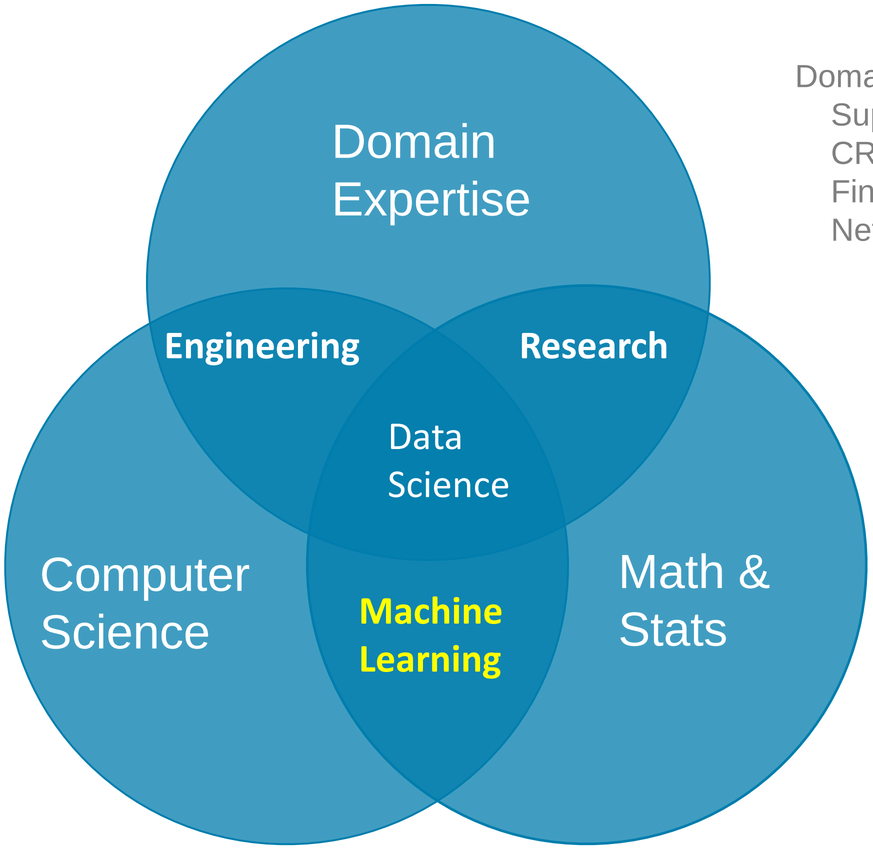
Introduction to Machine Learning

Agenda

8:30am - 9am	**Breakfast, Socialize**
9:00am – 10:00am	**Kickoff, Overview of Machine Learning**
10:00am – 10:15am	**Break**
10:15am – 12:00pm	**Lab 1 - Machine Learning w/ Python & Spark pipeline**
12:00 pm – 1pm	**Lunch**
1pm – 2:15pm	**Lab 2 – Building an ML model w/ GUI**
2:15pm – 2:30pm	**Break**
2:30pm – 3:45pm	**Lab 3 - Intro to Principal Component Analysis w/ Spark**
4pm – 4:30pm	**Wrap up – Feedback from attendees**

Machine Learning and Data Science….



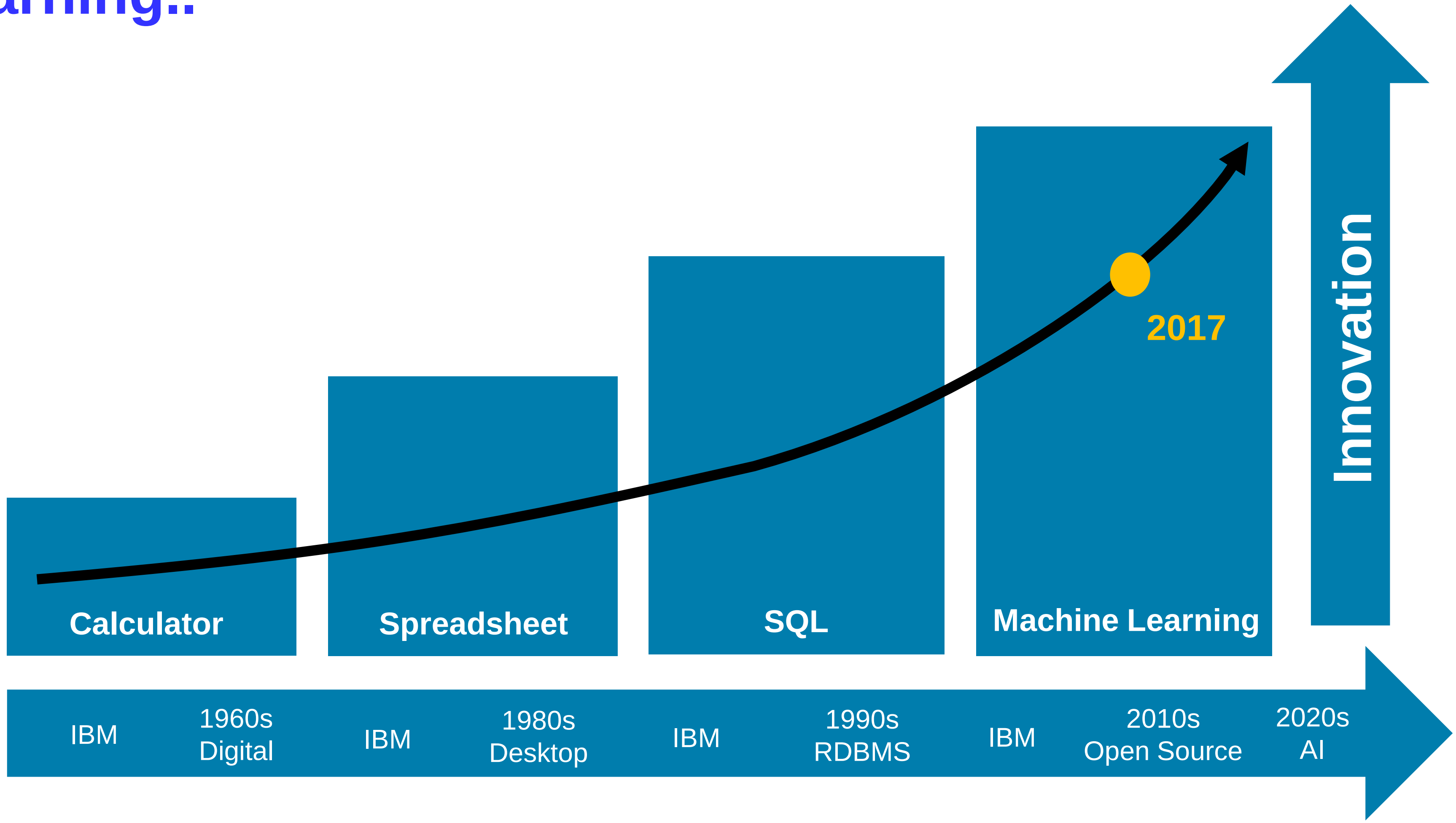
Domain Knowledge
- Supply Chain
- CRM
- Financials
- Networking

Scripting, SQL
Python, R Scala
Data Pipelines
Big Data/ Apache
Spark

Mathematics
Computational

Data Science Projects Require Multiple Skills

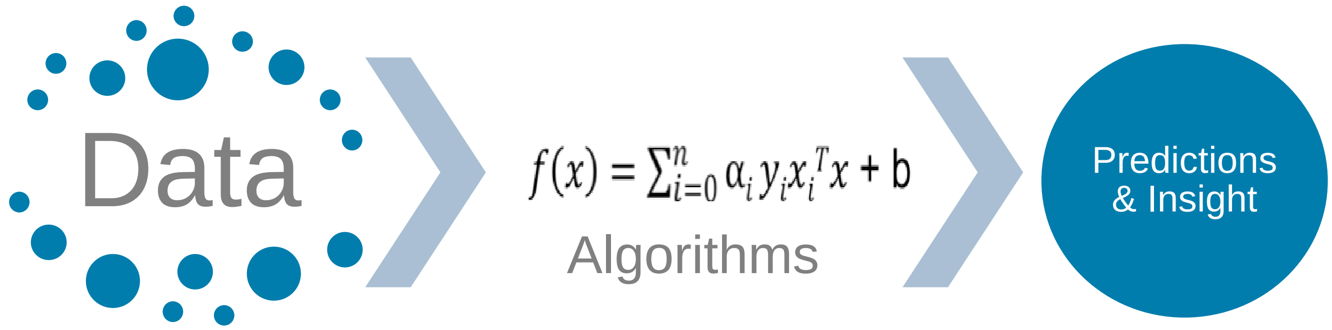
Future of Data Science is Democratizing Machine Learning..



But what is Machine Learning?

“Computers that learn without being explicitly programmed”

“Using algorithms to understand patterns in data”

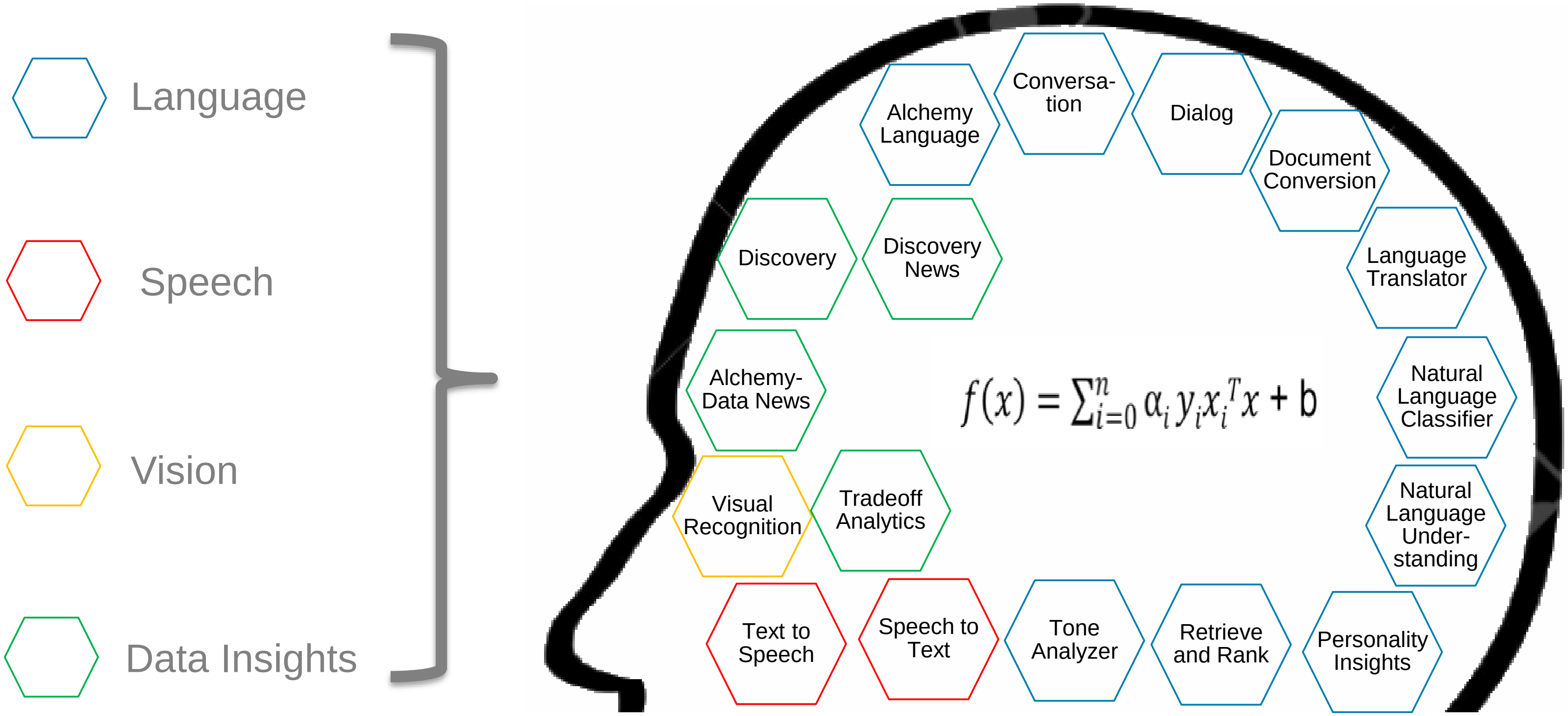


But what is Artificial Intelligence?

