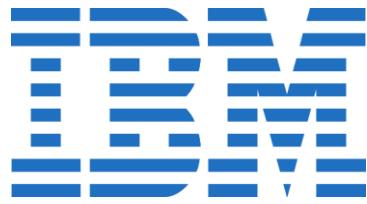


# Hands on Introduction to Machine Learning with IBM Watson Studio



Power of data. Simplicity of design. Speed of innovation.

**Bernie Beekman**  
**Michael Cronk**  
**Aaron McKay**  
**James Parry**

# Agenda

8:30am – 9:00am	<b>Breakfast, Socialize</b>
9:00am – 10:30am	<b>Introduction to Machine Learning Presentation</b> <b>Watson Studio</b> <b>Lab Overview</b>
10:30am – 10:45am	<b>Break</b>
10:45am – 11:45am	<b>Lab 1 - Machine Learning with XGBoost</b>
11:45am – 12:30pm	<b>Lab 2 – Continuous Learning with Watson Machine Learning</b>
12:00pm – 1:00pm	<b>Lunch</b>
1:00pm – 2:00pm	<b>Lab 3 – Neural Network Modeling and Deployment</b>
2:00pm – 4:00pm	<b>Lab 4 – WML, SPSS, Data Refinery, Spark ML</b>
4:00pm – 4:30pm	<b>Wrap Up</b>

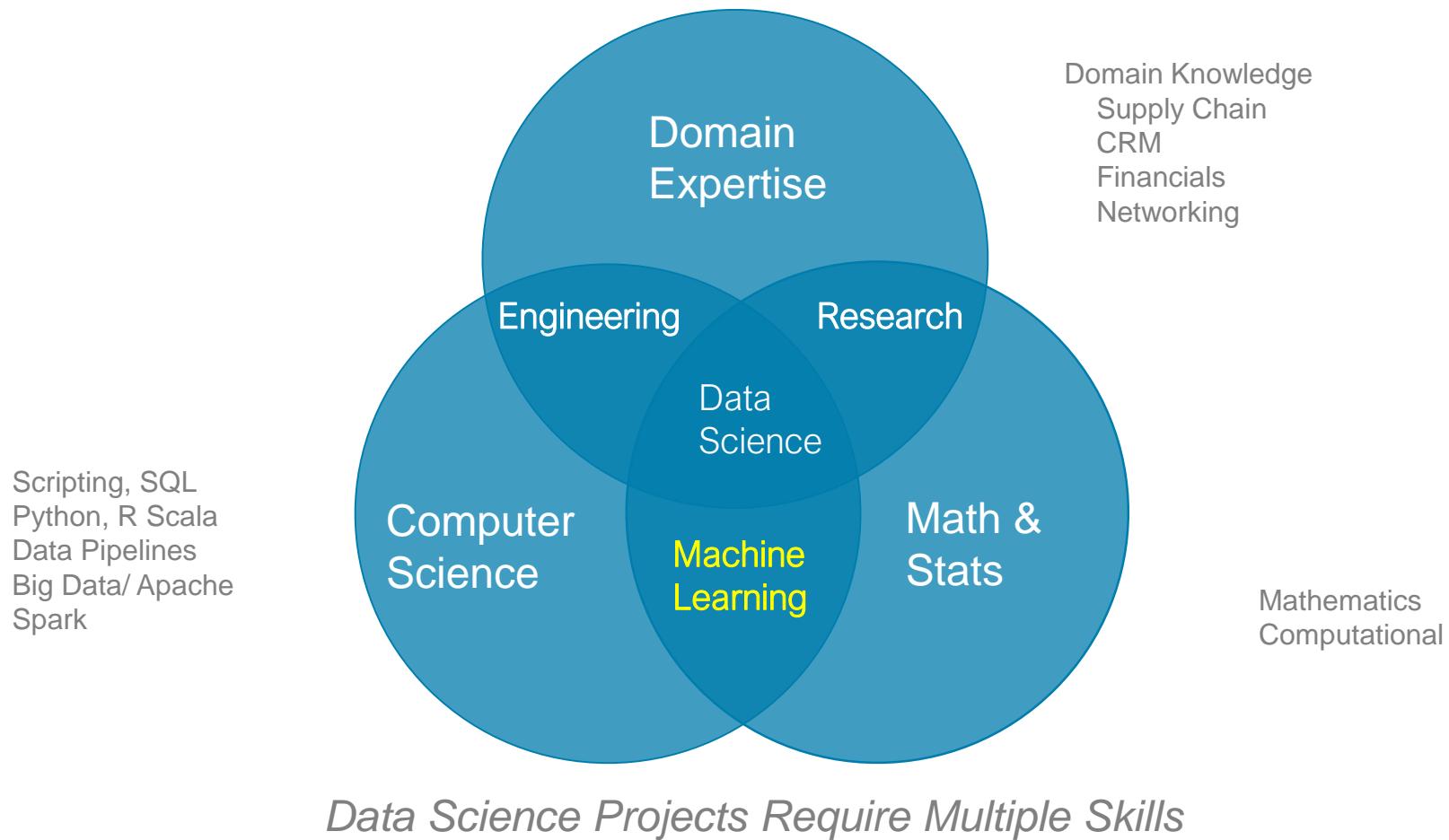
# Presentation Outline

- **Introduction to Machine Learning**
- **Watson Studio**
- **Lab Overview**

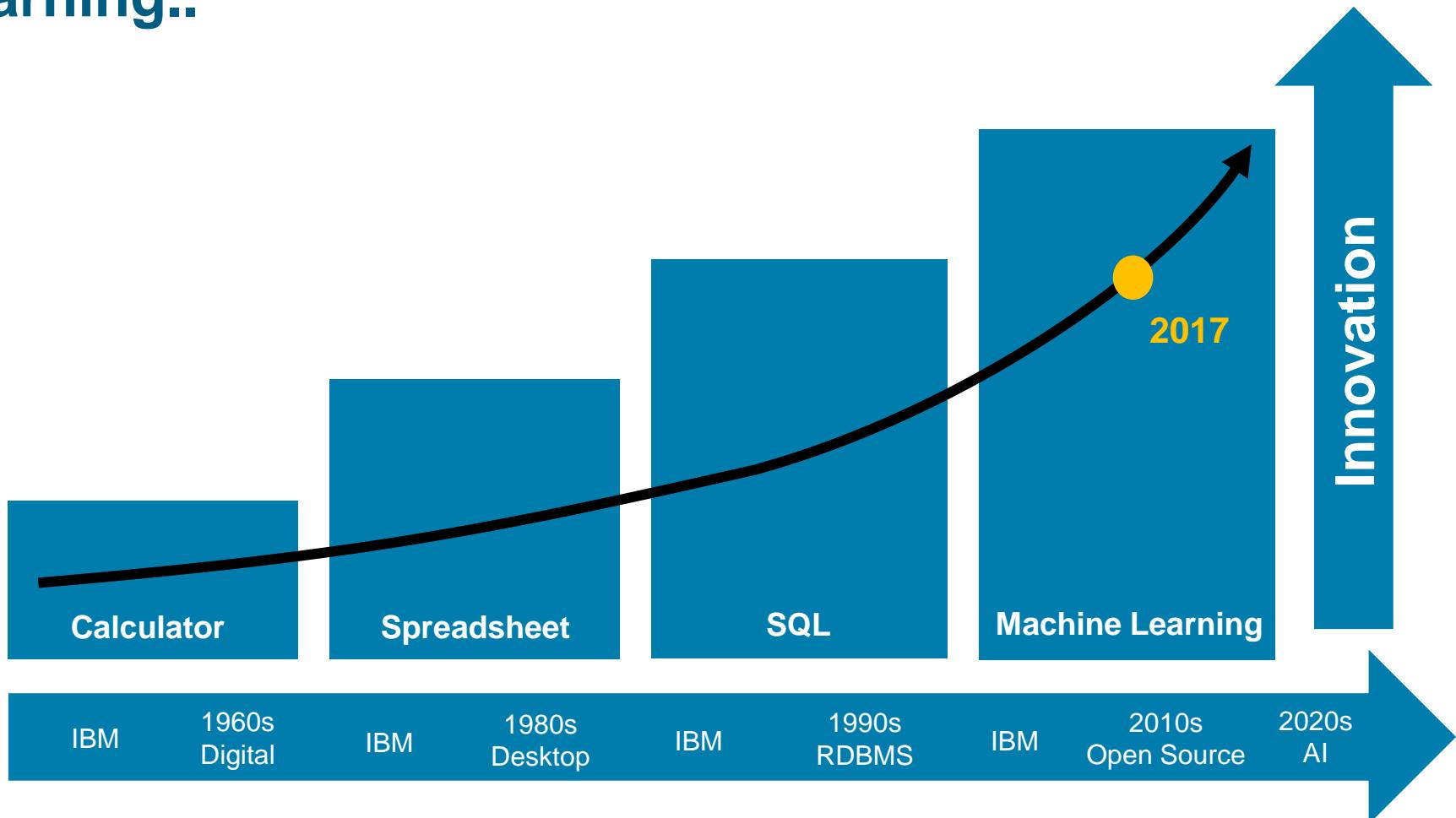
# Introduction to Machine Learning

- Overview 
- Data Science Methodology
- Data Understanding
- Data Preparation
- Categories of Machine Learning
- Learning Challenges
- Machine Learning Algorithms
- Evaluation

