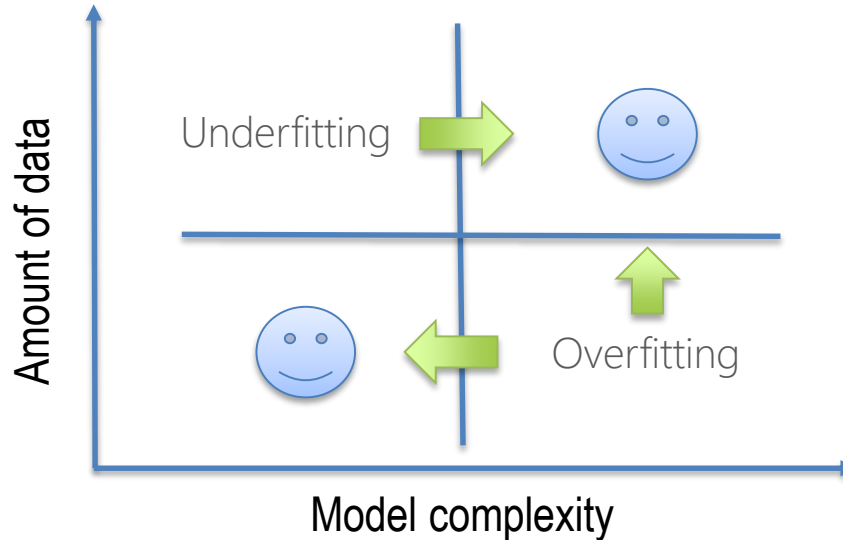
**Appropriate-fitting**



**Over-fitting**

# Reasons for Overfitting

- Too little data
- Too complex model (too many free parameters)



# Some solutions to overfitting

- Increase the amount of data
- Use a simpler model / Decrease number of parameters to tune (train)  
(see Occam's razor)
- Do not "overtune" parameters  
(e.g. Early stopping in Neural Networks)
- Integrate over many predictors  
(see Ensemble models)

# Occam's Razor



1287 – 1347

*"Among competing hypotheses, the one with the fewest assumptions should be selected"*



Select the simplest model (the one with fewer parameters) ..... that still gives adequate performance on the available data



**“Simplicity  
is the  
ultimate  
sophistication.”**

— Leonardo da Vinci

# THE KISS PRINCIPLE

# KISS



# THE KISS PRINCIPLE

# KEEP IT SIMPLE, STUPID

The KISS design principle states that most systems work best if they are kept simple rather than made complicated

Originated at Lockheed "Skunk Works" in the '60

U-2, Blackbird, Nighthawk, Raptor



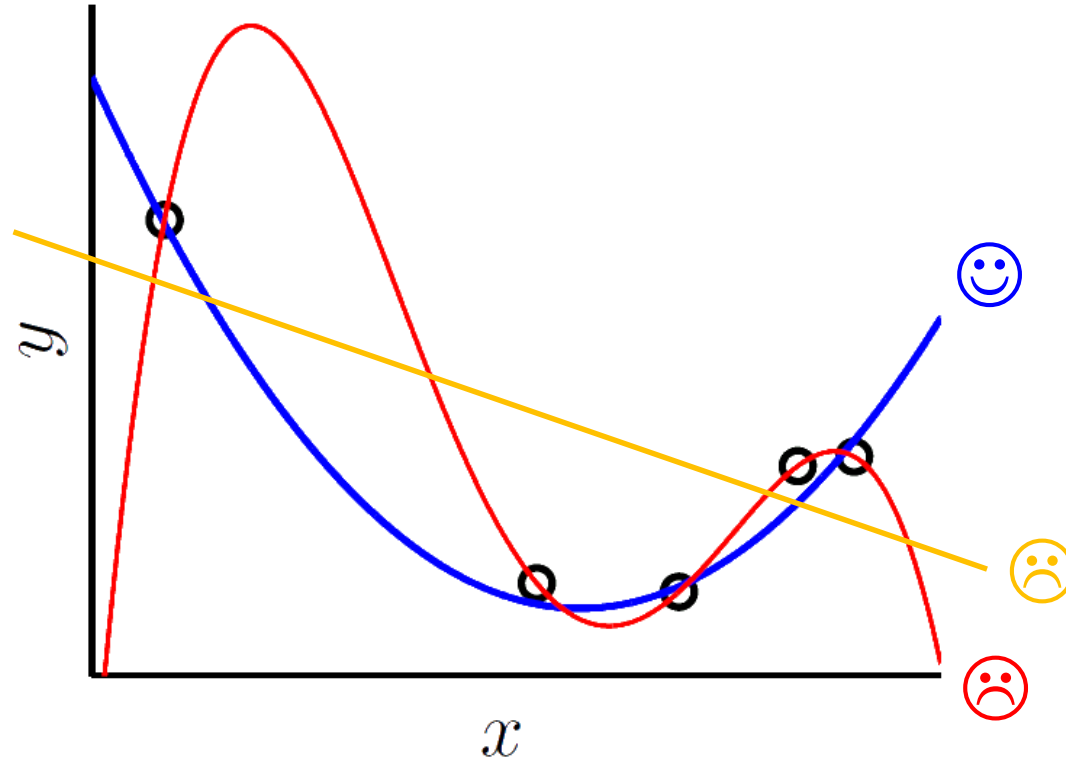
*“Everything should be made as  
simple as possible, but not simpler.”*