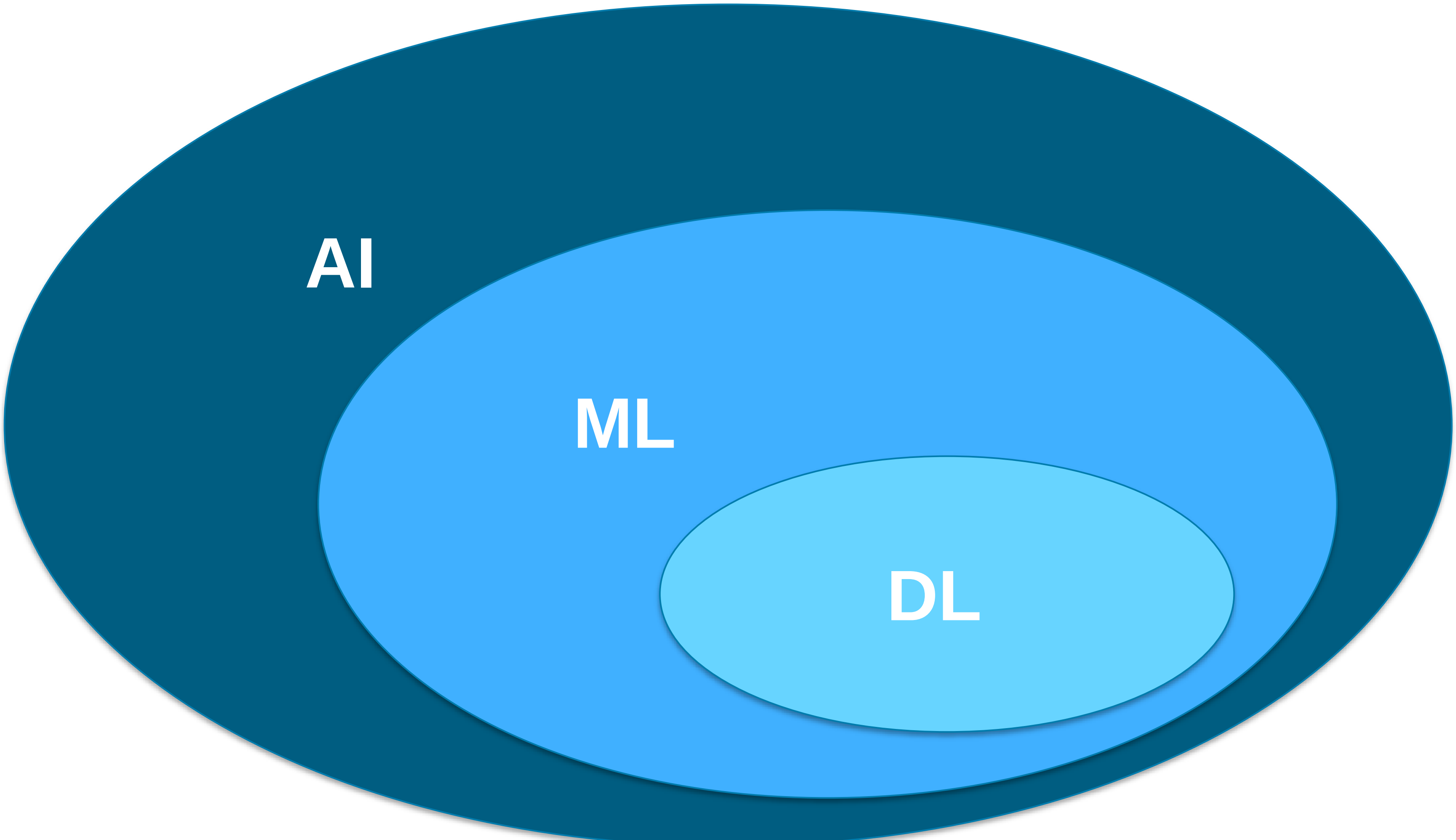
A theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages..

Machine Learning = Artificial Intelligence???

Data + Algorithms = Scored AI Models



Understanding AI, ML & DL Relationship…



Top 10 Use Cases for Data Science & Machine Learning

HEALTHCARE:
Patient Diagnosis

FINANCE:
Fraud Detection

MANUFACTURING:
Anomaly Detection

RETAIL:
Inventory Optimization

GOVERNMENT:
Smarter Services

TRANSPORTATION:
Demand Forecasting

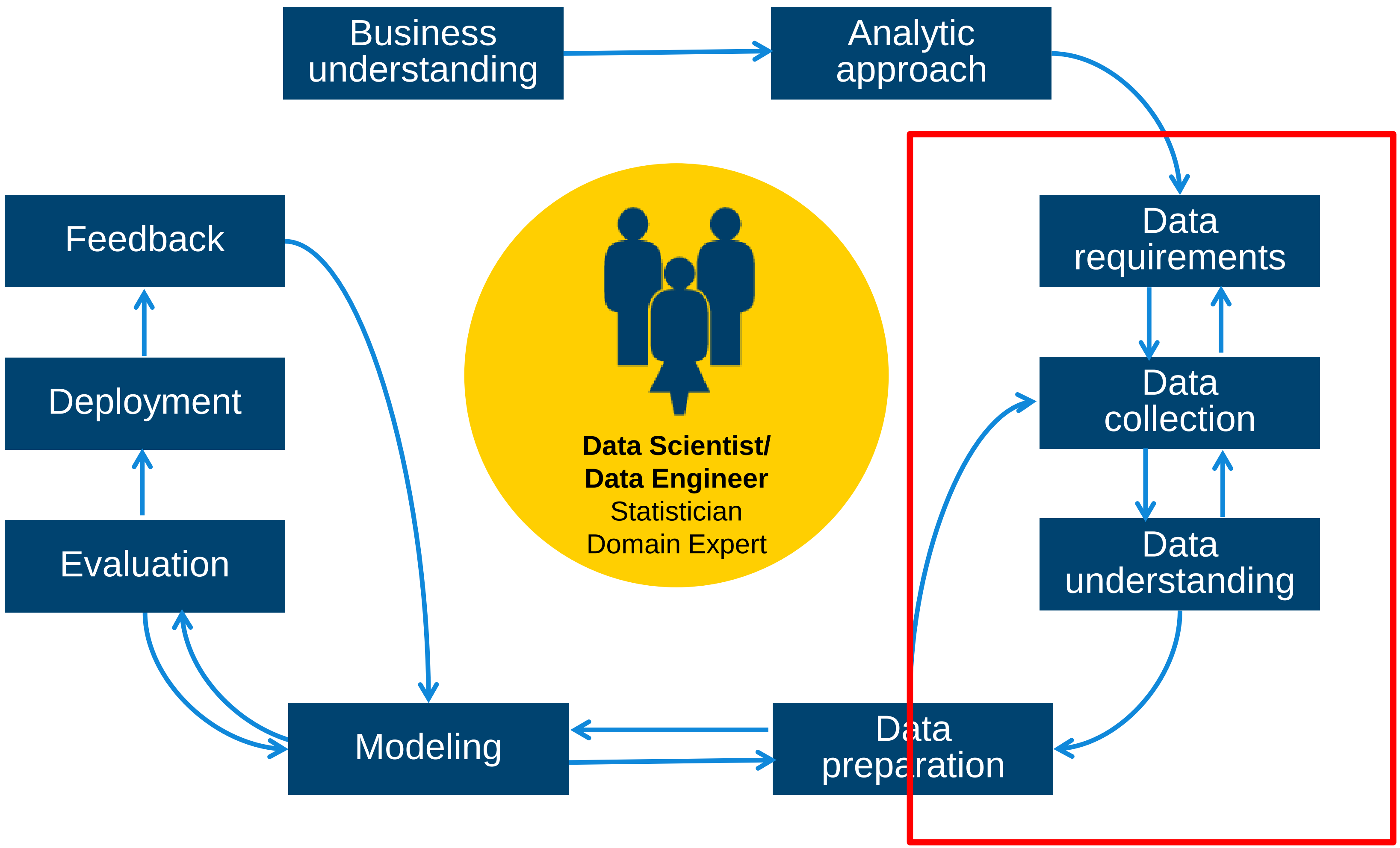
NETWORKS:
Intrusion Detection

E-COMMERCE:
Recommender Systems

MEDIA:
Interaction & Speed

EDUCATION:
Research Insight

Data Science Methodology

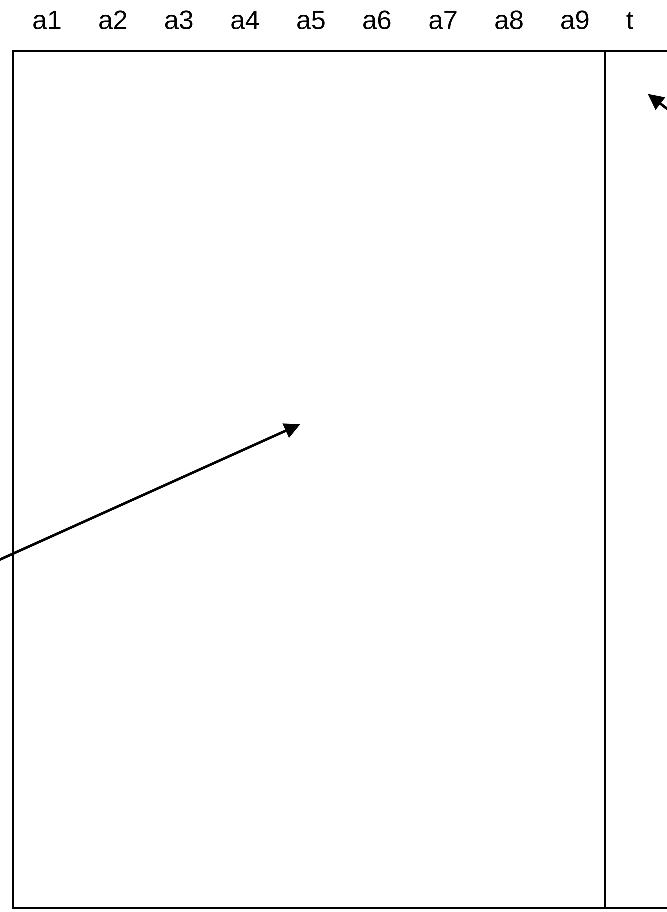


Matrix for Machine Learning

<u>Known as:</u>
- Attributes
- Features
- Predictor variables
- Explanatory variables

<u>Scale variables:</u>
- Continuous variables, which can be measured on an interval scale or ratio scale
- ‘Weight’, ‘Temperature’, ‘Salary’, etc…

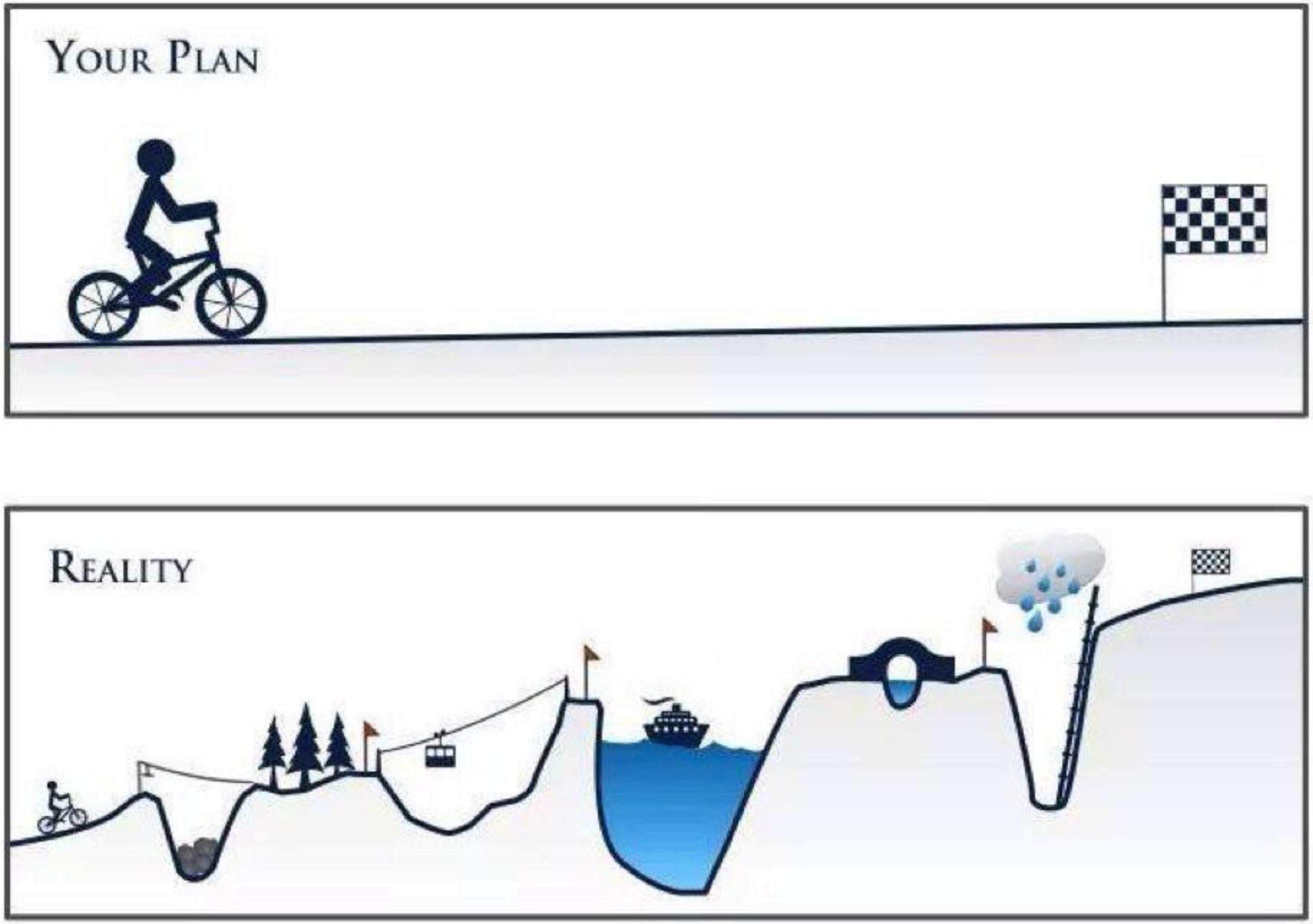
<u>Categorical variables:</u>
- Data with a limited number of distinct values or categories (nominal or ordinal)
- ‘Hair color’, ‘Gender’, ‘Grape varieties’, etc…