# Machine Learning and Data Science....

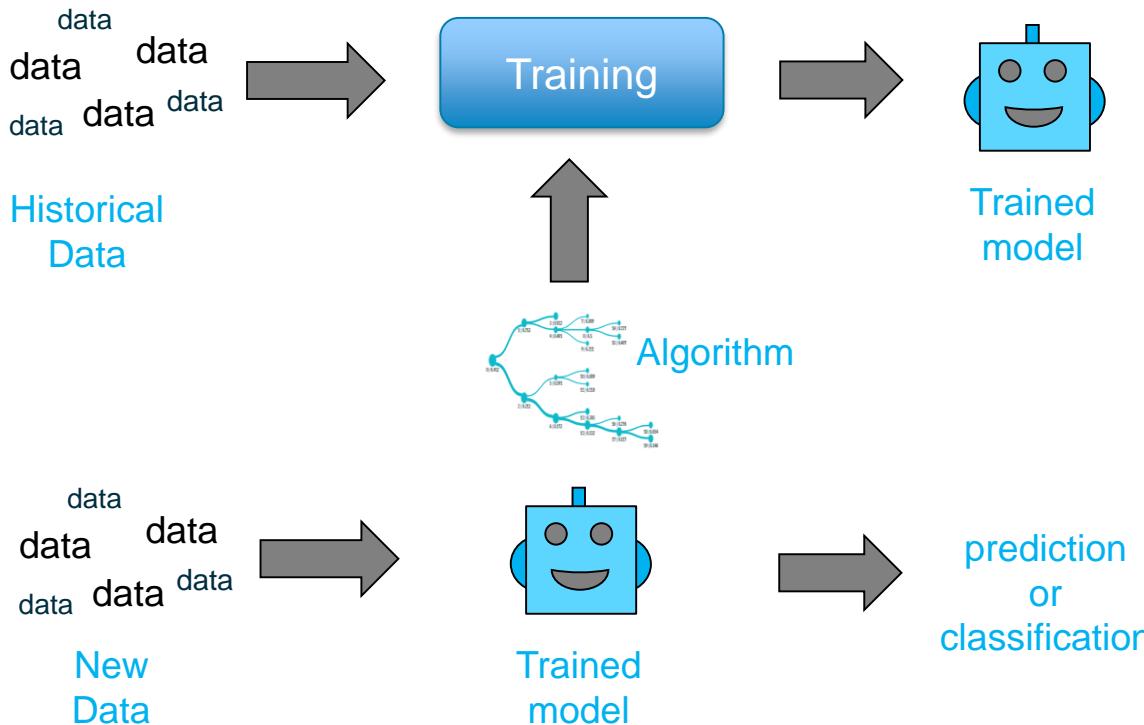


# Future of Data Science is Democratizing Machine Learning..



# But what is Machine Learning?

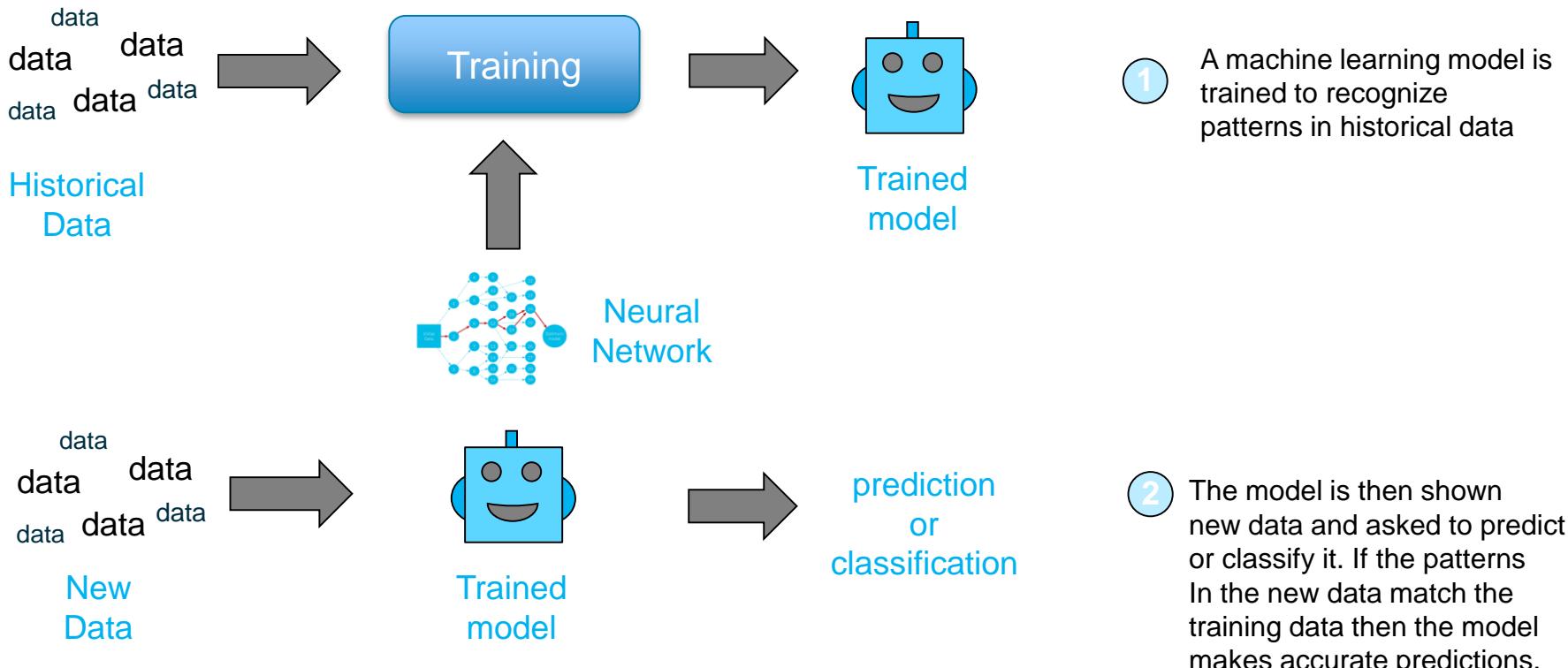
*“Computers that learn without being explicitly programmed”*



- 1 A machine learning model is trained to recognize patterns in historical data
- 2 The model is then shown new data and asked to predict or classify it. If the patterns in the new data match the training data then the model makes accurate predictions.