– Albert Einstein





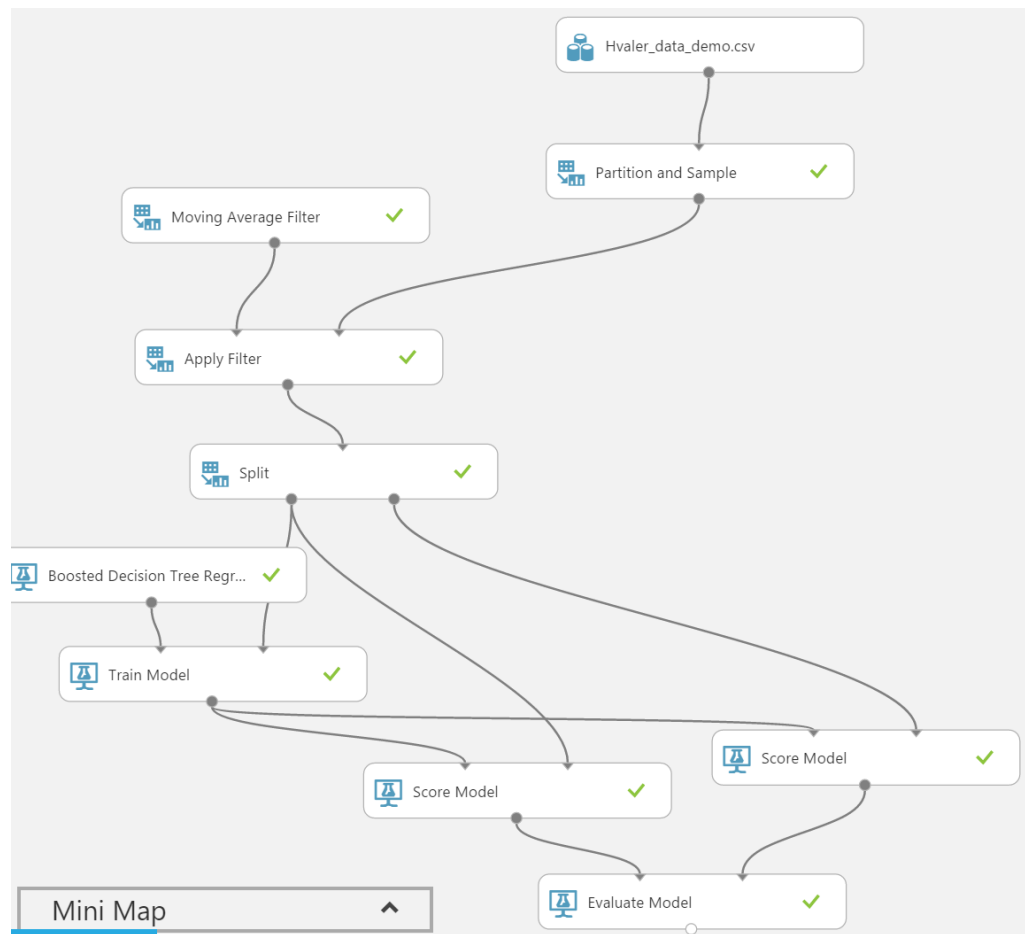
# Reduce overfitting with Occam's razor



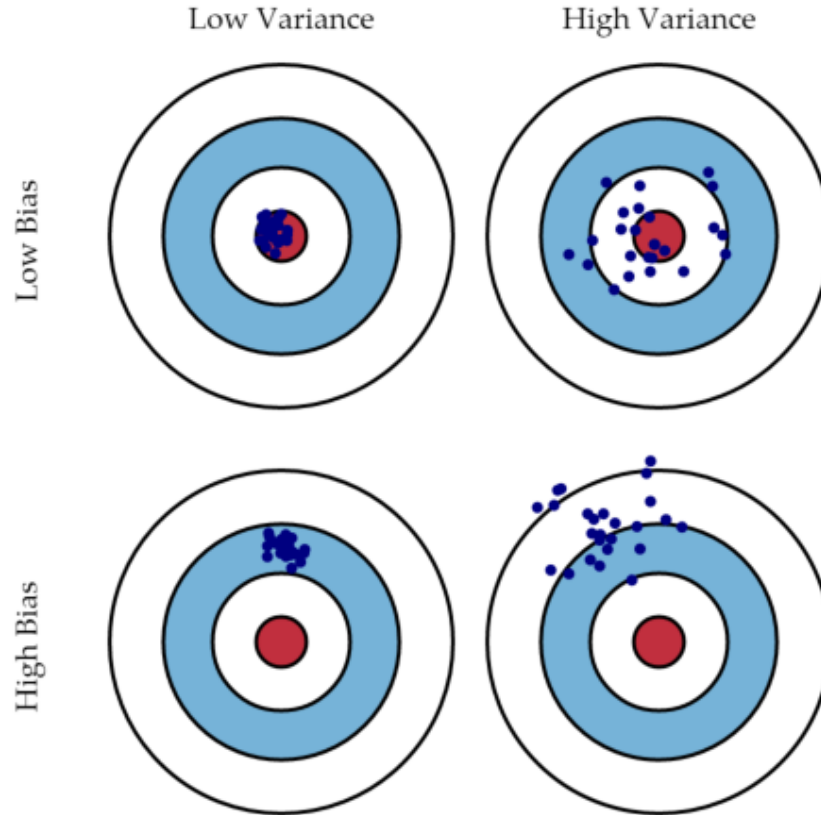
# Regularization

- Attempts to automatically implement Occam's razor
- Penalize complex models
  - Define a "penalty function" to quantify complexity of the model
    - E.g. number of free parameters
  - Since most of the training algorithms are at heart optimization problems where the model error is minimized, one adds a "complexity penalty term" and minimize the whole expression together
    - E.g. minimize error and number of free parameters at the same time