<u>Known as:</u>
- Label
- Target variable
- Dependent variable

Scale or Categorical

Plans never survive first contact with the data



Data Understanding – Data Audit

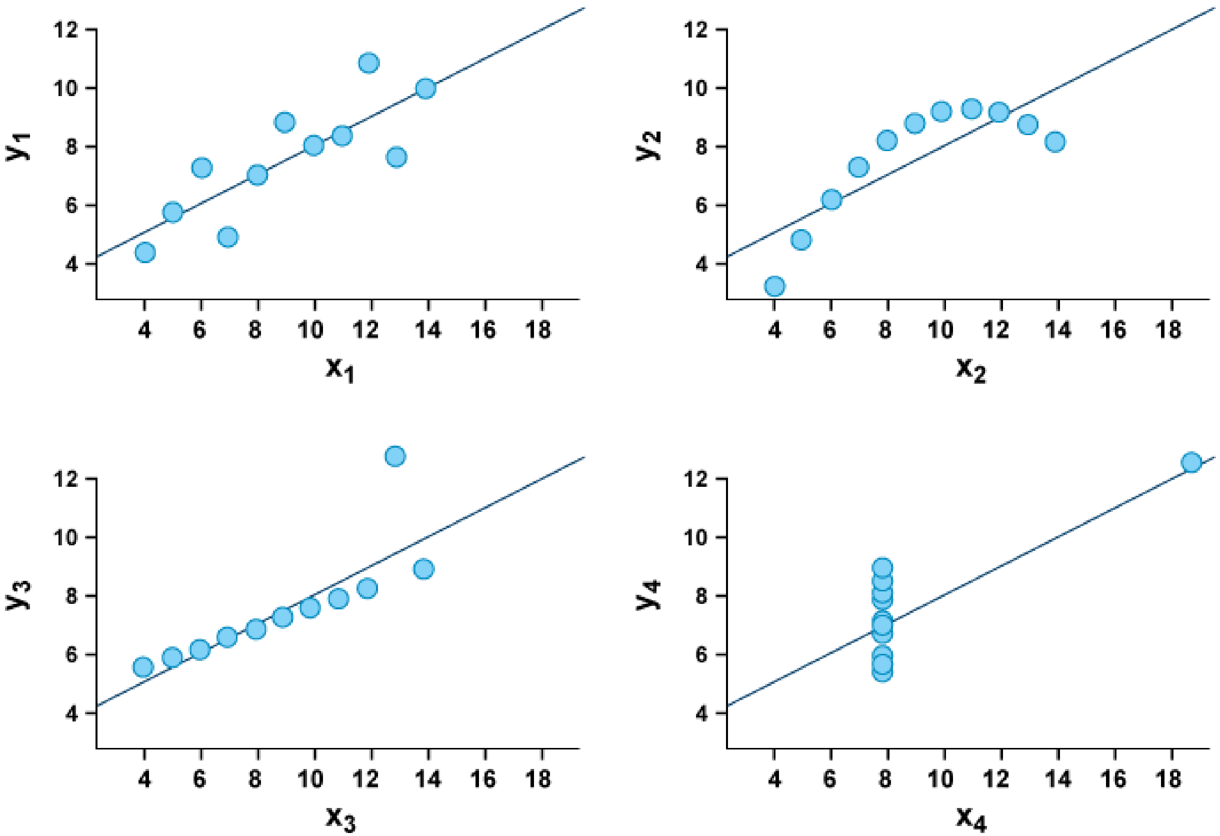
- **Data can be missing values**
  - Blank fields
  - Fields with dummy values (9999)
  - Fields with “U” or “Unknown”

- **Data can be corrupt or incoherent or anomalous:**
  - Data fields can be in the wrong place (strings where numbers are expected)
  - Spurious “End of Line” characters can chop original lines of data into several lines and cause data fields in the wrong place
  - Data entered in different formats: USA / US / United States

- **Data can be duplicated**

- **Handling these data quality issues (as part of data preparation) is often referred to as:**
  - Data cleansing / wrangling

Data Understanding: Visualizations



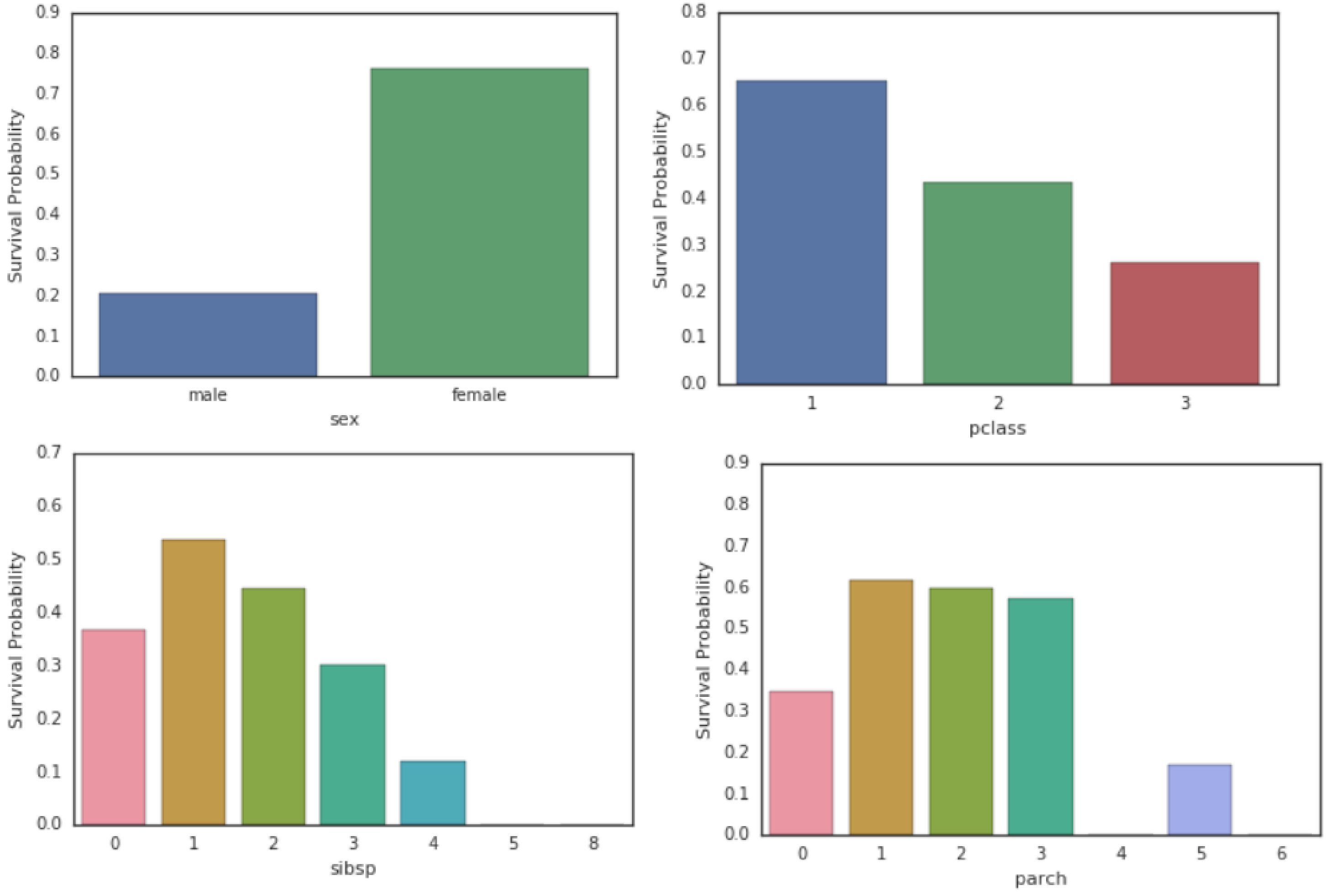
The four data sets have similar statistical properties:
- The mean of x is 9
- The variance of x is 11
- The mean of y is approx. 7.50
- The variance of y is approx. 4.12
- The correlation is 0.816

As shown the linear regression lines are approx. y=3.00+0.500x.

- **Anscombe's quartet**
  - The four datasets have nearly identical statistical properties (mean, variance, correlation), yet the differences are striking when looking at the simple visualization

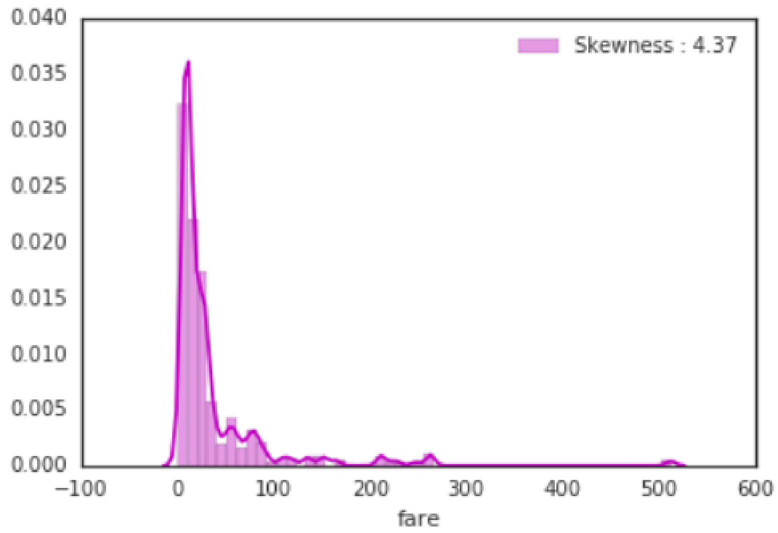
Data Understanding: Visualizations

- **Titanic Data**
- **Univariate Relationships**

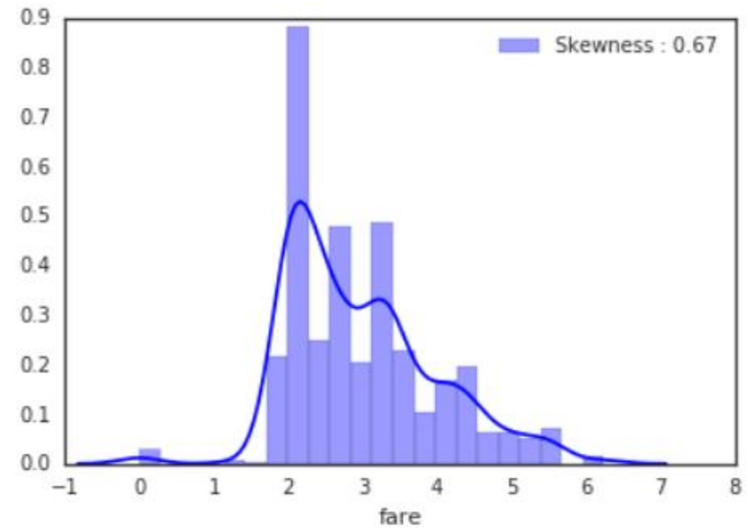


Data Understanding: Visualizations

- **Titanic Data**
- **Skewed Data**



Original Data



After Log Transform

Data Preparation

- **Data preparation can be very time consuming depending on:**
  - The state of the original data
    - Data is typically collected in a “human” friendly format
  - The desired final state of the data (as required by the machine learning models and algorithms)
    - The desired final state is typically some “algorithm” friendly format
  - There may be a need for a (long) pipeline of transformations before the data is ready to be consumed by a model:
    - These transformations can be done manually (write code)
    - These transformations can be done through tools

Data Preparation – Transformation

- **Data may need to be transformed to match algorithms requirements:**
  - Tokenizing (typical in text processing)

  - Vectorizing (several algorithms in Spark MLlib require this)
    - Transform data into Vector arrays
    - Can be done manually (write Python or Scala code)
    - Can be done using tools (VectorAssembler in the new ML package)
      - (TF-IDF in text processing)
      - Word2Vec

  - Bucketizing
    - Transform a range of continuous values into a set of buckets

Data Preparation - Transformation

- **Data may need to be transformed to match algorithms requirements:**
  - Standardization
    - Transform numerical data to values with zero mean and unit standard deviation
    - Linear Regression with SGD in Spark MLlib requires this

```
labelDataOneVar.take(5).foreach(println)
scaledDataOneVar.take(5).foreach(println)
(119900.0,[1634.0])
(399990.0,[2756.0])
(399900.0,[3710.0])
(399900.0,[2362.0])
(399900.0,[3515.0])
(119900.0,[-0.7785438395777196])
(399990.0,[0.8850502978217635])
(399900.0,[2.2995501258780084])
(399900.0,[0.30086483424297883])
(399900.0,[2.0104228025331783])
Took 5 seconds.
```