# But what is Deep Learning?

*“Computers that learn without being explicitly programmed”*

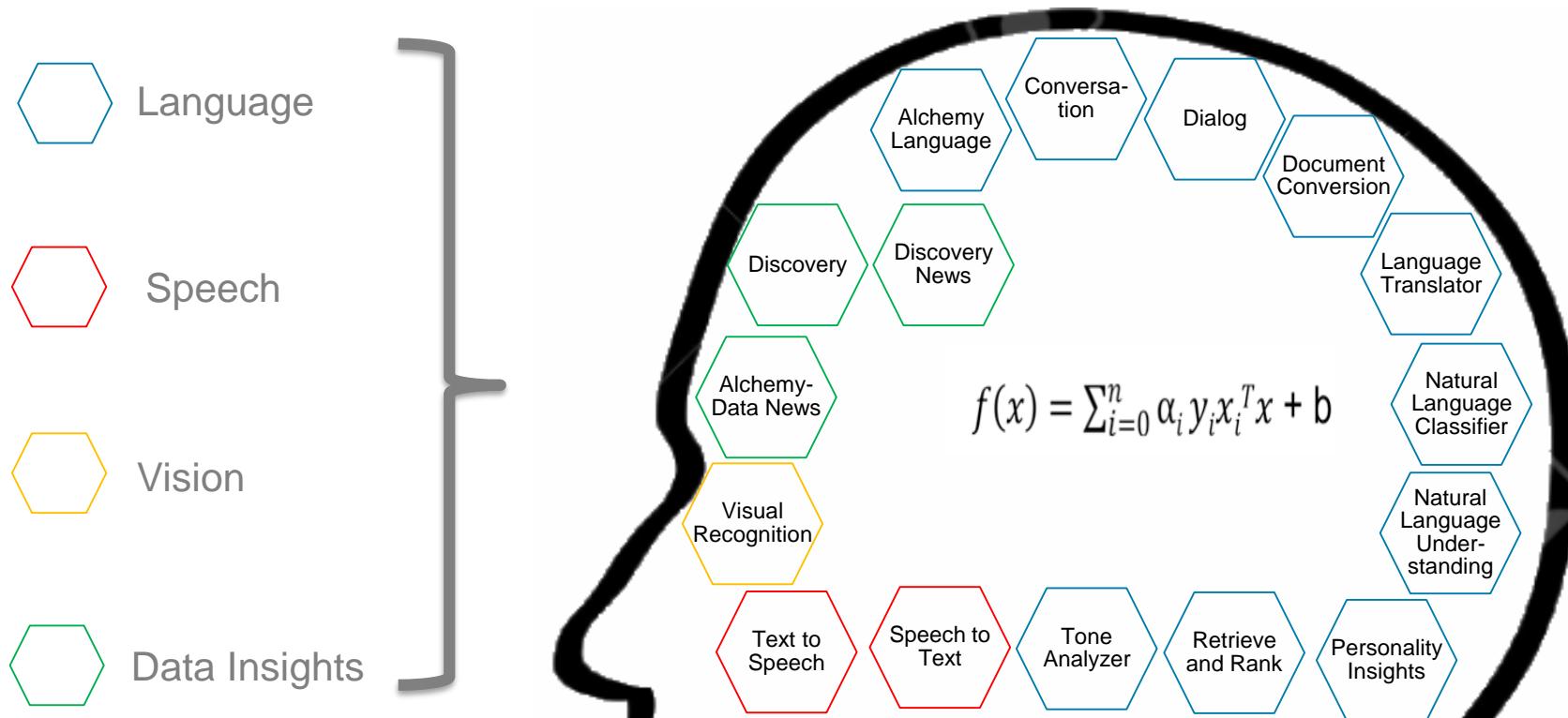


# But what is Artificial Intelligence?

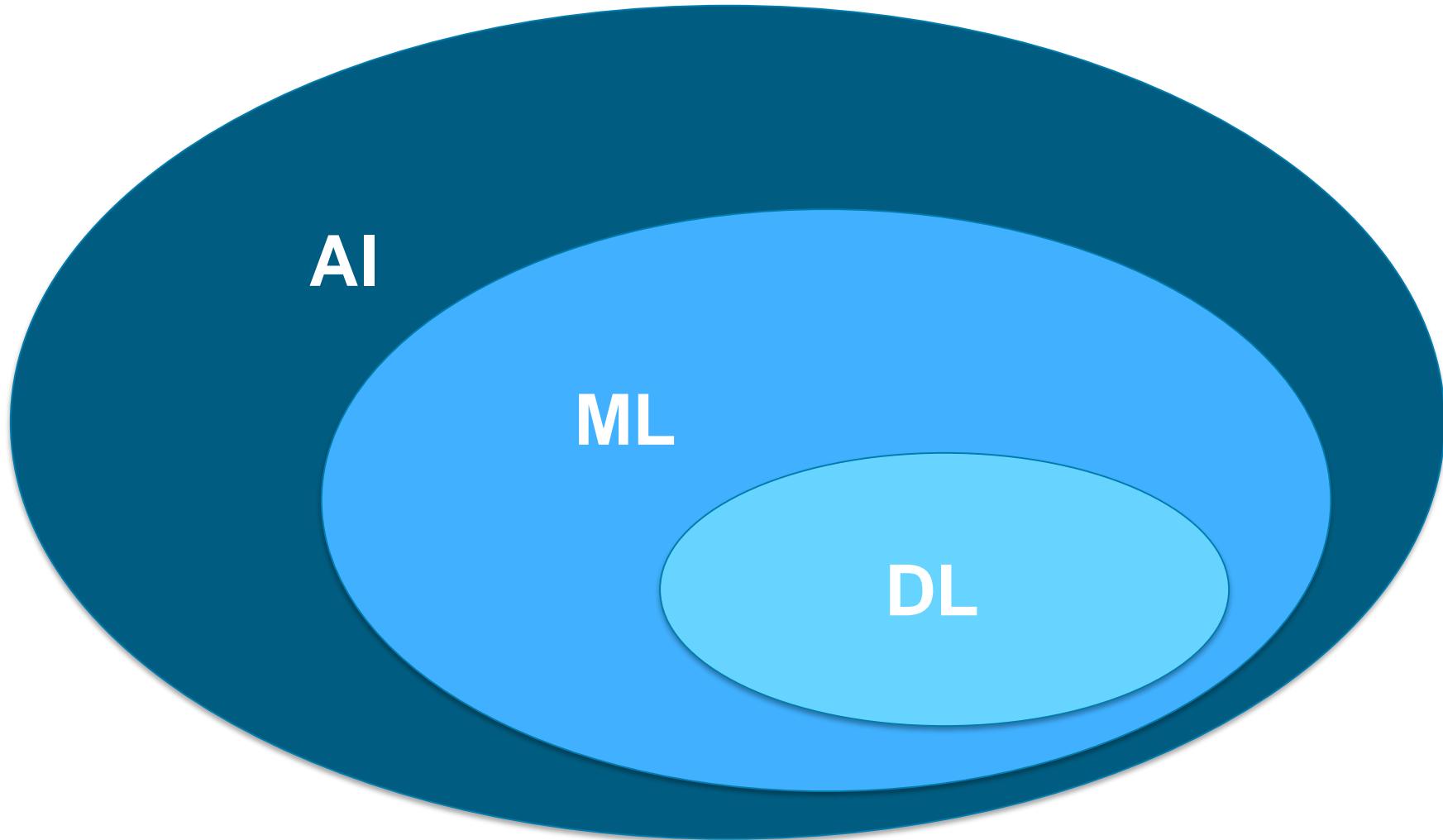
A theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages..

# Machine Learning = Artificial Intelligence???

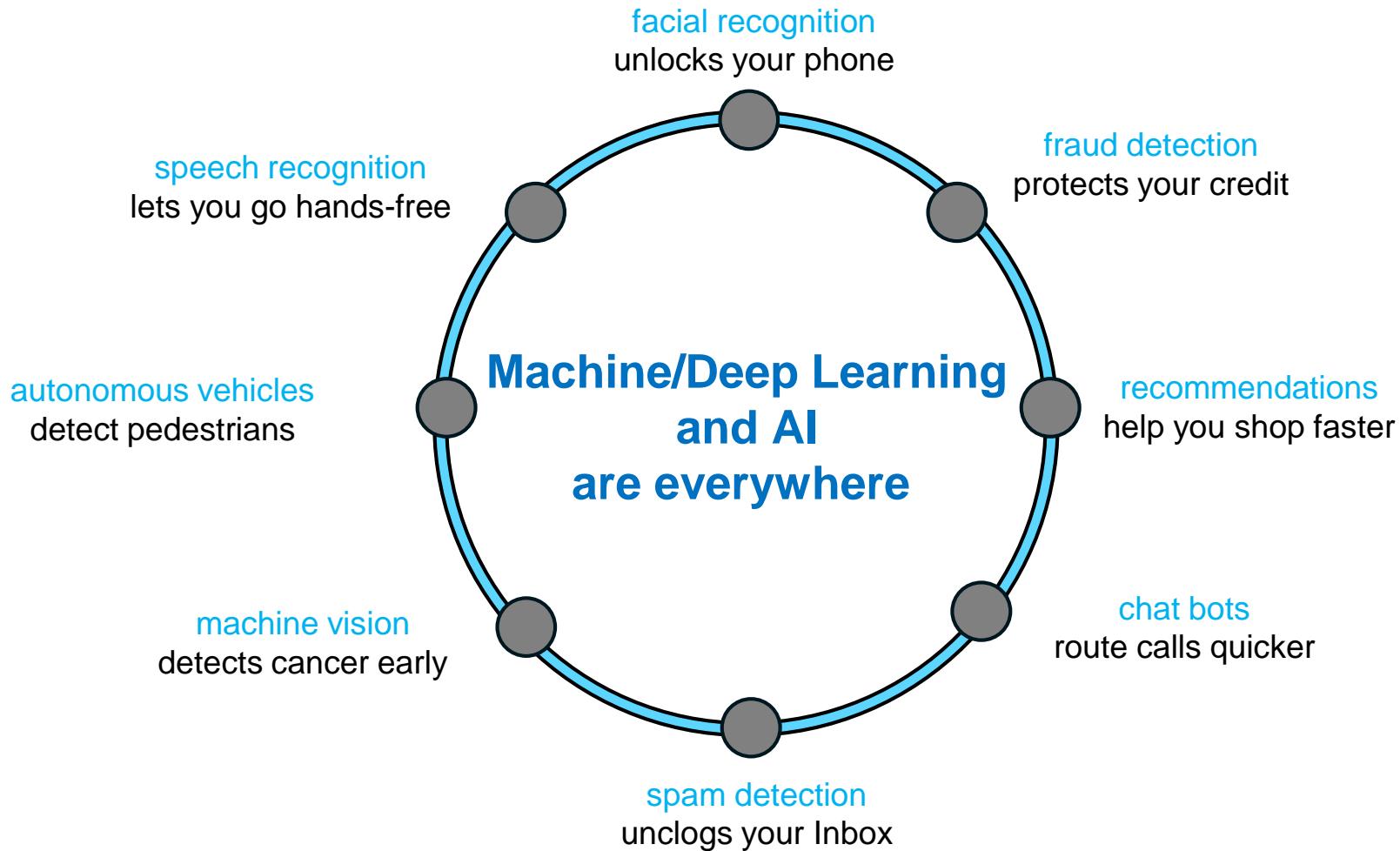
Data + Algorithms = Scored AI Models



# Understanding AI, ML & DL Relationship...



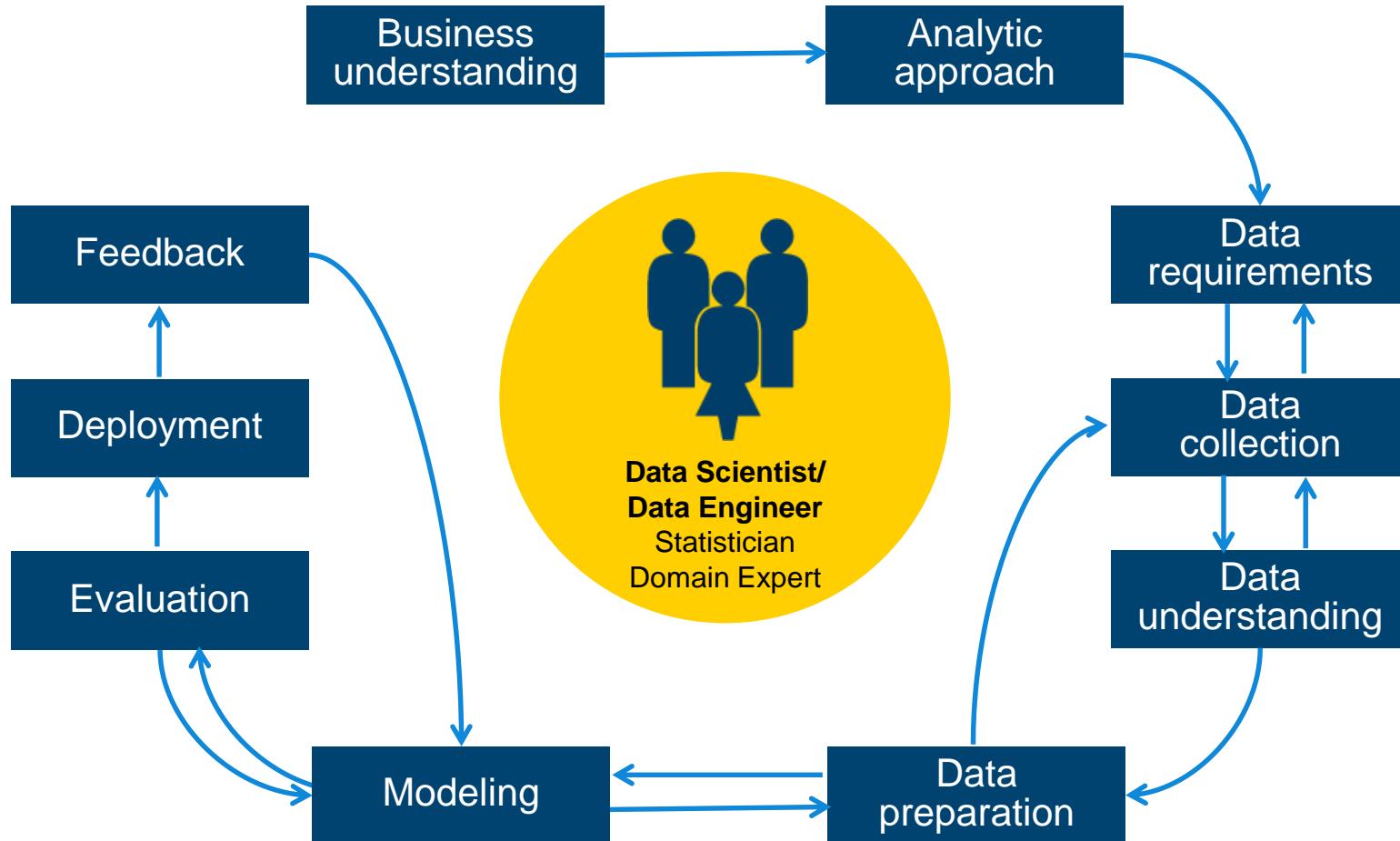
# The future is now



# Introduction to Machine Learning

- Overview
- Data Science Methodology 
- Data Understanding
- Data Preparation
- Categories of Machine Learning
- Learning Challenges
- Machine Learning Algorithms
- Model Evaluation