# Overfitting – Hands on



# Bias & Variance

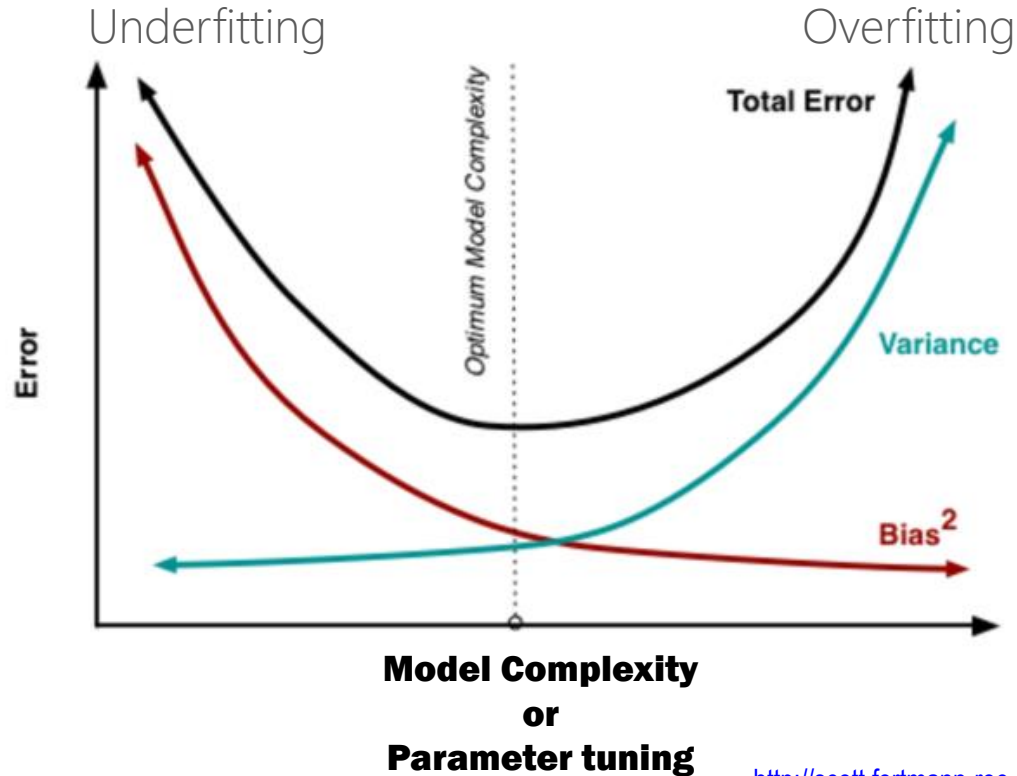


<http://scott.fortmann-roe.com/docs/BiasVariance.html>

Underfitting  $\approx$  Not modeling the data  $\approx$  High Bias

Overfitting  $\approx$  Modeling the noise  $\approx$  High Variance

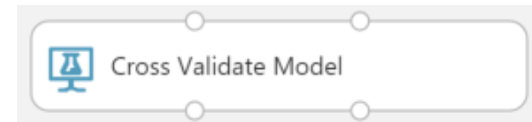
# Bias & Variance



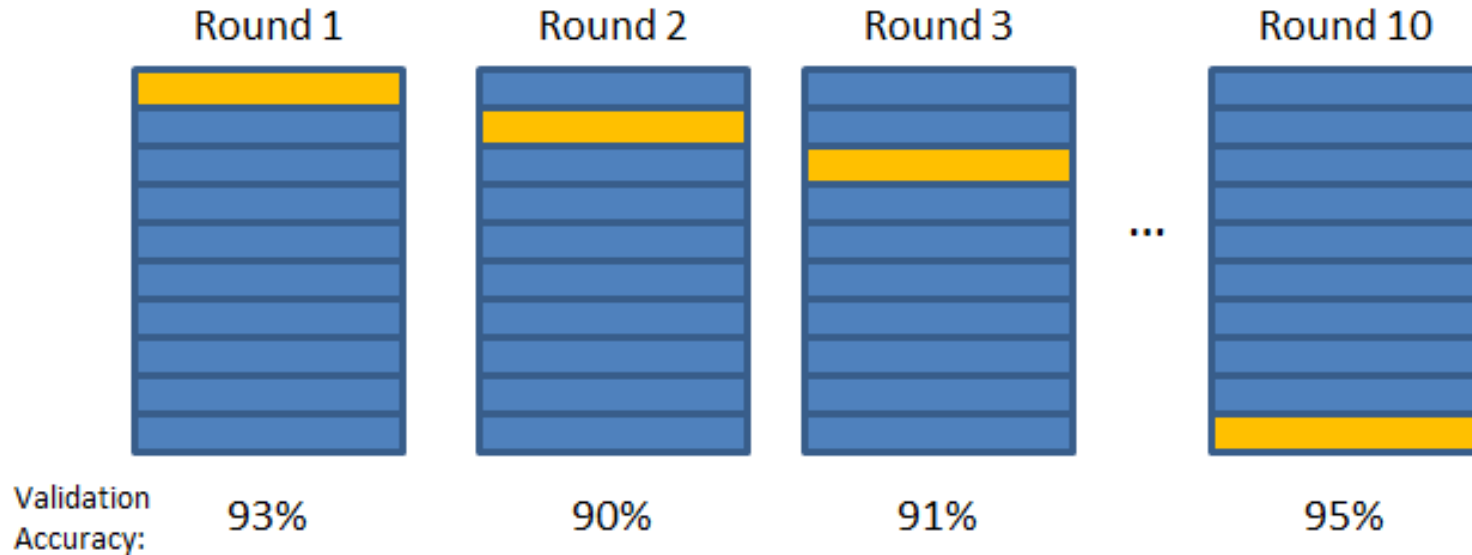
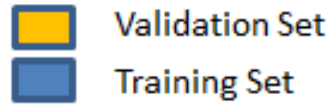
<http://scott.fortmann-roe.com/docs/BiasVariance.html>