House price

Square footage

House price

Standardized Square footage

Data Preparation - Transformation

- **Data may need to be transformed to match algorithms requirements:**
  - Normalization
    - Transform data so that each Vector has a Unit norm
  - Categorical values need to be converted to numbers
    - This is required by Spark MLlib classification trees
    - Marital Status: {“Widowed”, “Married”, “Divorced”, “Single”}
    - Marital Status: {0, 1, 2, 3}
    - You cannot do this if the algorithm could infer: Single = 3 X Married ☺

Data Preparation – Transformation

- **Data may need to be transformed to match algorithms requirements:**
  - Dummy encoding
    - When categorical values cannot be converted to consecutive numbers
    - Marital Status: {“Single”, “Married”, “Divorced”, “Widowed”}
    - Marital Status: {“0001”, “0010”, “0100”, “1000”}
    - This is necessary if the algorithm could make some wrong inference from the numerical based categorical encoding:
      - Single = 3
      - Married = 2
      - Divorced = 1
      - Widowed = 0
        - Single = Married + Divorced
        - Single = Divorced x 3
        - (this is a contrived example, but you get the idea ☺, replace marital status with colors…)

Data Preparation – Dimensionality Reduction

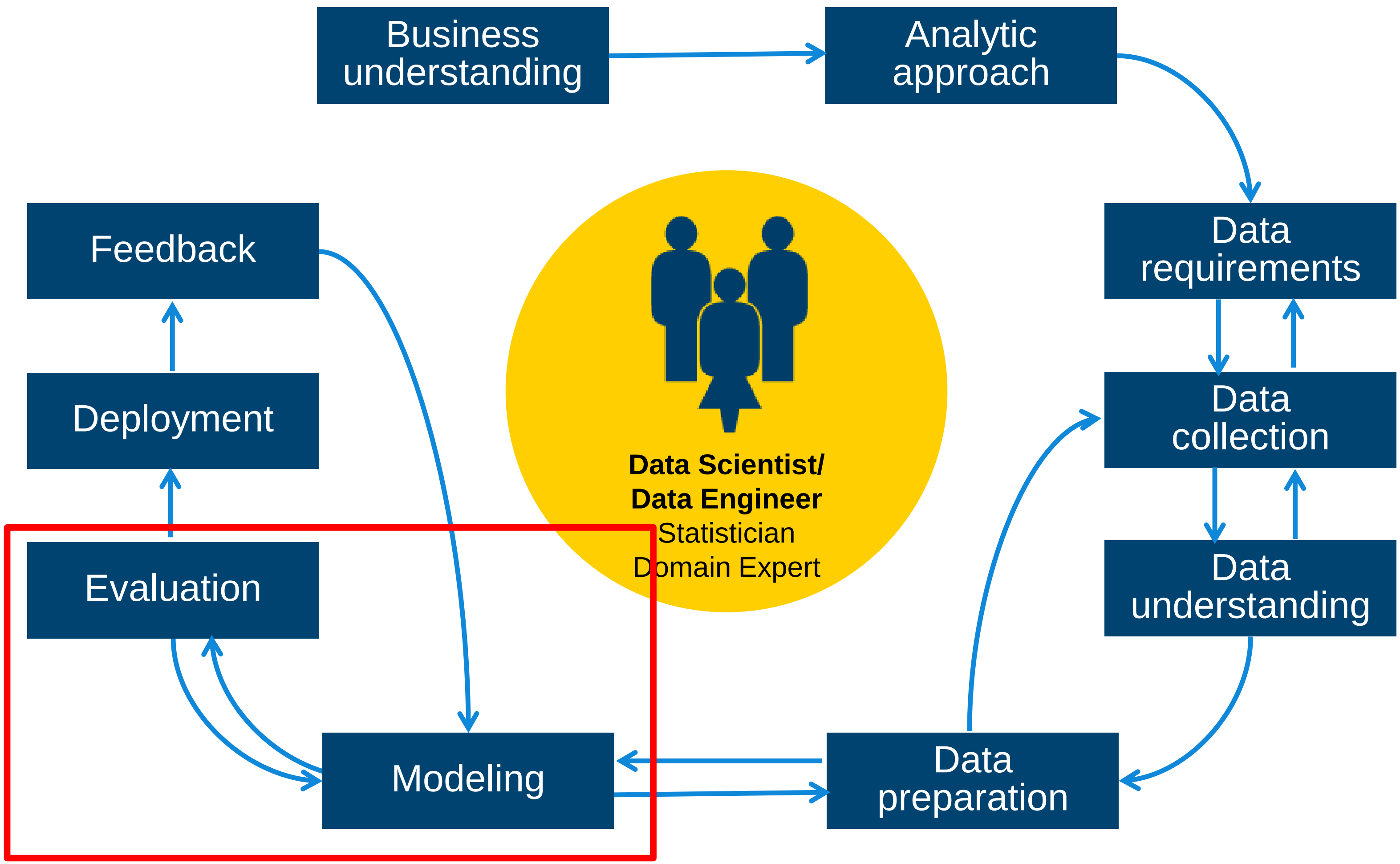
- **Data dimensionality may need to be reduced:**

- **The idea behind reducing data dimensionality is that raw data tends to have two subcomponents:**
  - "Useful features" (aka structure)
  - Noise (random and irrelevant)
  - Extracting the structure makes for better models

  - Examples of applications of dimensionality reduction
    - Extracting the important features in face/pattern recognition
    - Removing stop words when working on text classification
    - Stemming: fishing, fished, fisher ➔ fish

  - Examples methods of dimensionality reduction
    - Principal Component Analysis
    - Singular Value Decomposition

Data Science Methodology



Machine Learning – A more formal definition

Tom Mitchell of Carnegie Mellon University provides a widely quoted, more formal definition of machine learning

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E"

Machine Learning vs Human Learning

- **In many aspects, ML not fundamentally different from HL:**
  – Repeat the same task over and over again to gain experience.
  – Action of repeating the same task is referred to as "practice"
  – With practice and experience, we get better at learned tasks.

- **Examples:**
  – Learning how to play a music instrument
  – Learning how to play a sport (golf, tennis, etc…)
  – Practicing for a math exams doing exercises
  – A teacher or coach will measure performance to evaluate progress
  – Practice makes perfect

Machine Learning Examples

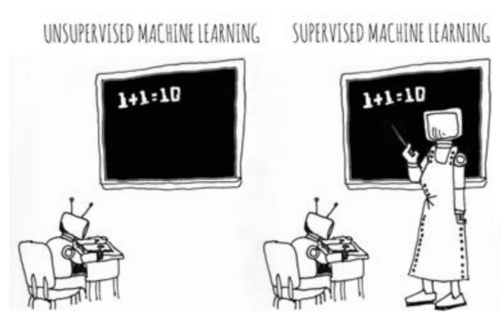
- Is this cancer ? (Medical diagnosis)
- Is this legitimate or fraud (spam) ?
- What is the market value of this house ?
- Which of these people are good friends with each other ?
- Will this engine fail (when) ?
- Will this person like this movie ?
- Who is this ?
- What did you say ? (Speech recognition)

Machine Learning solves problems that cannot be tackled by numerical means alone.

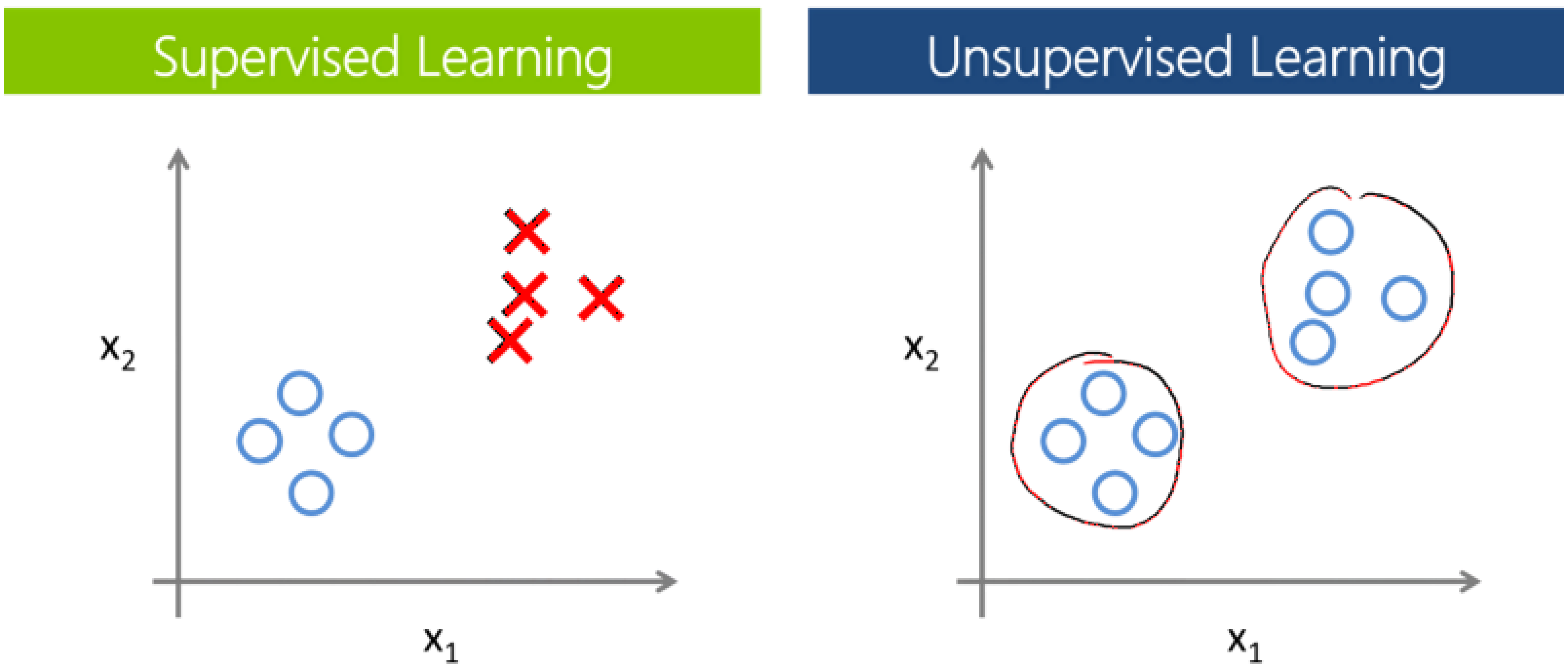
Categories of Machine Learning

- **Supervised learning**
  - The program is “trained” on a pre-defined set of “training examples”, which then facilitate its ability to reach an accurate conclusion when given new data
  - The algorithm is presented with example inputs and their desired outputs (correct results)
  - The goal is to learn a general rule that maps inputs to outputs

- **Unsupervised learning**
  - No labels are given to the learning algorithm, leaving it on its own to find structure (patterns and relationships) in its input
  - Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning)



Supervised vs. Unsupervised Learning



Categories of Machine Learning

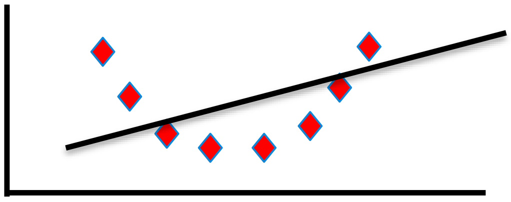
Technique	Usage	Algorithms
Classification (or prediction)	• Used to predict group membership (e.g., will this employee leave?) or a number (e.g., how many widgets will I sell?)	• Decision Trees • Logistic Regression • Random Forests • **Naïve Bayes** • Linear Regression • Lasso Regression etc
Segmentation	• Used to classify data points into groups that are internally homogenous and externally heterogeneous. • Identify cases that are unusual	• K-means • Gaussian Mixture • Latent Dirichlet allocation etc
Association	• Used to find events that occur together or in a sequence (e.g., market basket)	• FP Growth etc

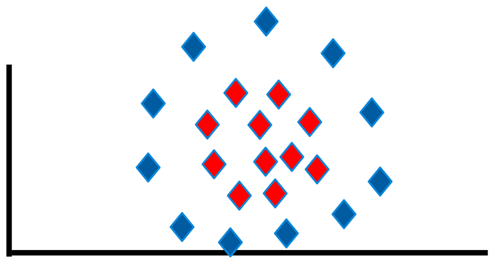
Categories of Machine Learning

Learning challenges

- **Under fitting:**
  - Not knowing enough “basic” concepts, i.e. not being well-equipped enough to tackle learning at hand:
    - You can’t study calculus without knowing some algebra.
    - You can’t learn playing hockey without knowing how to skate.
    - You can’t learn polo without knowing how to ride.
  - This can lead to under fitting in Machine Learning: The chosen model is just not “sophisticated”, “rich”, enough to capture the concept.