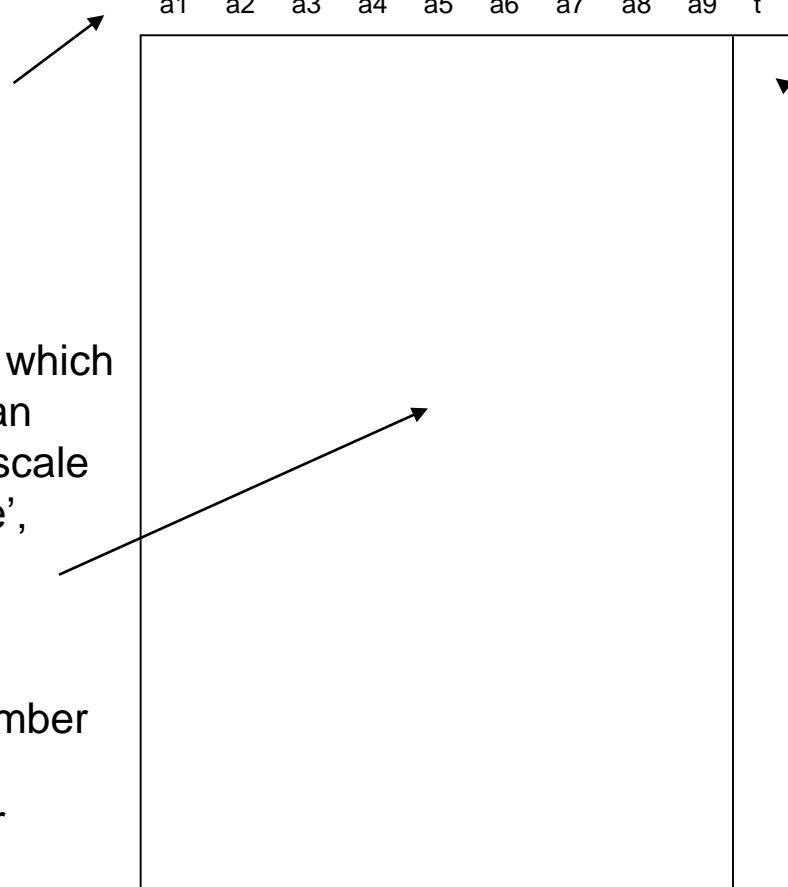
# Data Science Methodology



# Matrix for Machine Learning

## Known as:

- Attributes
- Features
- Predictor variables
- Explanatory variables



## Scale variables:

- Continuous variables, which can be measured on an interval scale or ratio scale
- 'Weight', 'Temperature', 'Salary', etc...

## Known as:

- Label
  - Target variable
  - Dependent variable
- Scale or Categorical

## Categorical variables:

- Data with a limited number of distinct values or categories (nominal or ordinal)
- 'Hair color', 'Gender', 'Grape varieties', etc...

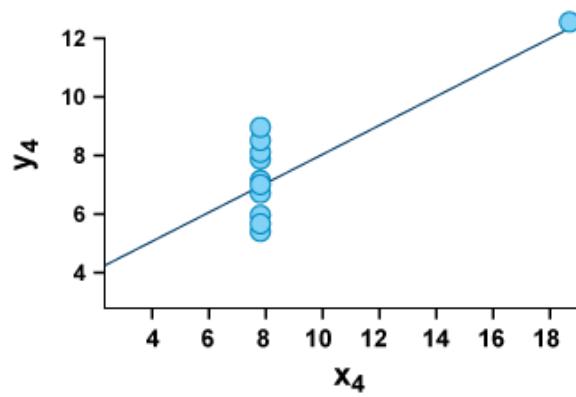
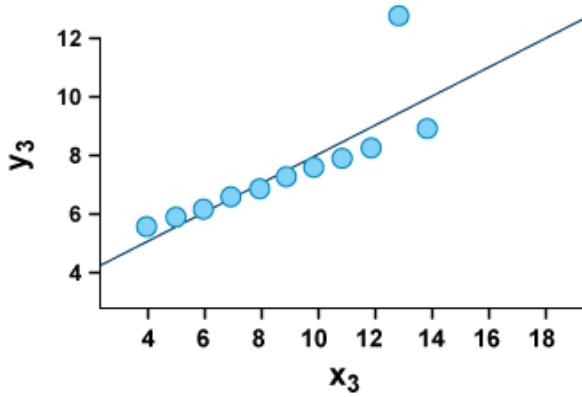
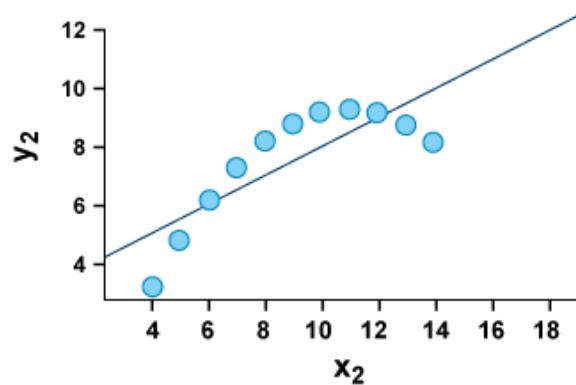
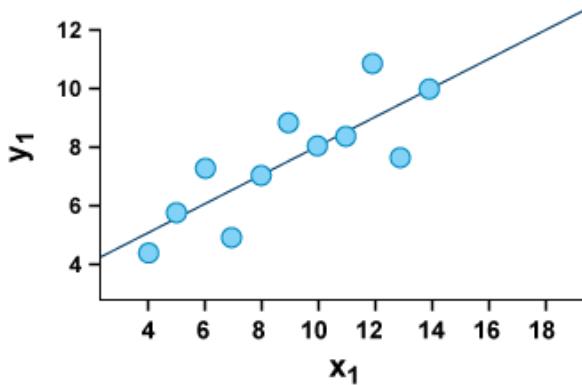
# Introduction to Machine Learning

- Overview
- Data Science Methodology
- Data Understanding 
- Data Preparation
- Categories of Machine Learning
- Learning Challenges
- Machine Learning Algorithms
- Model Evaluation

# 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

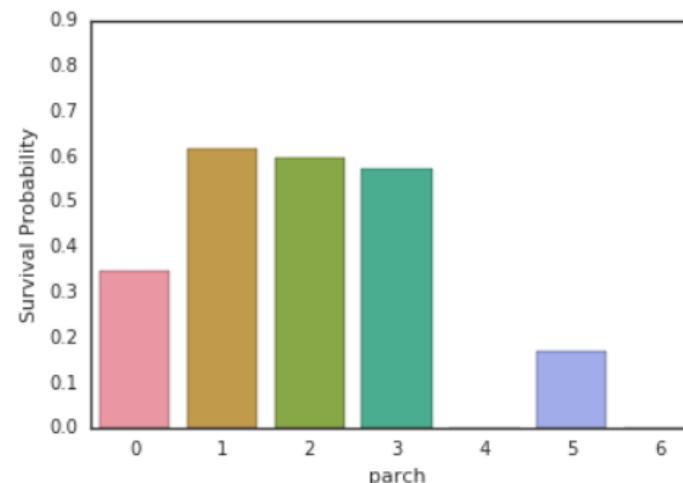
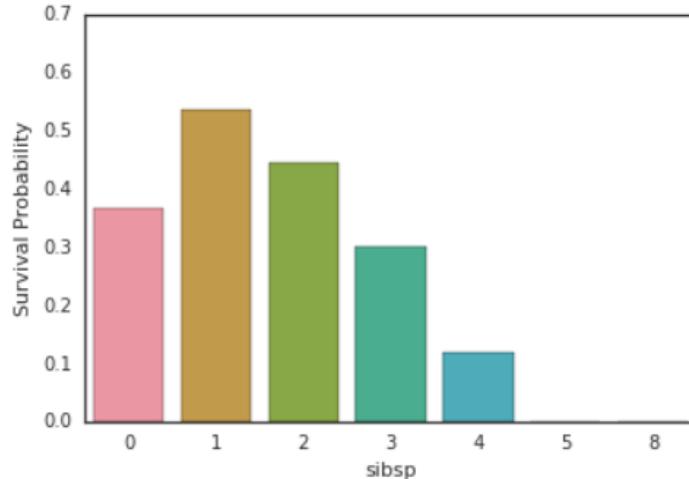
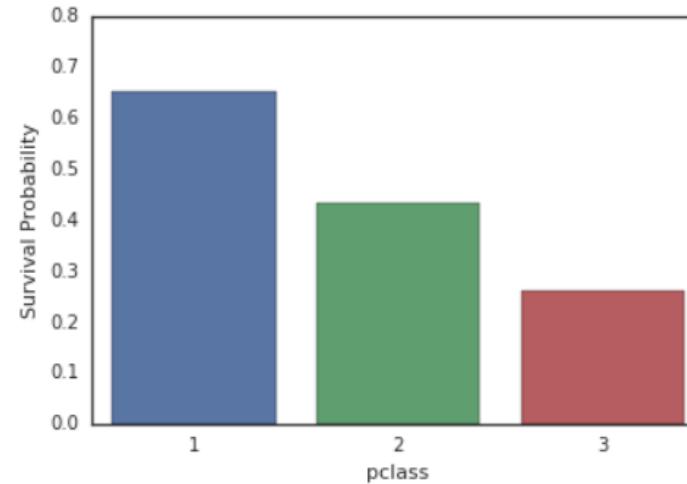
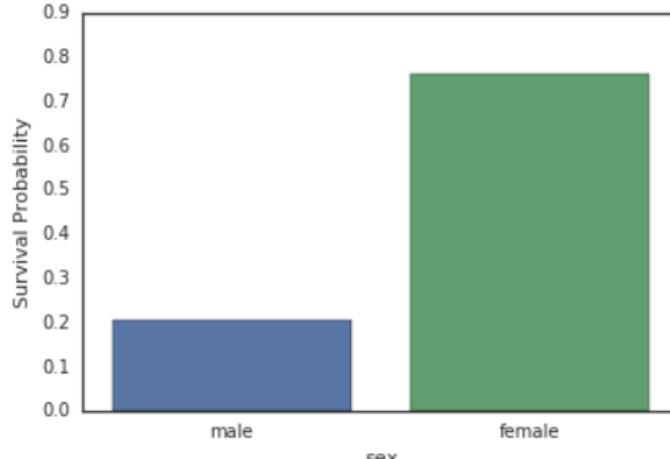
As shown the linear regression lines are approx.  $y=3.00+0.500x$ .