# Cross Validation

- Method to estimate the generalization error of a machine learning model/algorithm
- Make multiple models on different subsets of the available data
- Average the test results to produce an estimate of the generalization error
- <https://msdn.microsoft.com/library/azure/75fb875d-6b86-4d46-8bcc-74261ade5826>



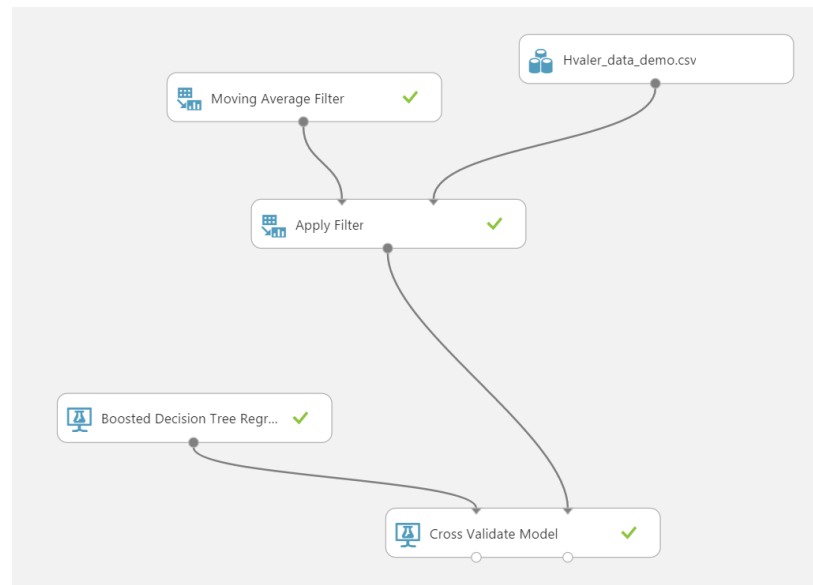
# 10-fold Cross Validation



Final Accuracy = Average(Round 1, Round 2, ...)