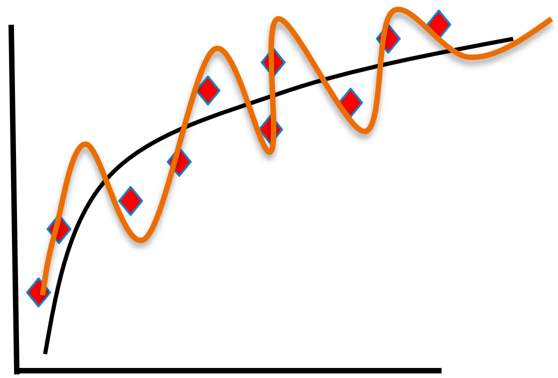
aX + BY + C



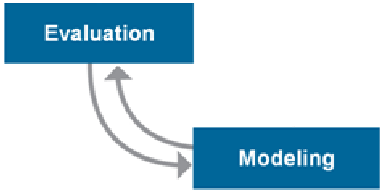


Learning challenges

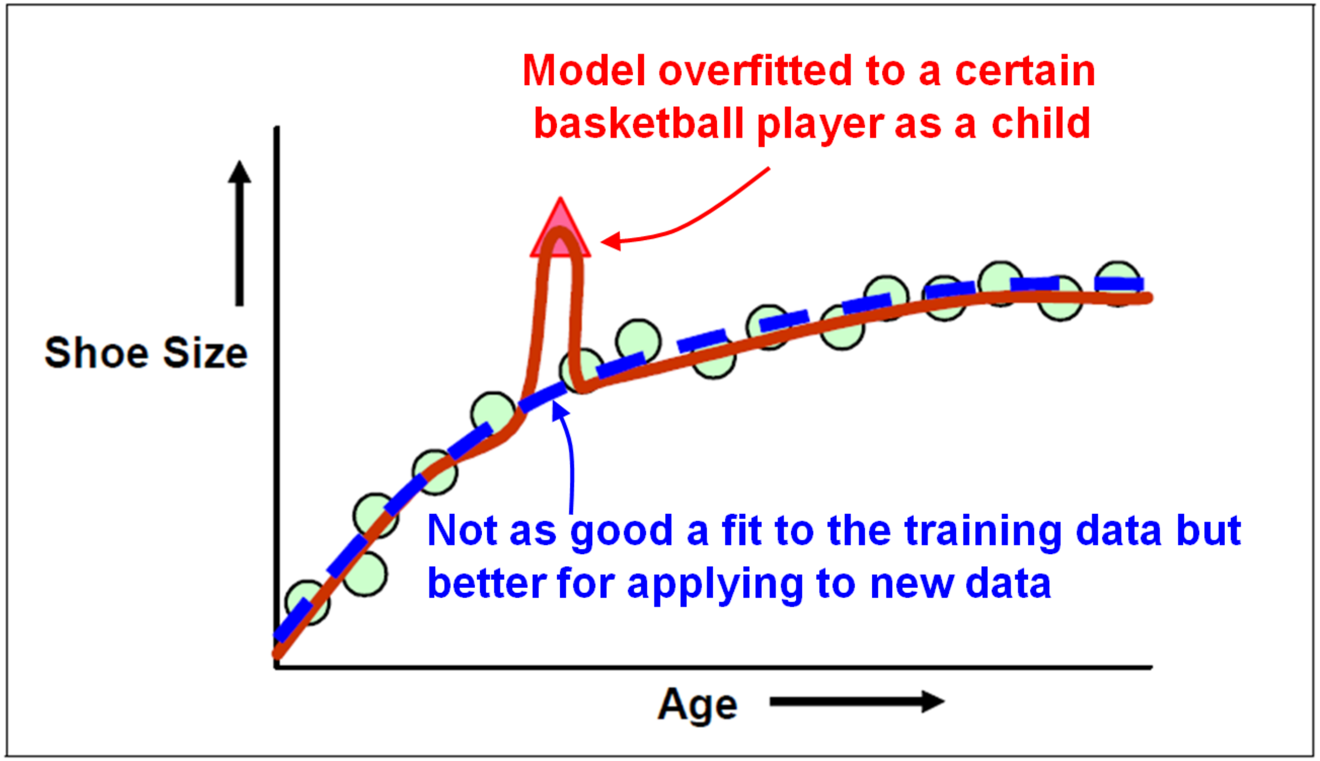
- **Over fitting:**
  - Hyper-sensitivity to minor fluctuations, ending up in modeling a lot of the unwanted noise in the data:
  - This can lead to over fitting in Machine Learning.



Model overfitting

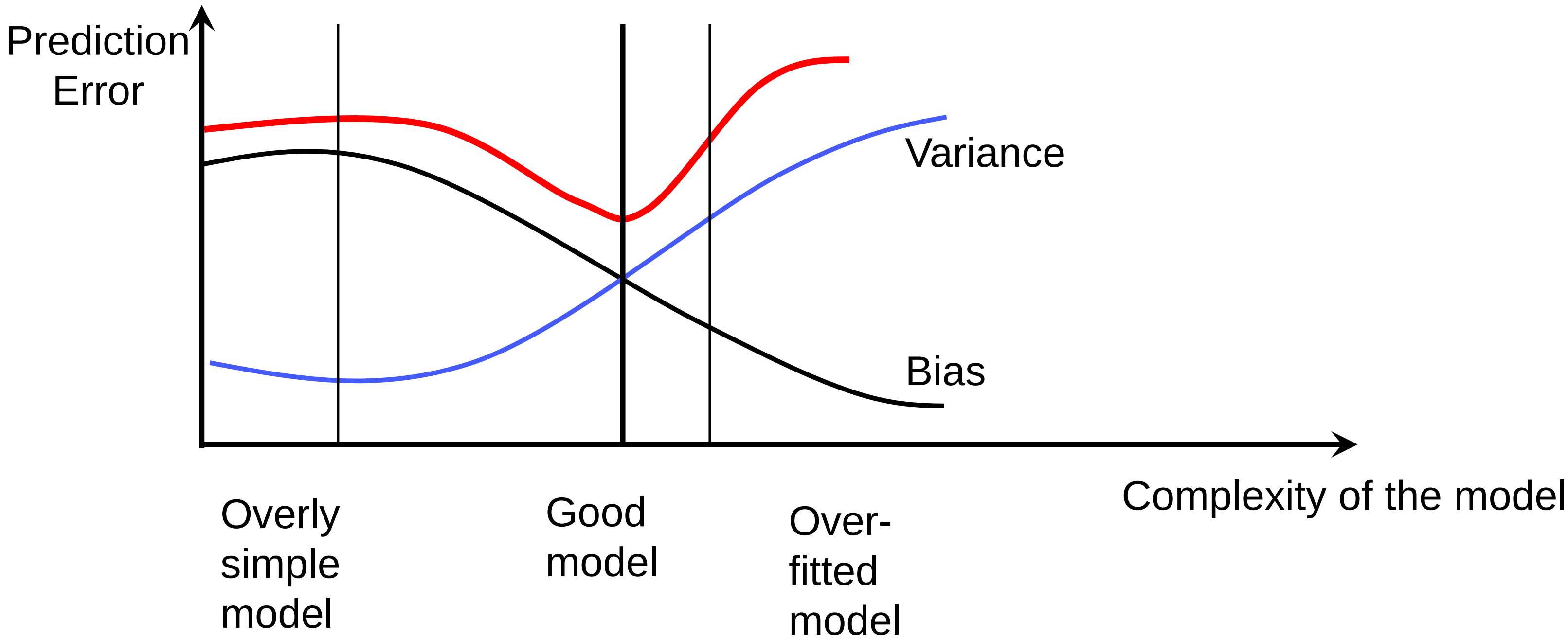


- When building a predictive model, there is a risk of overfitting the model to the training data.
- The model fits the training data very well, but it does not perform well when applied to new data.



Learning challenges

- **Compromise between bias and variance:**



Graphical illustration of bias vs variance

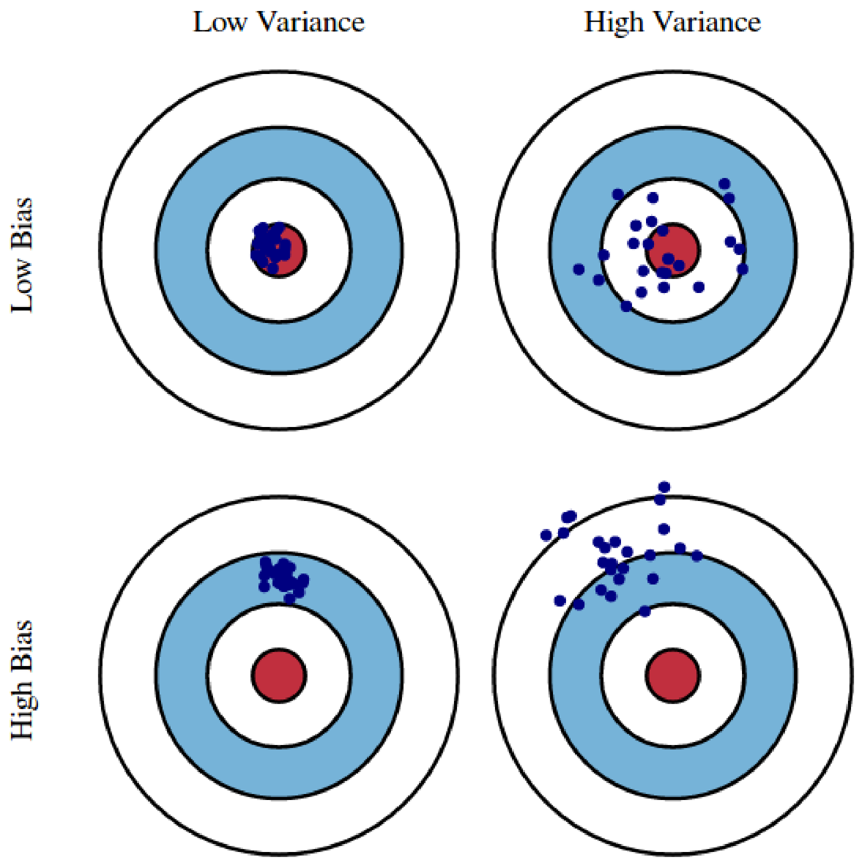
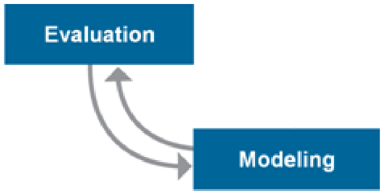


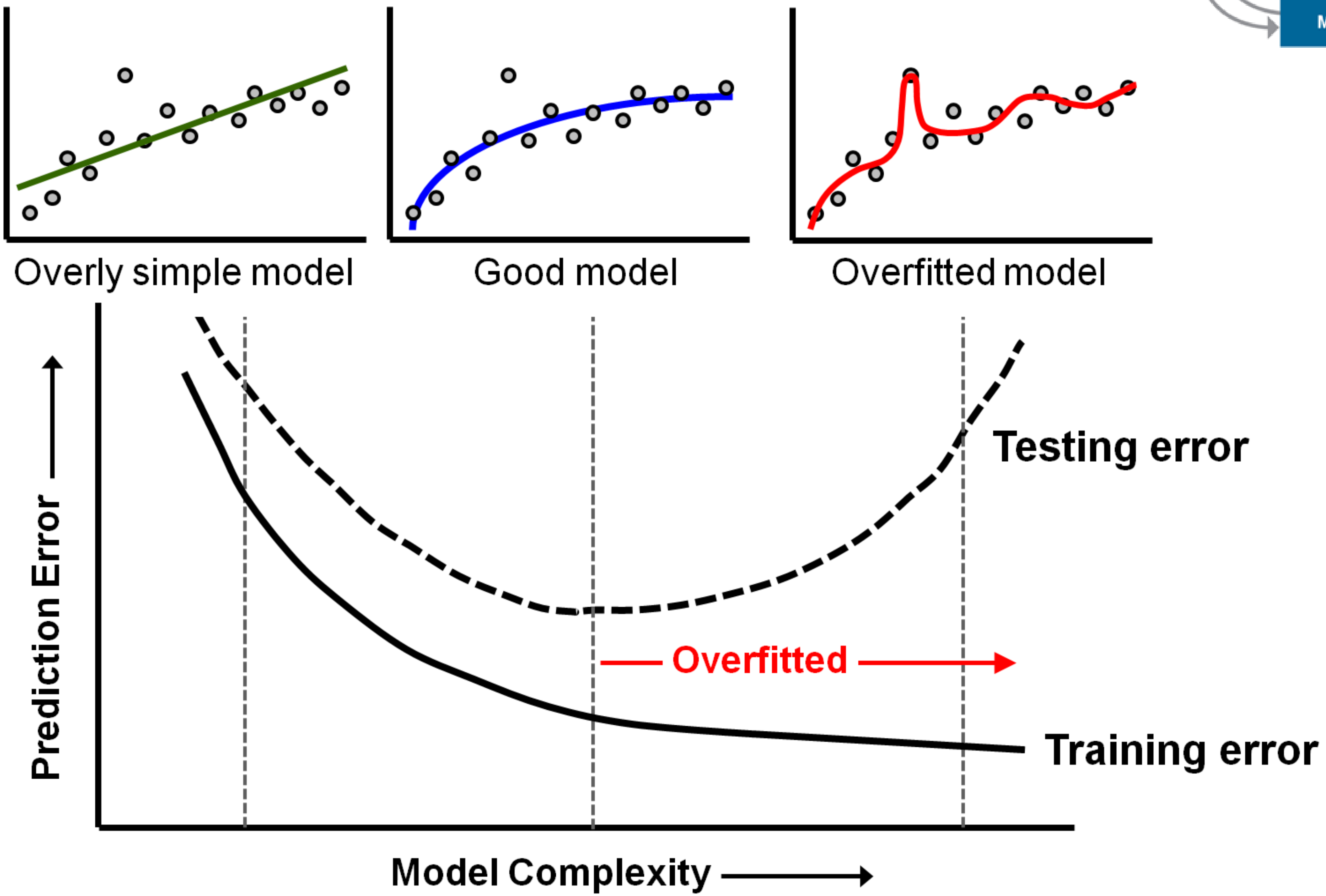
Fig. 1 Graphical illustration of bias and variance.