## ■ 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

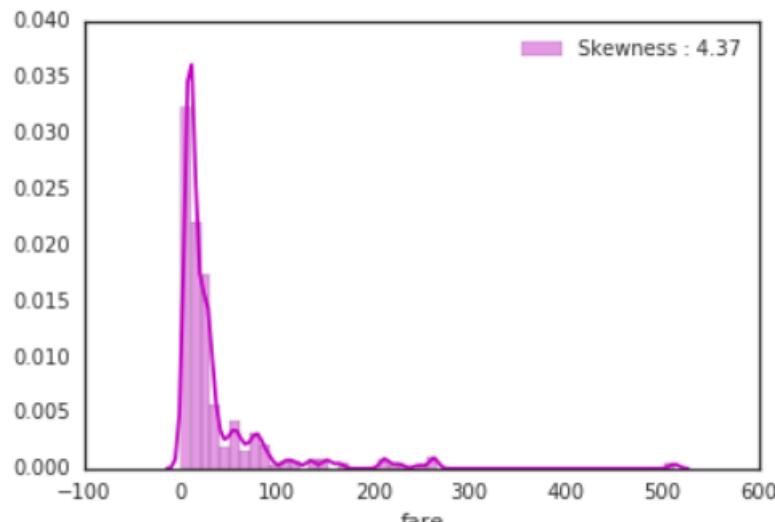
# Data Understanding: Visualizations

- Titanic Data
- Univariate Relationships

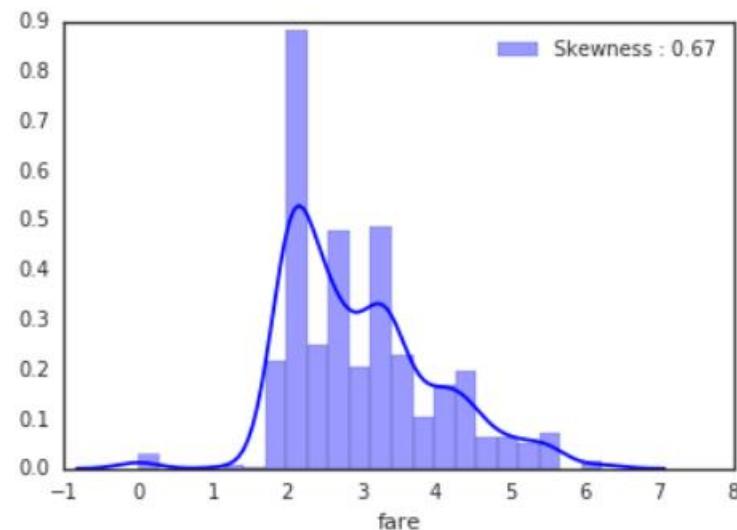


# Data Understanding: Visualizations

- Titanic Data
- Skewed Data



Original Data



After Log Transform

# Introduction to Machine Learning

- Overview
- Data Science Methodology
- Data Understanding
- Data Preparation 
- Categories of Machine Learning
- Learning Challenges
- Machine Learning Algorithms
- Model Evaluation

# Data Preparation

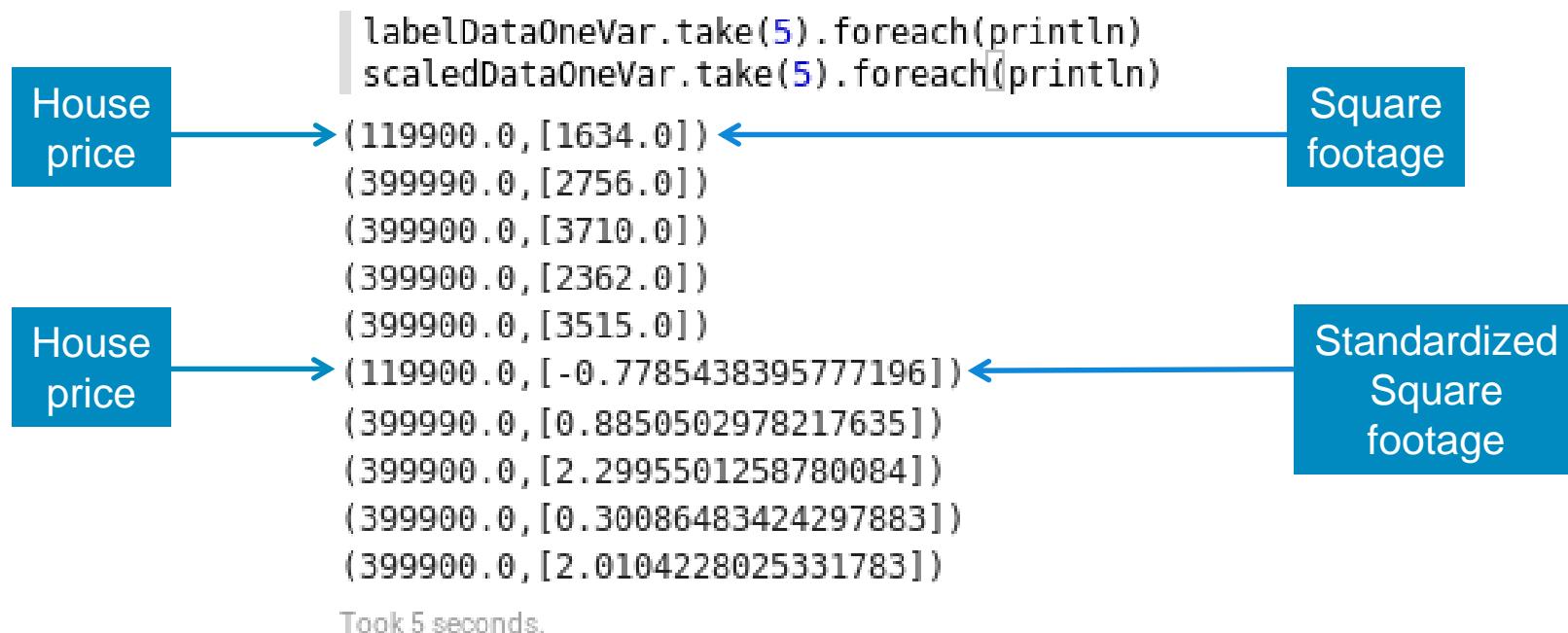
- **Data preparation can be very time consuming depending on:**
  - The state of the original data
    - Data is typically collected in a “human” friendly format
  - The desired final state of the data (as required by the machine learning models and algorithms)
    - The desired final state is typically 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



# 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: **fish**ing, **fishe**d, **fisher** → fish
  - Examples methods of dimensionality reduction
    - Principal Component Analysis
    - Singular Value Decomposition
    - Autoencoders

# Introduction to Machine Learning

- Overview
- Data Science Methodology
- Data Understanding
- Data Preparation
- Categories of Machine Learning 
- Learning Challenges
- Machine Learning Algorithms
- Model Evaluation

# 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.**

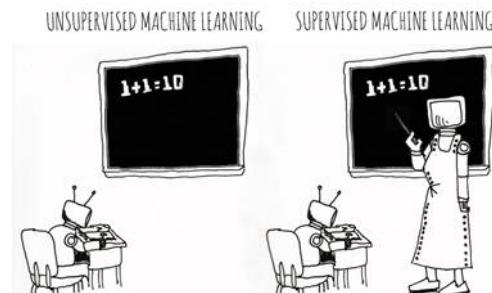
# Categories of Machine Learning

## ▪ 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

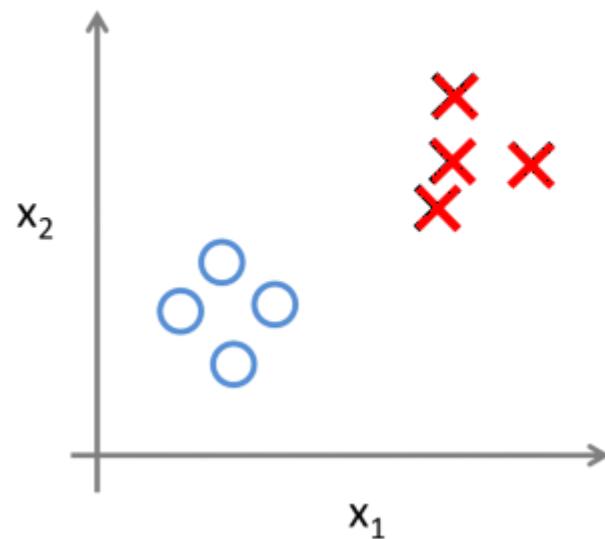
## ▪ Unsupervised learning

- 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)