# Cross validation– Hands on



# Class imbalance problem

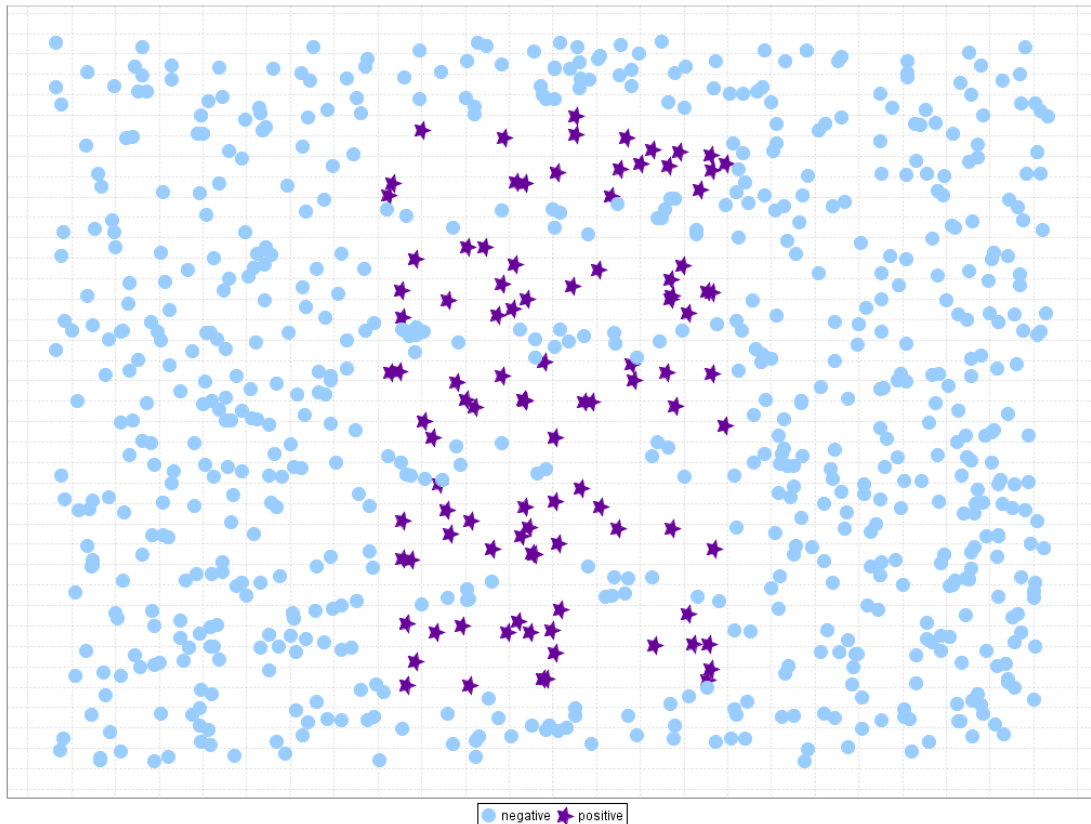
Imagine we have

9900 examples of class A

100 examples of class B

A model that always predicts A will  
be 99% accurate

This is obviously not what we want!