Learning challenges

- **Diminishing returns:**
  - People can:
    - Have more or less talent
    - get bored or enthusiastic
  - Machines will not, however:
  - Making progress initially is usually more easy, but improving gets harder as we move along. We may need to try different learning methods, styles to keep going:
    - Machine learning algorithms have hyper-parameters which need to be tuned properly.
    - It may be necessary to use more than just one single method / algorithm to reach the goal.

When to stop training a model

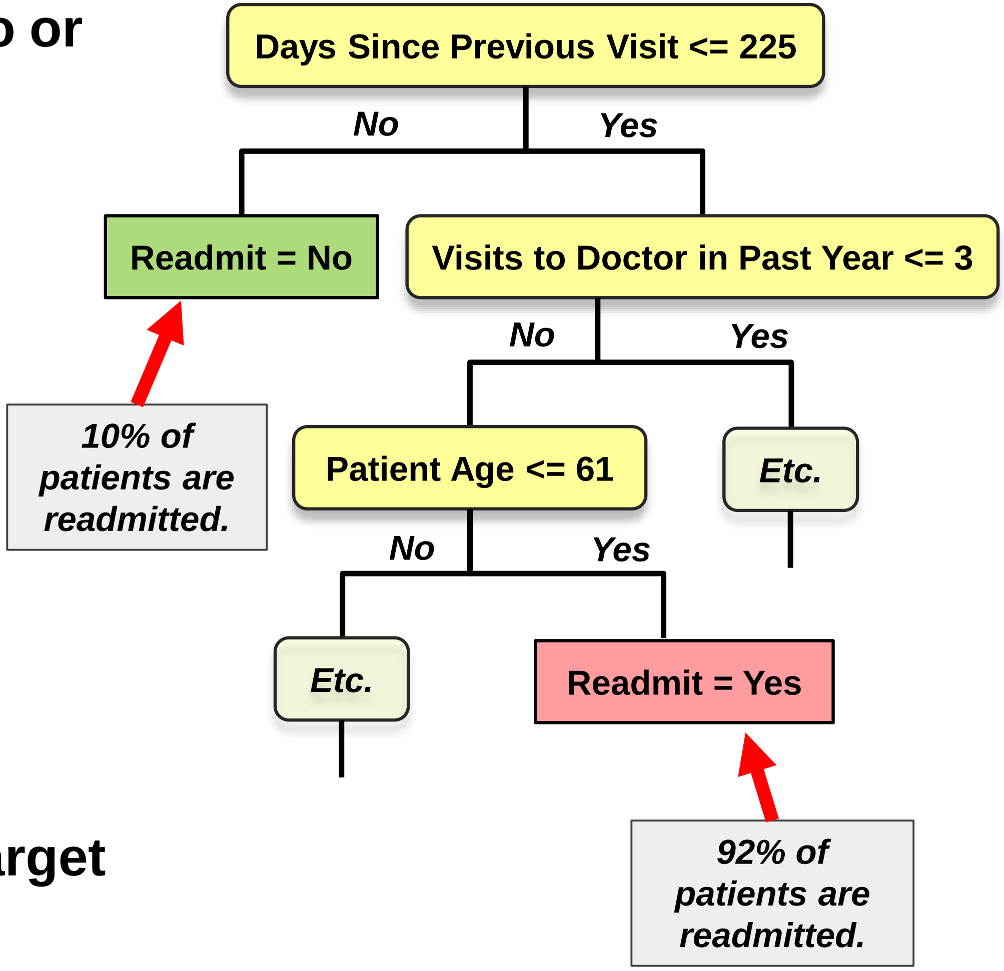




Classification – Decision tree (supervised)

Modeling

- **Class variable (target) with two or more outcomes.**
- **Splits records in a tree-like series of nodes along mutually-exclusive paths.**
  - Algorithm decides which variable and threshold value to use at each split
  - New records are predicted (classified) based on the leaf assignment
  - Accurate
  - Explicit decision paths
- **Can also handle continuous target ("regression tree").**



Classification – Naïve Bayes (supervised)

Modeling

- **Two or more outcomes.**
- **Assumes independence among explanatory variables, which is rarely true (thus “naïve”).**
- **Despite its simplicity, often performs very well… widely used.**
- **Significant use cases:**
  - Text categorization (spam vs. legitimate, sports or politics, etc.) using word frequencies as the features
  - Medical diagnosis (*e.g.*, automatic screening)
  - To mark an email as spam or not spam
  - Check a piece of text expressing positive emotions, or negative emotions?
  - Used for face recognition software.

Classification – Naïve Bayes

Modeling

Outlook	Temp	Humidity	Windy	Play golf
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Classification – Naïve Bayes

Modeling

Frequencies and probabilities for the weather data:

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Classification – Naïve Bayes

Modeling

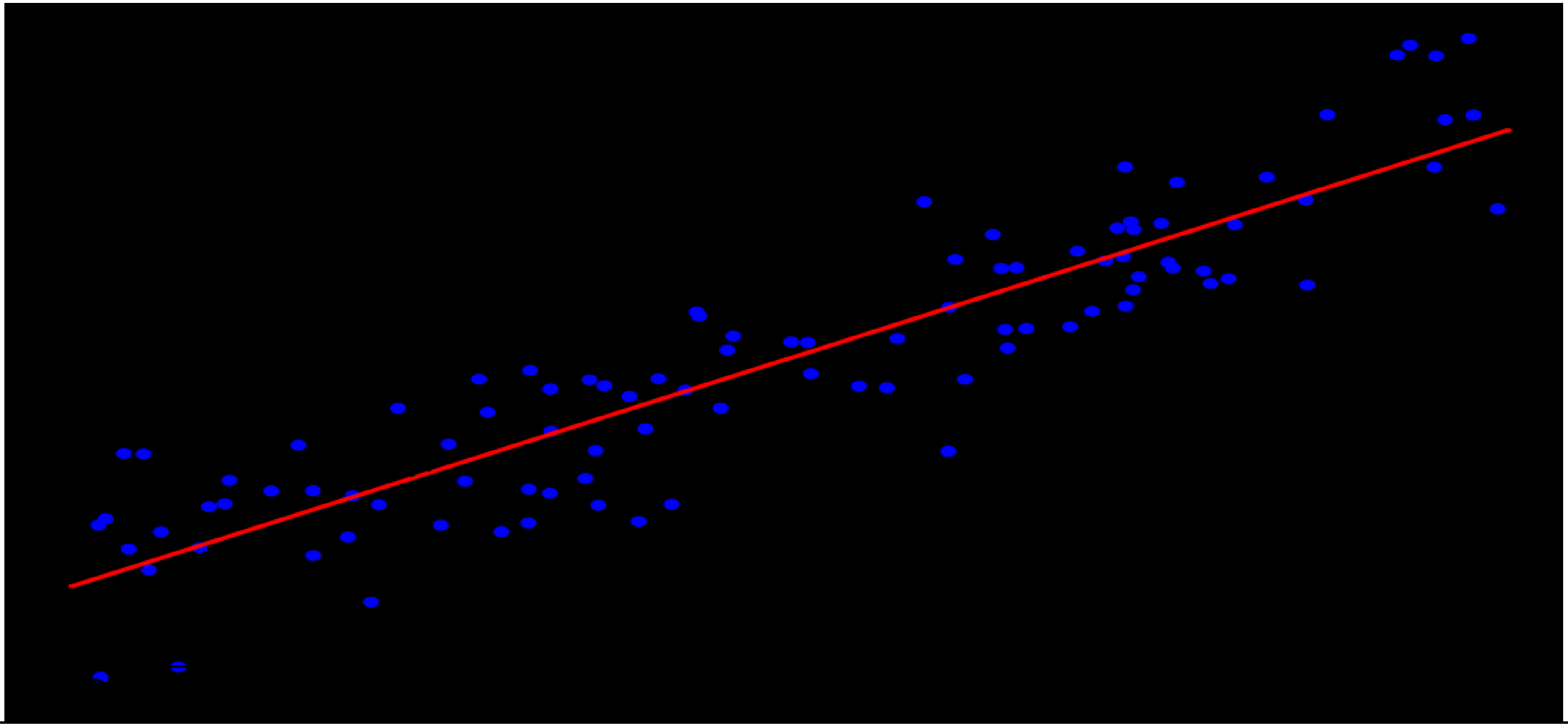
- L(yes) = 2/9 * 3/9 * 3/9 * 3/9 = 0.0082
- L(no) = 3/5 * 1/5 * 4/5 * 3/5 = 0.0577

- P(yes) = 0.0082 * 9/14 = 0.0053
- P(no) = 0.0577 * 5/14 = 0.0206

- The decision would be: NO.

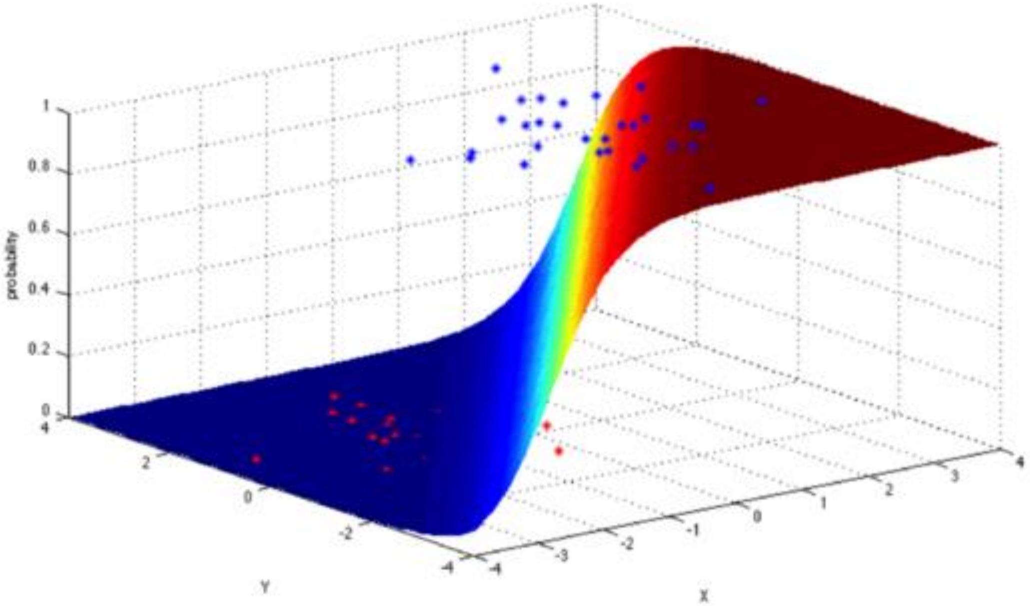
Linear Regression (supervised)

- **Draw a line, and then for each of the data points, measure the vertical distance between the point and the line, and add these up; the fitted line would be the one where this sum of distances is as small as possible.**
- **Use case:**
  – Housing prices