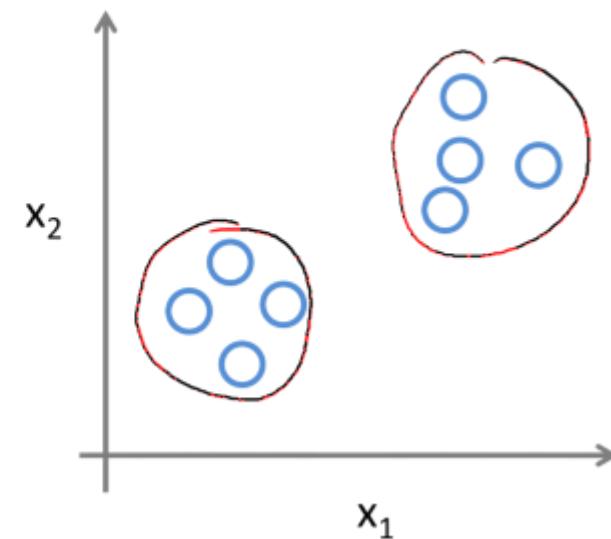


# Supervised vs. Unsupervised Learning

Supervised Learning



Unsupervised Learning



# Categories of Machine Learning

Technique	Usage	Algorithms
Classification (or prediction)	<ul style="list-style-type: none"><li>Used to predict group membership (e.g., will this employee leave?) or a number (e.g., how many widgets will I sell?)</li></ul>	<ul style="list-style-type: none"><li>Decision Trees</li><li>Logistic Regression</li><li>Random Forests</li><li><b>Naïve Bayes</b></li><li>Linear Regression</li><li>Lasso Regression</li><li>etc</li></ul>
Segmentation	<ul style="list-style-type: none"><li>Used to classify data points into groups that are internally homogenous and externally heterogeneous.</li><li>Identify cases that are unusual</li></ul>	<ul style="list-style-type: none"><li>K-means</li><li>Gaussian Mixture</li><li>Latent Dirichlet allocation</li><li>etc</li></ul>
Association	<ul style="list-style-type: none"><li>Used to find events that occur together or in a sequence (e.g., market basket)</li></ul>	<ul style="list-style-type: none"><li>FP Growth</li><li>etc</li></ul>

# Categories of Machine Learning



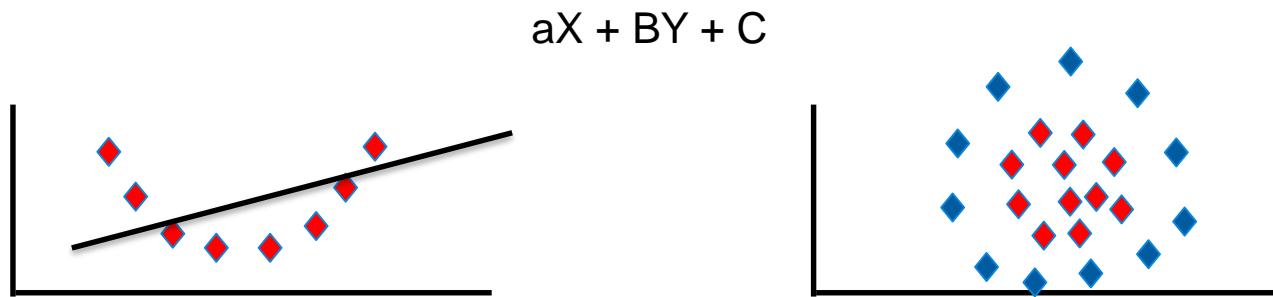
# Introduction to Machine Learning

- Overview
- Data Science Methodology
- Data Understanding
- Data Preparation
- Categories of Machine Learning
- Learning Challenges 
- Machine Learning Algorithms
- Model Evaluation

# Learning challenges

- **Under fitting:**

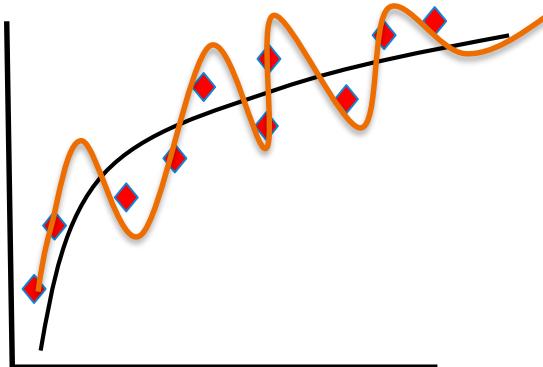
- Not knowing enough “basic” concepts, i.e. not being well-equipped enough to tackle learning at hand:
  - You can't study calculus without knowing some algebra.
  - You can't learn playing hockey without knowing how to skate.
  - You can't learn polo without knowing how to ride.
- This can lead to under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



# Learning challenges

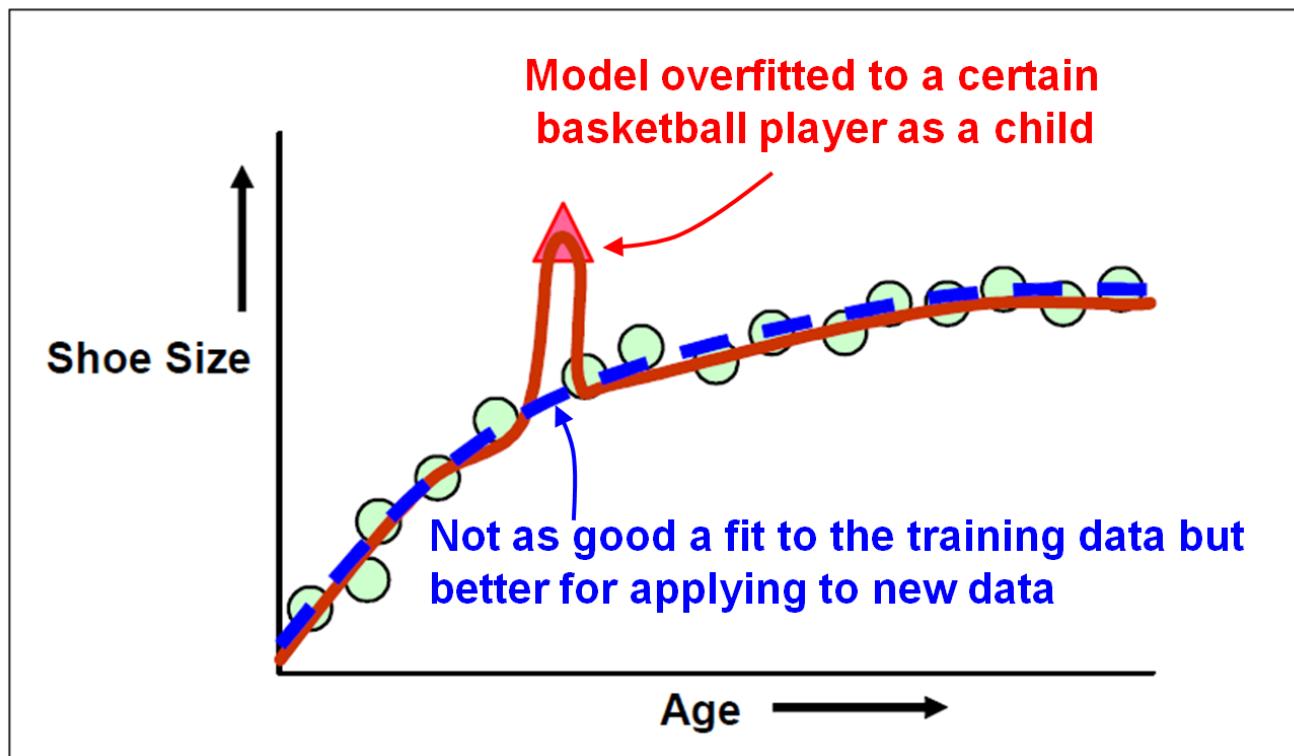
- **Over fitting:**

- 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.