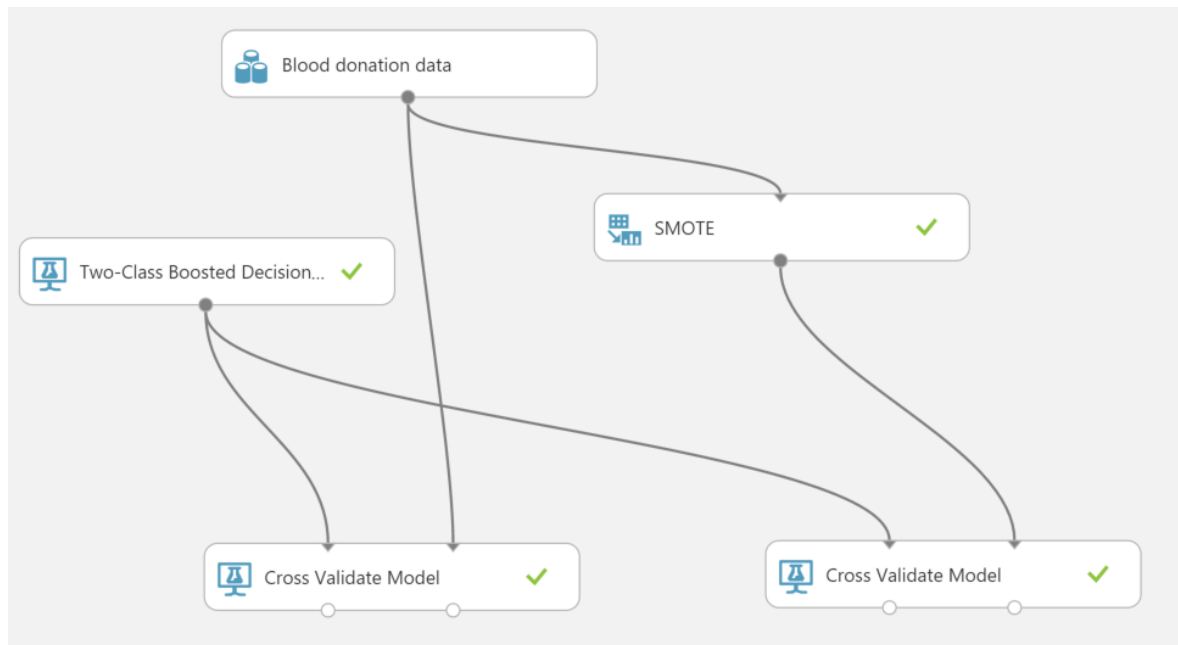


# Some solutions the class imbalance problem

- Downsampling
  - Select a subsample of the A examples such that it's size matches the set of B examples
- Upsampling
  - Produce artificial B examples ones until its size matches the set of A examples
  - SMOTE (Synthetic Minority Over-sampling Technique)
  - <https://msdn.microsoft.com/library/azure/9f3fe1c4-520e-49ac-a152-2e104169912a>
- Asymmetric training
  - Use an asymmetric error function to artificially balance the training process