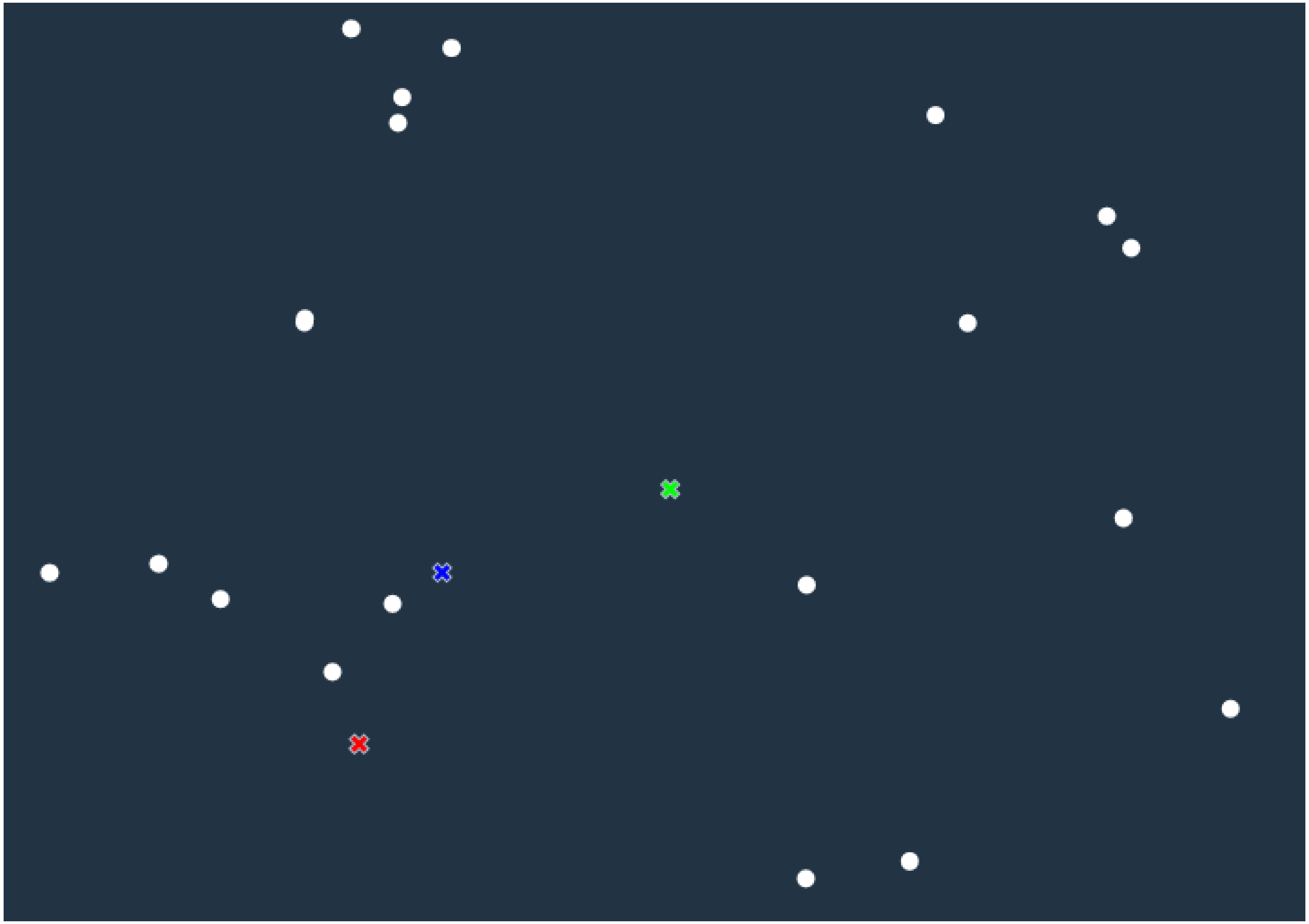
Logistic Regression (supervised)

- **Logistic regression is a powerful statistical way of modeling a binomial outcome with one or more explanatory variables. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.**