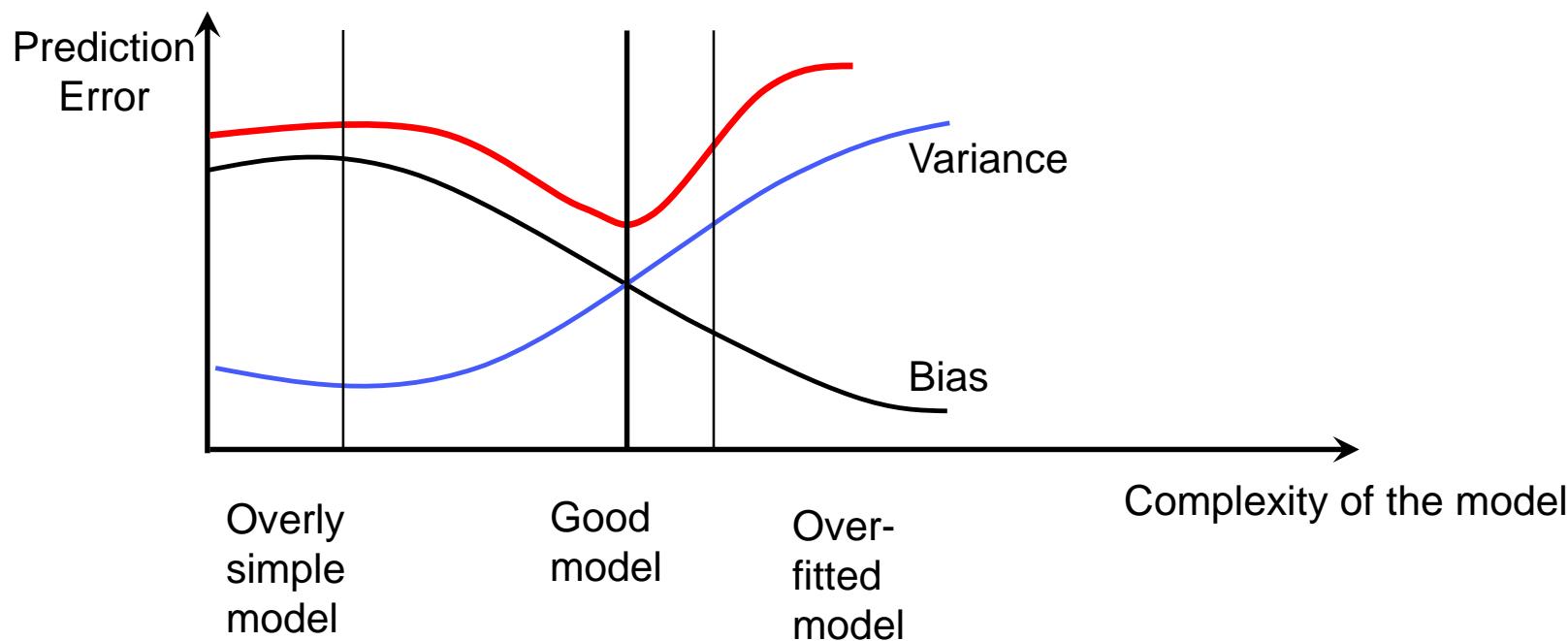
# Model overfitting

- 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

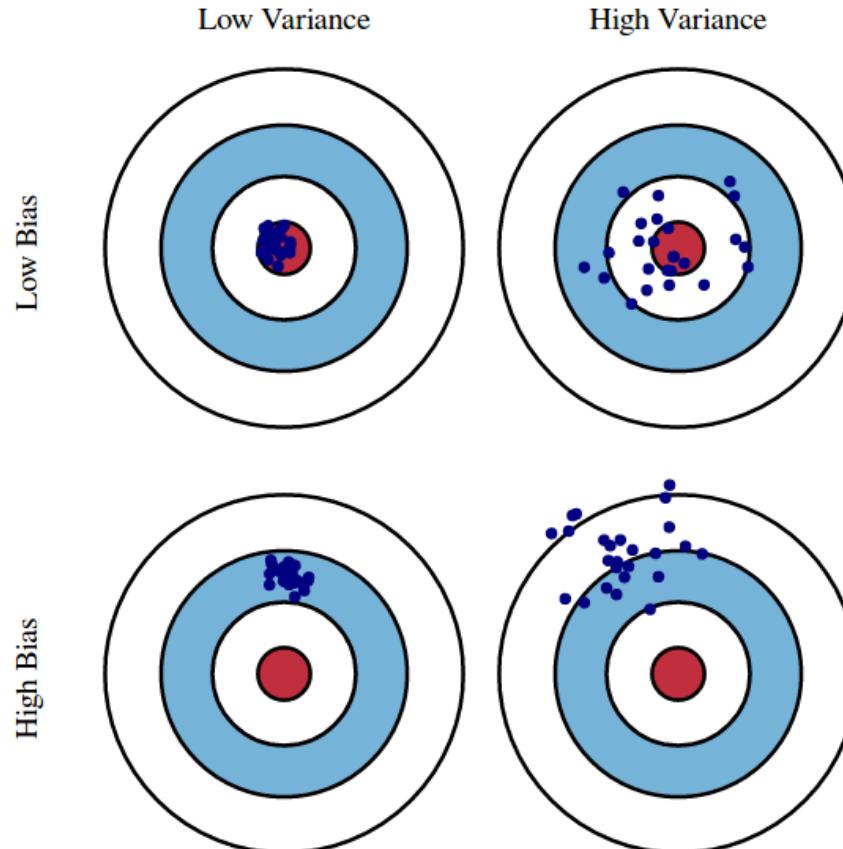


Fig. 1 Graphical illustration of bias and variance.

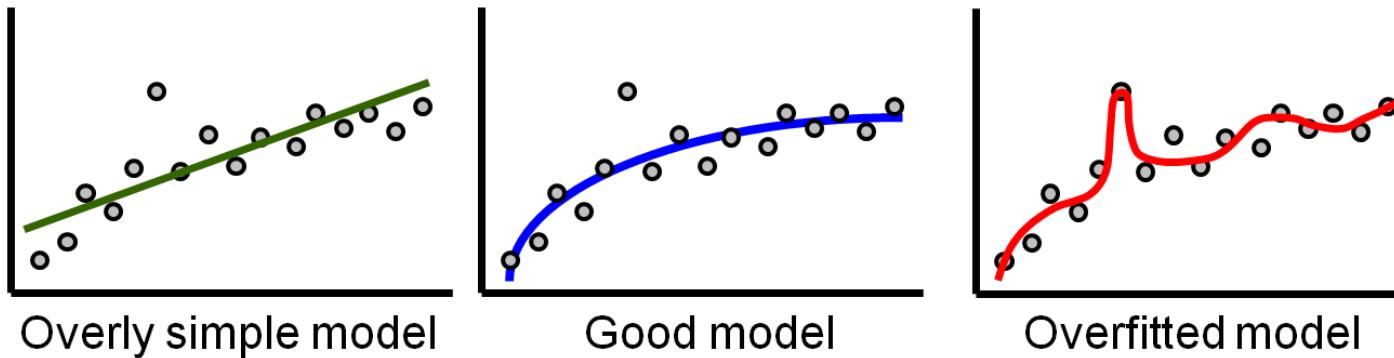
Source: <http://scott.fortmann-roe.com/docs/BiasVariance.html>

# 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.

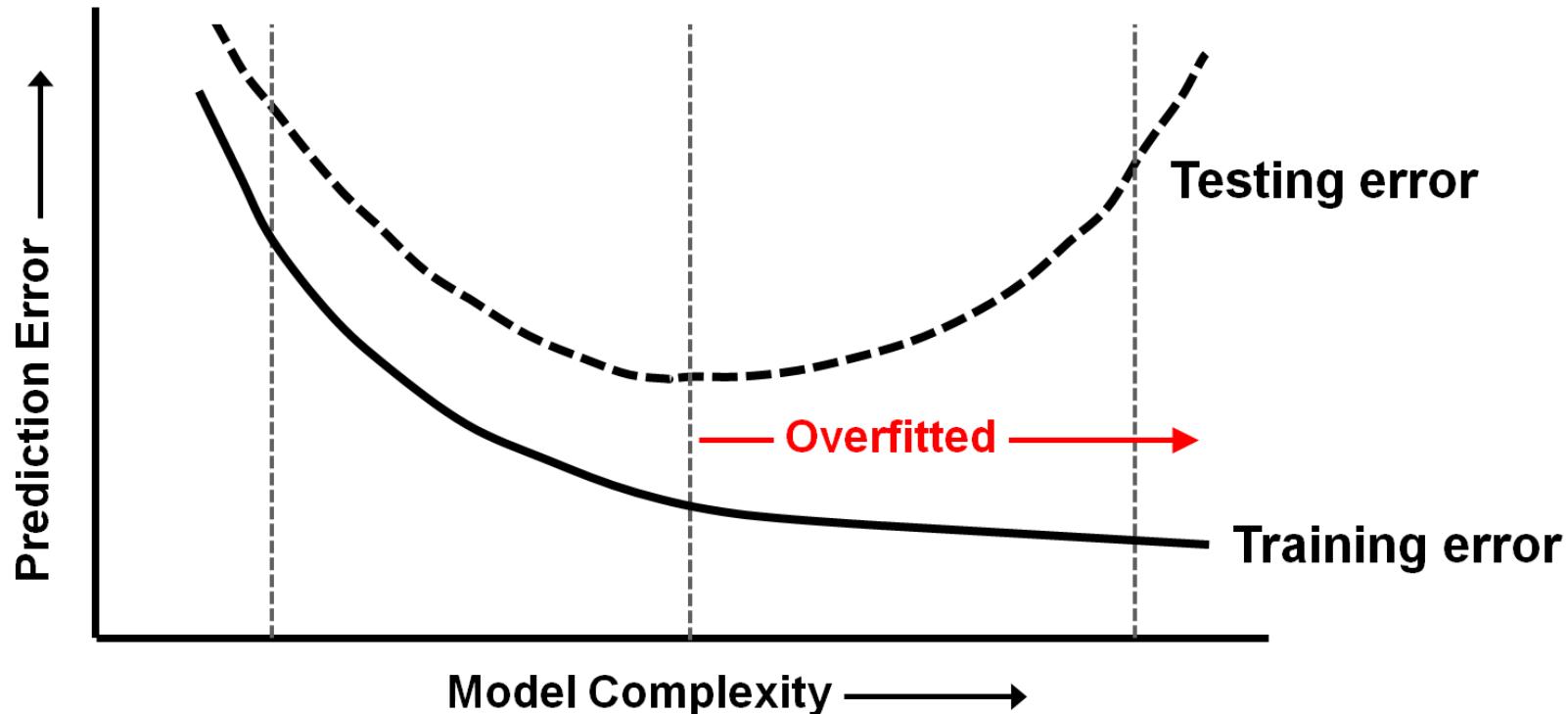
# When to stop training a model



Overly simple model

Good model

Overfitted model



# Introduction to Machine Learning

- Overview
- Data Science Methodology
- Data Understanding
- Data Preparation
- Categories of Machine Learning
- Learning Challenges
- Machine Learning Algorithms 
- Model Evaluation