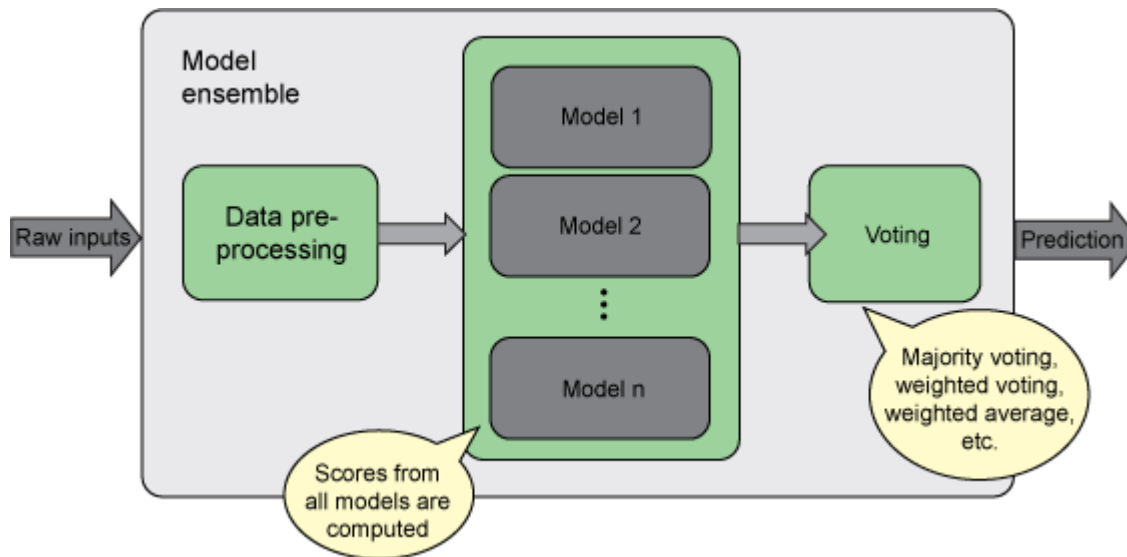


# Class Imbalance – Hands on



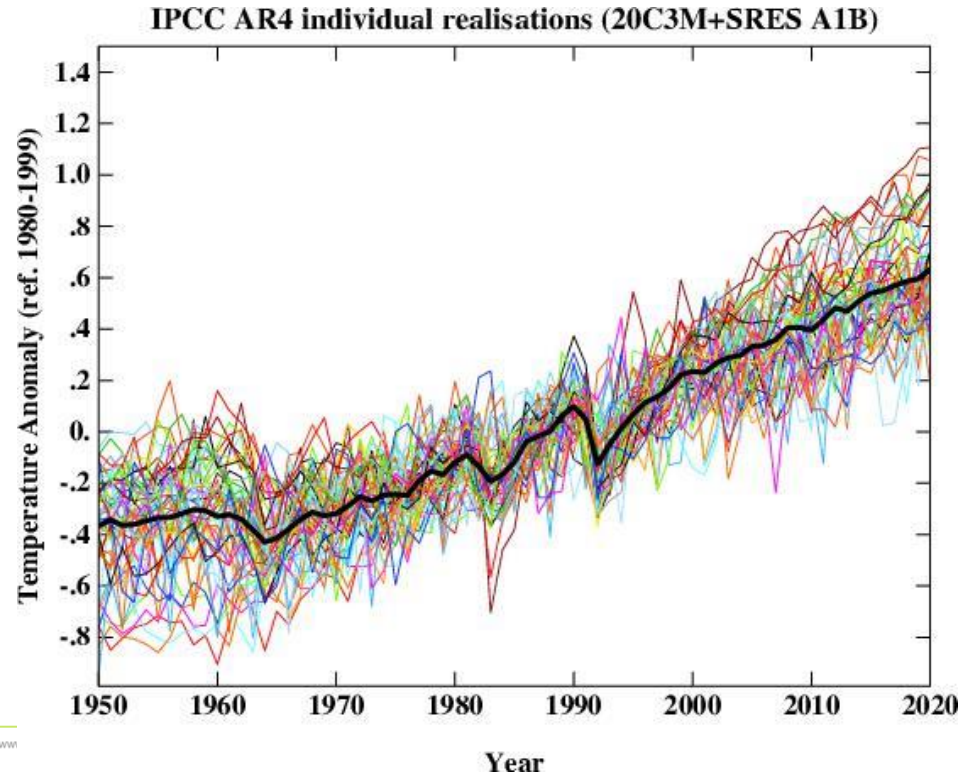
# Ensemble modeling

*"The Many Are Smarter Than the Few"*



# Ensemble Modeling

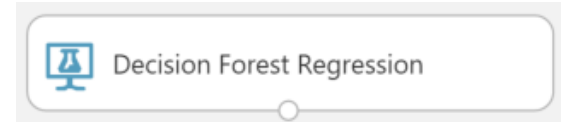
- Effectively reduces the variance of the model estimate
- Mitigates overfitting
- The diversity of the individual models is key



# Ensemble learning methods

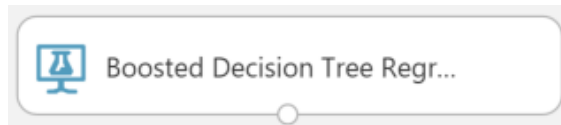
## ■ Bagging

- Each model is trained on a random subset of the data
- Averaged
- (<https://msdn.microsoft.com/en-us/library/azure/dn905862.aspx>)



## ■ Boosting

- Incrementally add new models
- Data previously estimated or classified unaccurately is given higher "weight" in the new model to be added to the ensemble
- (<https://msdn.microsoft.com/en-us/library/azure/dn905801.aspx>)



## ■ Stacking

- Each model is trained on a random subset of the data
- A new model weights the contributions of each individual model

## Machine Learning in ML Studio

### Anomaly Detection

One-class Support Vector Machine  
Principal Component Analysis-based Anomaly Detection  
Time Series Anomaly Detection\*

### Classification

#### Two-class Classification

Averaged Perceptron  
Bayes Point Machine  
Boosted Decision Tree  
Decision Forest  
Decision Jungle  
Logistic Regression  
Neural Network  
Support Vector Machine

#### Multi-class Classification

Decision Forest  
Decision Jungle  
Logistic Regression  
Neural Network  
One-vs-all

### Clustering

K-means Clustering

### Recommendation

Matchbox Recommender

### Regression

Bayesian Linear Regression  
Boosted Decision Tree  
Decision Forest  
Fast Forest Quantile Regression  
Linear Regression  
Neural Network Regression  
Ordinal Regression  
Poisson Regression

### Statistical Functions

Descriptive Statistics  
Hypothesis Testing T-Test  
Linear Correlation  
Probability Function Evaluation

### Text Analytics

Feature Hashing  
Named Entity Recognition  
Vowpal Wabbit

### Computer Vision

OpenCV Library

### Data/Model Visualization

- Scatterplots
- Bar Charts
- Box plots
- Histogram
- R and Python Plotting Libraries
- REPL with Jupyter Notebook
- ROC, Precision/Recall, Lift
- Confusion Matrix
- Decision Tree\*

### Training

- Cross Validation
- Retraining
- Parameter Sweep

<https://studio.azureml.net>

Guest Access Workspace: Free trial access without logging in.

Free Workspace: Free persisted access, no Azure subscription needed.

Standard Workspace: Full access with SLA under an Azure subscription.

Cross browser drag & drop ML workflow designer.

Zero installation needed.