Clustering – K-means method

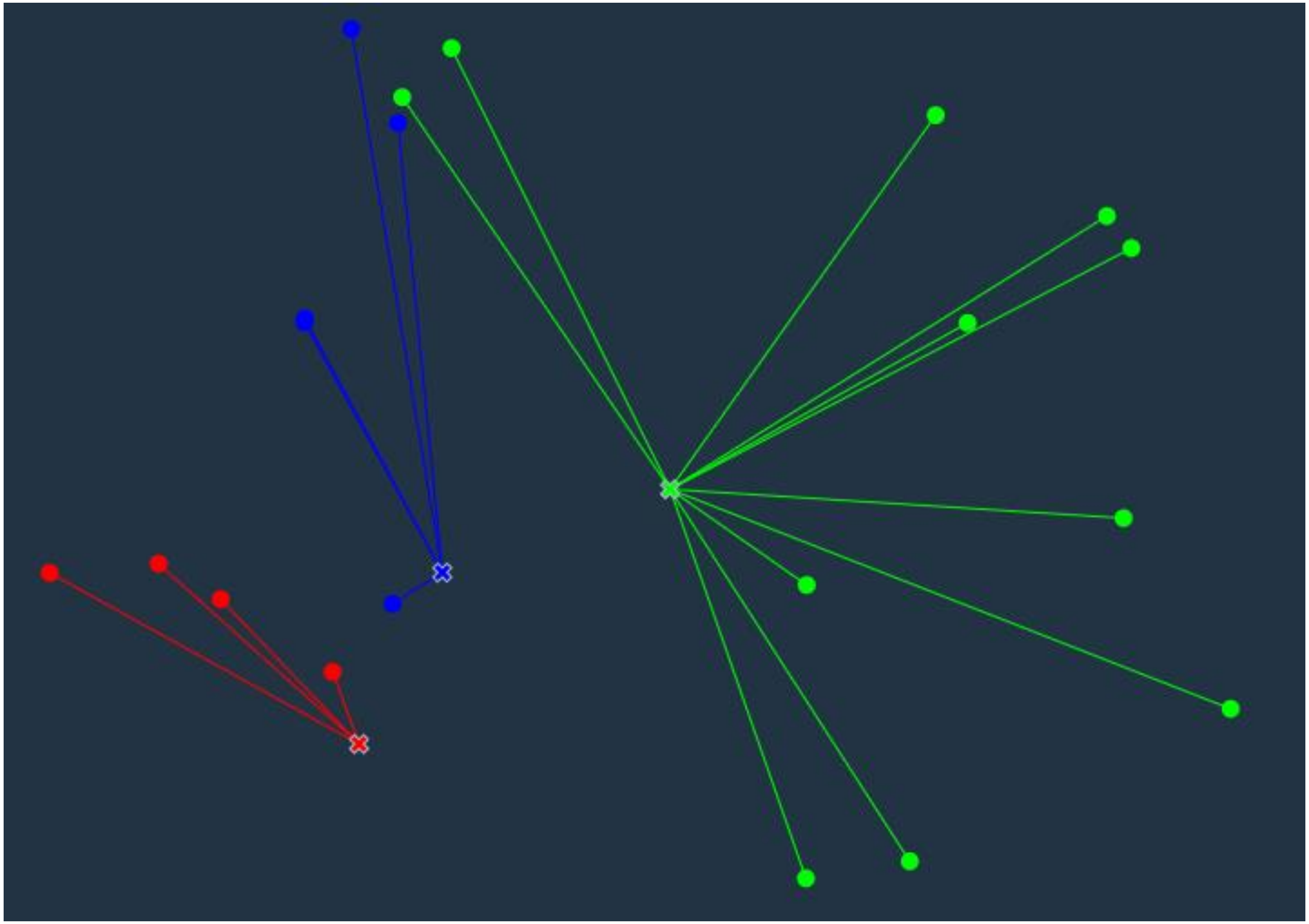
Modeling



Start with 20 data points and 3 clusters

Clustering – K-means method

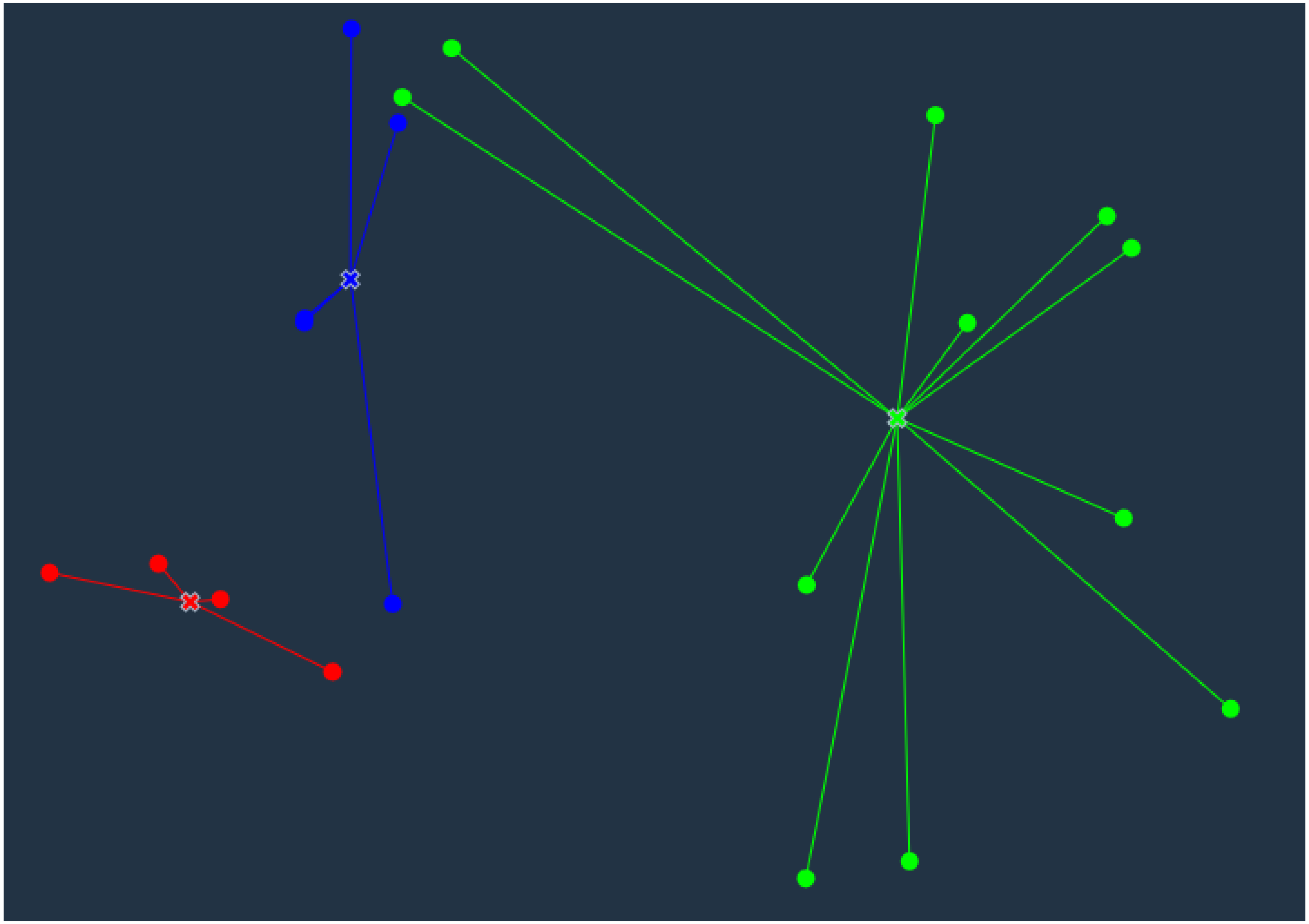
Modeling



Assign each data point to the nearest cluster

Clustering – K-means method

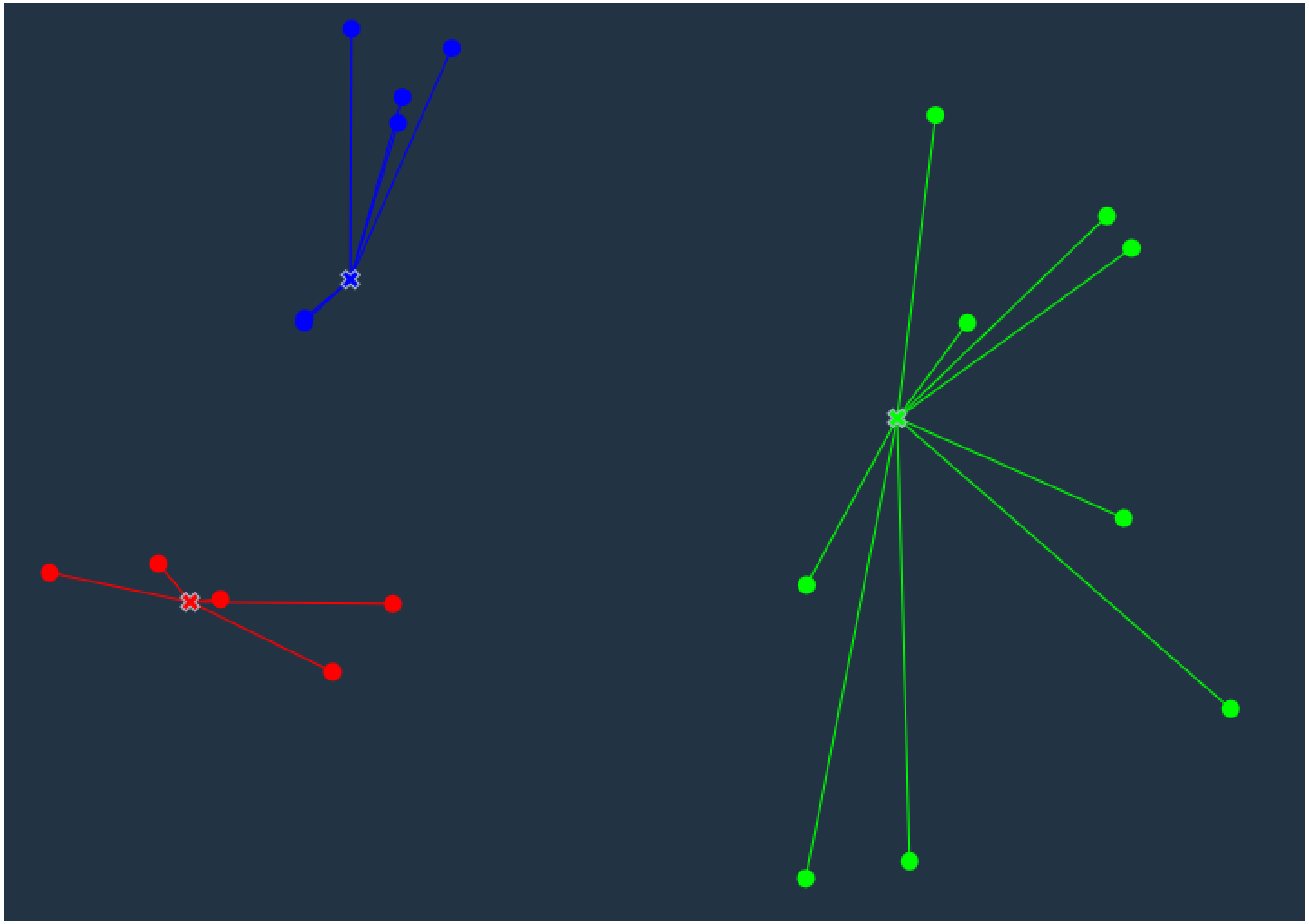
Modeling



Calculate centroids of new clusters

Clustering – K-means method

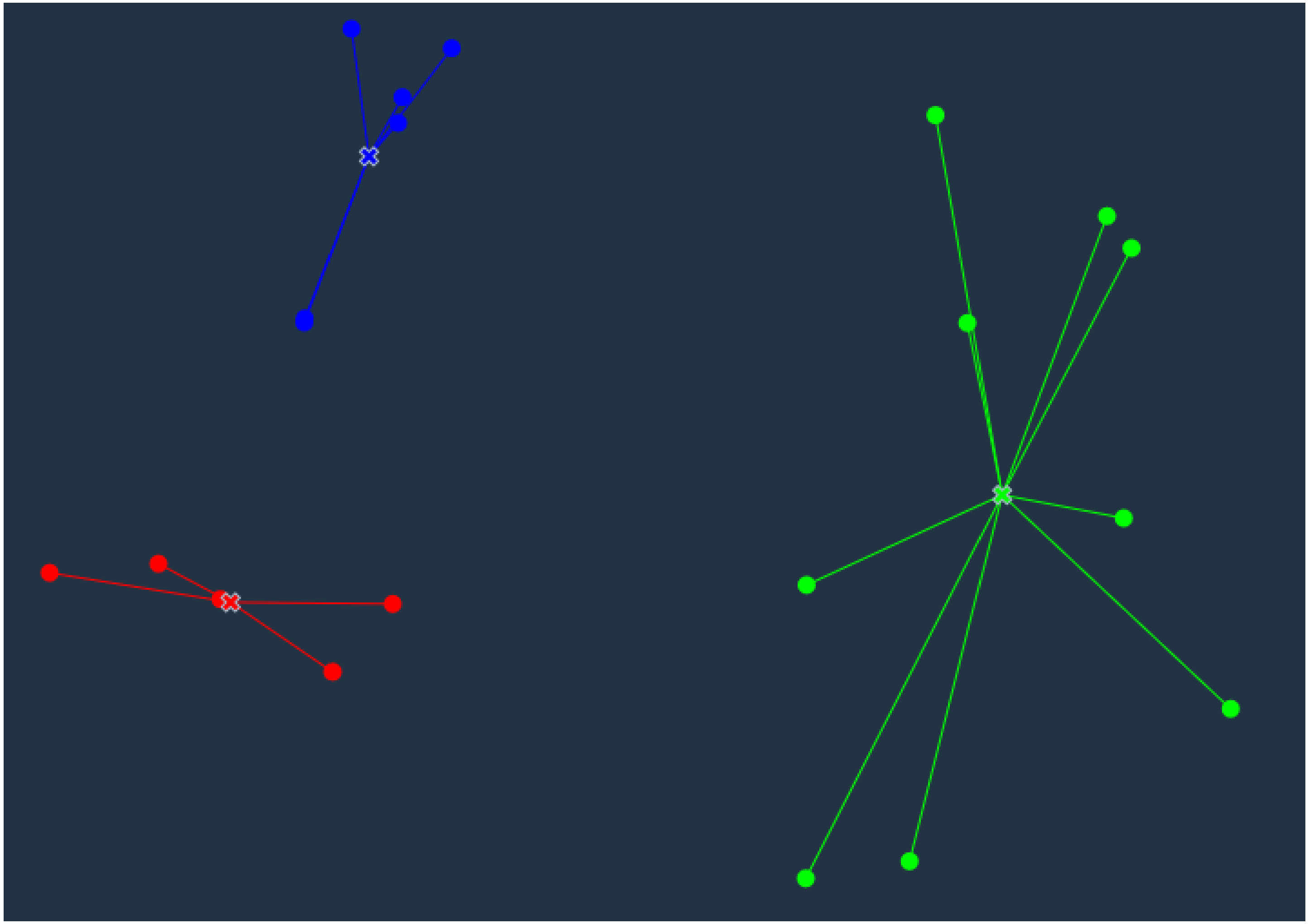
Modeling



Assign each data point to the nearest cluster

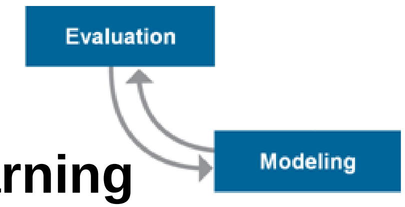
Clustering – K-means method

Modeling



Calculate centroids of new clusters…until convergence

Training, testing, & validation sets



- **During the model development process, supervised learning techniques employ training and testing sets and sometimes a validation set.**
  - Historical data with known outcome (*target*, *class*, *response*, or *dependent variable*)
  - Source data randomly split or sampled… mutually exclusive records
- **Why?**
  - Training set ➔ build the model (**iterative**)
  - Validation set ➔ tune the parameters & variables during model building (**iterative**)
    - Assess model quality during training process
    - Avoid overfitting the model to the training set
  - Testing set ➔ estimate accuracy or error rate of model (**once**)
    - Assess model's expected performance when applied to new data

Model Evaluation: Confusion Matrix

Confusion matrix is more useful measure than simply using prediction accuracy

– Provides a better visualization of the performance of the algorithm
– Examine the count of each of these boxes

Predicted

Actual

	Has Disease	No Disease
Has Disease	true positive (tp) ✓	false negative (fn) No Treatment
No Disease	false positive (fp) Unnecessary Treatment	true negative (tn) ✓

Precision = tp/(tp + fp)   Recall = sensitivity= True Positive Rate tp/(tp + fn)

FPR = fp/(fp + tn) = 1 – specificity   ROC = plot of TPR/FPR at different thresholds

Model Evaluation

- When you are building a classifier, it is important to understand the PREVALANCE of the condition that you are building a model for,
  i.e. how common or uncommon this condition effectively is…

- Imagine you are working towards building a classifier for some medical condition and your training and testing data sets yield the following model

	Test positive	Test negative
Disease (100)	95 (True Positive)	5 (False Negative)
Normal (100)	5 (False Positive)	95 (True Negative)

- With 95% sensitivity & specificity, this sounds like a great test…

Model Evaluation

- **What truly matters to the users of your new model / test (doctors, bankers, practitioners) is the PREDICTIVE VALUE of the test:**
  - If the test is positive, then what is the actual chance of being sick?
  - Is it 95% ?

- **Let's run the test on a population of 1,000,000 where 1% individuals (10,000) are actually suffering from this condition:**

	Test positive	**Test negative**
Disease (10000)	9500 (95% True Positive)	500 (5% False Negative)
Normal (990000)	49500 (5% False Positive)	940500 (95% True Negative)

Model Evaluation

	Test positive	Test negative
Disease (10000)	9500 (95% True Positive)	500 (5% False Negative)
Normal (990000)	49500 (5% False Positive)	940500 (95% True Negative)

- **Probability of being sick if the test is positive:**
  - (# of people truly sick) / # positive result tests
  - 9500 / (49500 + 9500) = **16.1%**
  - What is happening here:
    - The condition is RARE and the 5% FALSE POSITIVES are still way higher in numbers than the true positives.

- **Data analysis of the prevalence of the condition tells us that a test with 99% or higher sensitivity / specificity would be needed.**

Spark ML Pipeline Terminology

Spark ML standardizes APIs for machine learning algorithms to make it easier to combine multiple algorithms into a single pipeline, or workflow

- **DataFrame**: Spark ML uses DataFrame from Spark SQL as an ML dataset, which can hold a variety of data types

- **Transformer**: A Transformer is an algorithm which can transform one DataFrame into another DataFrame

- **Estimator**: An Estimator is an algorithm which can be fit on a DataFrame to produce a Transformer

- **Pipeline**: A Pipeline chains multiple Transformers and Estimators together in a sequence to specify an ML workflow

- **Parameter**: All Transformers and Estimators share a common API for specifying parameters

Questions to ask

- What are your goals?
- What are the criteria for success?
- Do you need labeled ($$) data?
- Look at your data. Clean it. What features are pertinent. How much do you have?
- How quickly does a new instance need to be classified? (online/batch)
- Do you need to scale?
- What resources do you have? Memory, compute nodes, GPU
- Would using ensembles help?
- When the goals are met – stop.

Data Scientist Issues

- **Rigid toolset**
  - Have to choose one and only one approach
  - Cannot easily connect all of the capabilities needed
  - Difficult to navigate between the various tools used

- **Fragmented and time consuming**
  - Using multiple disjointed environments
  - Separate on-ramp/community for each tool/environment
  - Does not have meta data or data lineage

- **Analytical Silo**
  - Difficult to maintain and version control project assets
  - Limited means of collaborating with team
  - Results are difficult to share

Data Science Experience

Brings together popular Data Science **Open Source tools** with IBM value-add functionalities coupled with **community and social** features

Learn	Create	Collaborate
Built-in learning to get started or go the distance with advanced tutorials	The best of open source and IBM value-add to create state-of-the-art data products	Community and social features that provide meaningful collaboration

External URL: http://datascience.ibm.com

Core Attributes of the Data Science Experience

IBM Data Science Experience

Community

- Find tutorials and datasets
- Read articles and papers
- Connect with Data Scientists
- Share comments
- Copy and share notebooks

Open Source

- Code in Scala/Python/R/SQL
- Jupyter Notebooks
- RStudio IDE and Shiny
- Apache Spark
- Your favorite libraries

IBM Added Value

- IBM Machine Learning
- SPSS Modeler Canvas
- Prescriptive Analytics - DOcplexcloud
- Projects and Version Control
- Managed Spark Service

Powered by IBM **Watson Data Platform**

* Closed beta

Docs, Forums, Blogs and Ideas

- Online documentation for DSX, DSX Local and DSX Desktop
- DSX discussion forum on Stack Overflow
- Blog posts from IBM Developers
- Give feedback on DSX to IBM for new features

Helpful links

Docs

Find the information you need. Watch videos of key tasks.

Discussion forum

Stack Overflow is a community of 6.9 million programmers just like you, helping each other. Join the conversation on Data Science Experience.

Blog

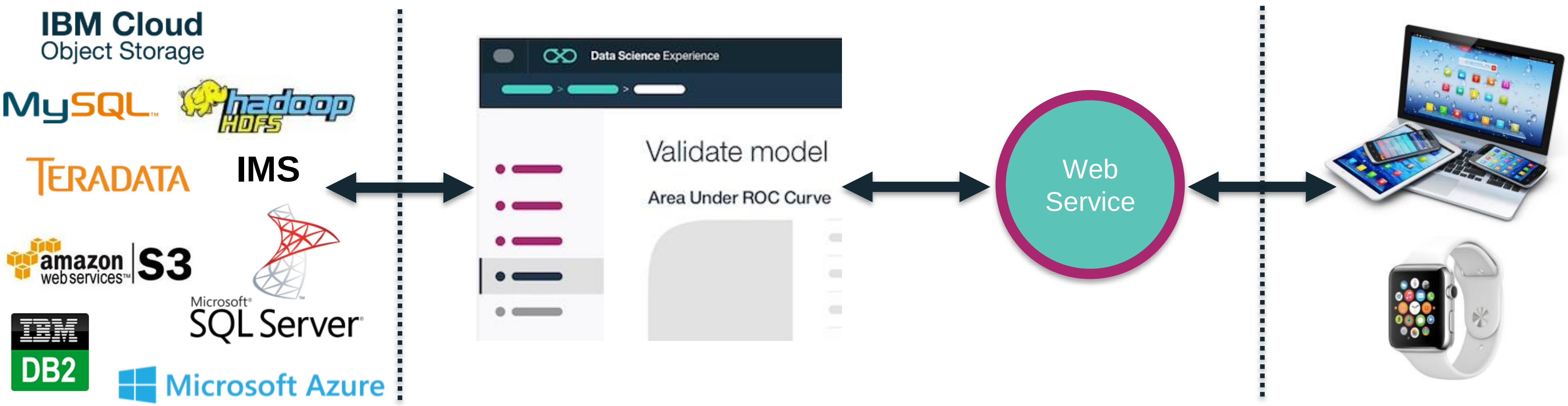
Read and follow our blog to keep up with the latest updates about Data Science Experience.

Got ideas?

Have feedback on Data Science Experience? Submit your ideas in our product forum or vote on ideas submitted by others.

Operationalize insights with IBM Machine Learning

IBM Machine Learning



Data Access:

- Easily connect to Behind-the-Firewall and Public Cloud Data
- Catalogued and Governed Controls through Watson Data Platform

Creating Models:

- Single UI and API for creating ML Models on various Runtimes
- Auto-Modeling and Hyperparameter Optimization

Web Service:

- Real-time, Streaming, and Batch Deployment
- Continuous Monitoring and Feedback Loop

Intelligent Apps:

- Integrate ML models with apps, websites, etc.
- Continuously Improve and Adapt with Self-Learning

Backup