### Unlimited Extensibility

- R Script Module
- Python Script Module
- Custom Module
- Jupyter Notebook

Built-in ML Algorithms

Import Data

Preprocess

Split Data

Train Model

Score Model

Training Experiment

One-click Operationalization

Predictive Experiment

### Make Prediction with Elastic APIs

- Request-Response Service (RRS)
- Batch Execution Service (BES)
- Retraining API

### Data Source

- Azure Blob Storage
- Azure SQL DB
- Azure SQL DW\*
- Azure Table
- Desktop Direct Upload
- Hadoop Hive Query
- Manual Data Entry
- OData Feed
- On-prem SQL Server\*
- Web URL (HTTP)

### Data Format

- ARFF
- CSV
- SVMLight
- TSV
- Excel
- ZIP

### Data Preparation

- Clean Missing Data
- Clip Outliers
- Edit Metadata
- Feature Selection
- Filter
- Learning with Counts
- Normalize Data
- Partition and Sample
- Principal Component Analysis
- Quantize Data
- SQLite Transformation
- Synthetic Minority Oversampling Technique

### Enterprise Grade Cloud Service

- SLA: 99.95% Guaranteed Up-time
- Azure AD Authentication
- Compute at Large Scale
- Multi-geo Availability
- Regulatory Compliance\*

### Community

- Gallery (<http://gallery.azureml.net>)
- Samples & Templates
- Workspace Sharing and Collaboration
- Live Chat & MSDN Forum Support

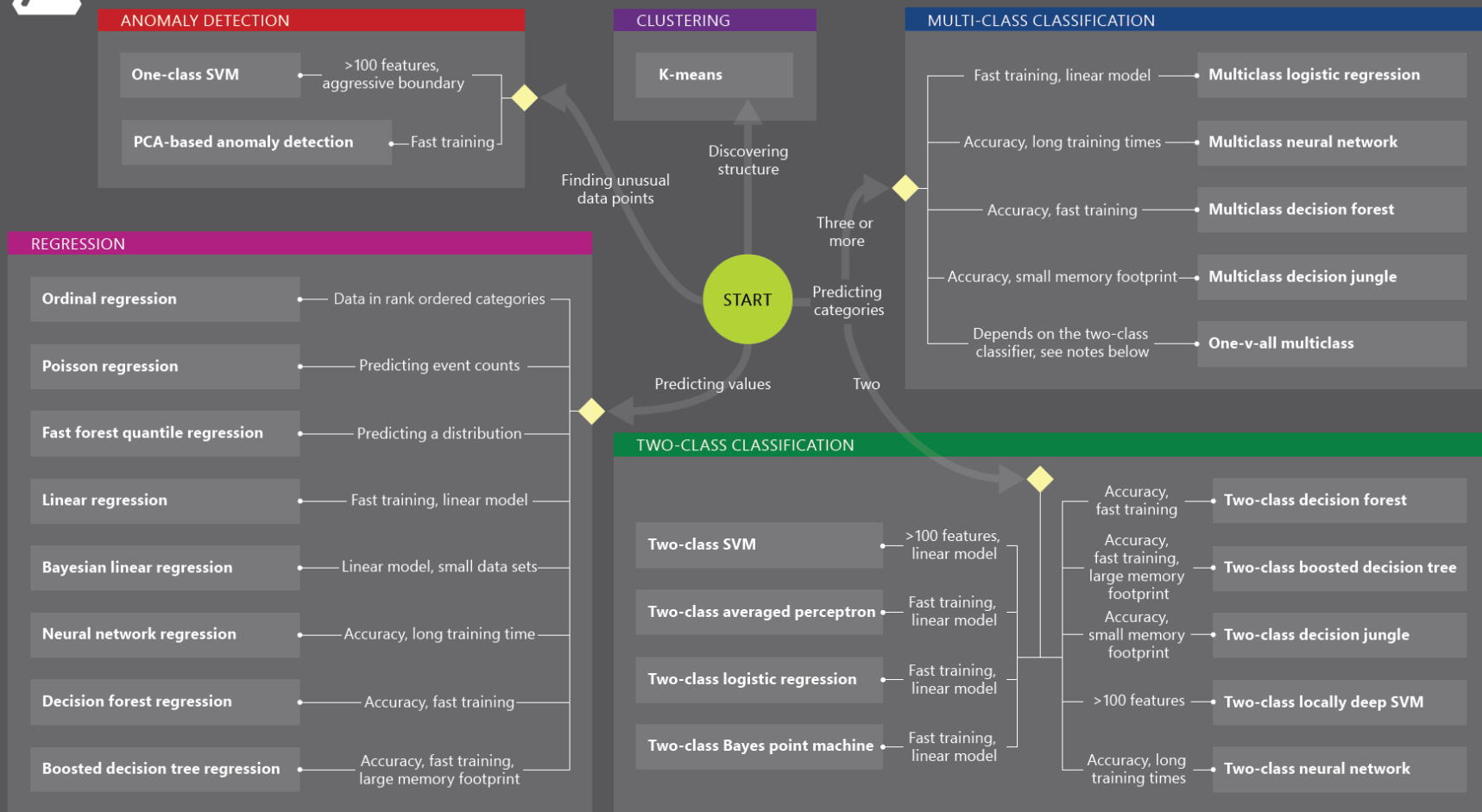
\* Feature Coming Soon





# Microsoft Azure Machine Learning: Algorithm Cheat Sheet

This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.



scikit-learn  
algorithm cheat-sheet