# Classification – Naïve Bayes (supervised)

- **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)
  - Check a piece of text expressing positive emotions, or negative emotions?
  - Used for face recognition software.
- **Maximum conditional probability**
  - $Prob(Target|Input) = Prob(Input|Target) * \frac{Prob(Target)}{Prob(Input)}$

# Classification – Naïve Bayes

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

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
overcast	4	0	mild	4	2	normal	6	1	true	3	3
rainy	3	2	cool	3	1						

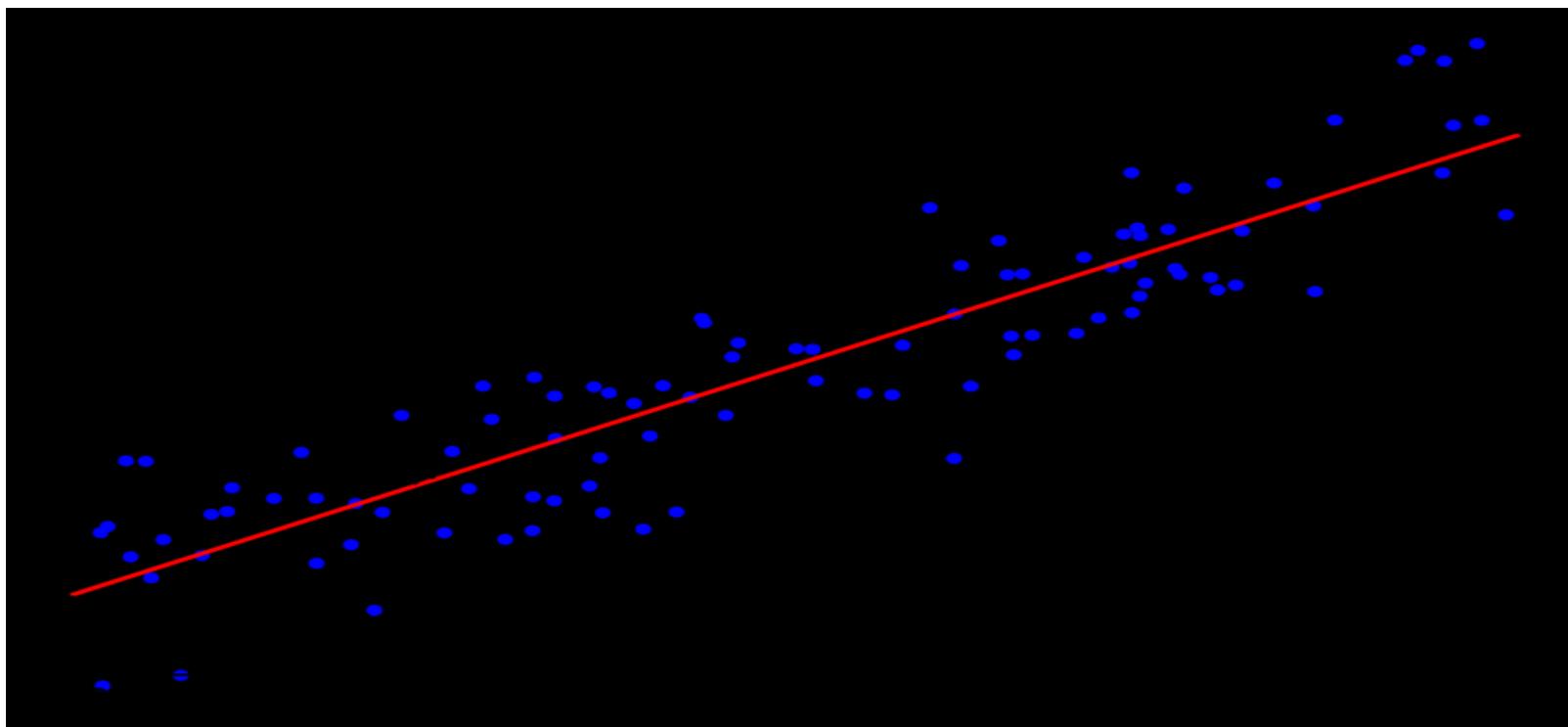
outlook		temperature		humidity		windy		play			
	yes	no	yes	no	yes	no	yes	no	yes	no	
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5
rainy	3/9	2/5	cool	3/9	1/5						

## Classification – Naïve Bayes

- $L(\text{yes}) = 2/9 * 3/9 * 3/9 * 3/9 = 0.0082$
- $L(\text{no}) = 3/5 * 1/5 * 4/5 * 3/5 = 0.0577$
- $P(\text{yes}) = 0.0082 * 9/14 = 0.0053$
- $P(\text{no}) = 0.0577 * 5/14 = 0.0206$
- The decision would be: NO.

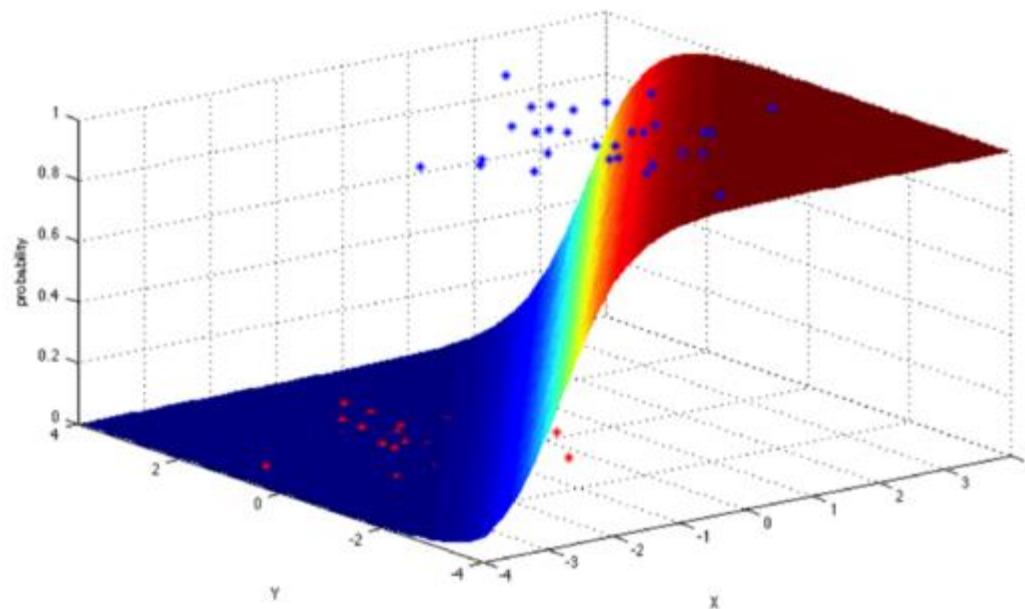
# Linear Regression (supervised)

- 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)

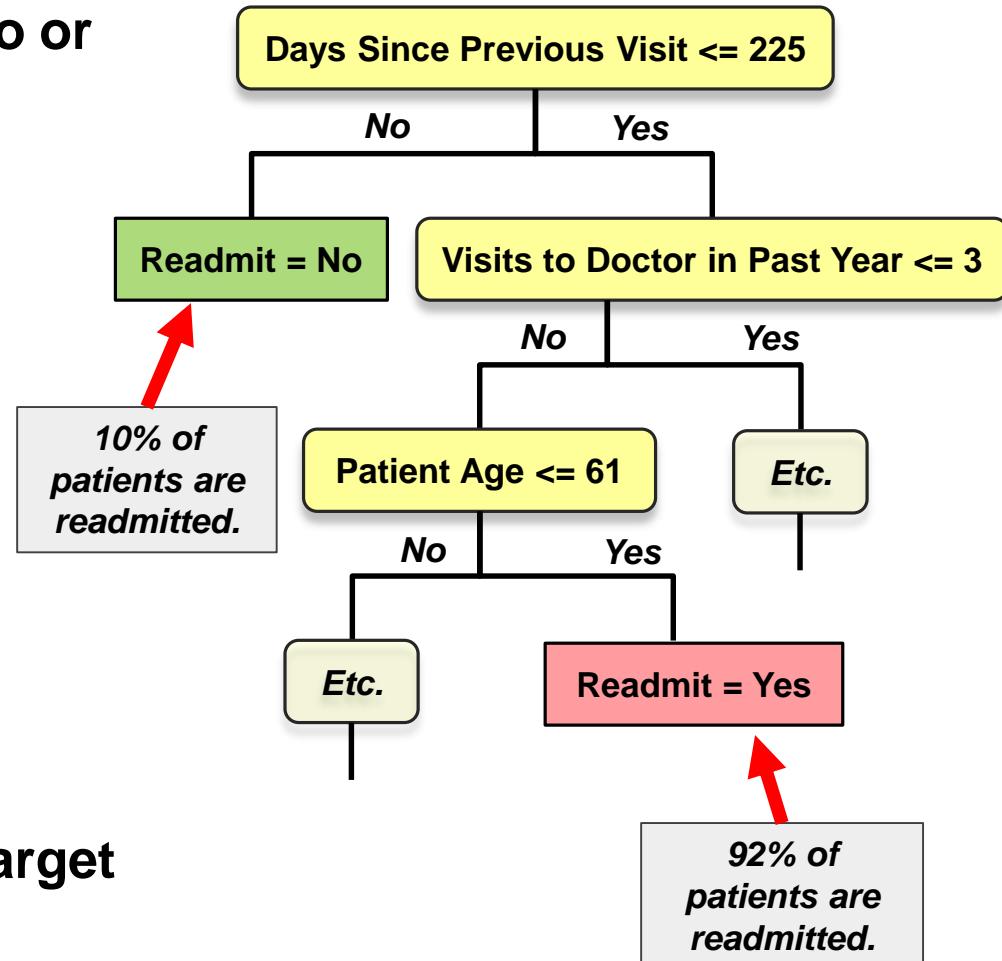
- Logistic regression is a powerful statistical way of modeling a binomial outcome with one or more explanatory variables. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the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



# Classification – Decision tree (supervised)

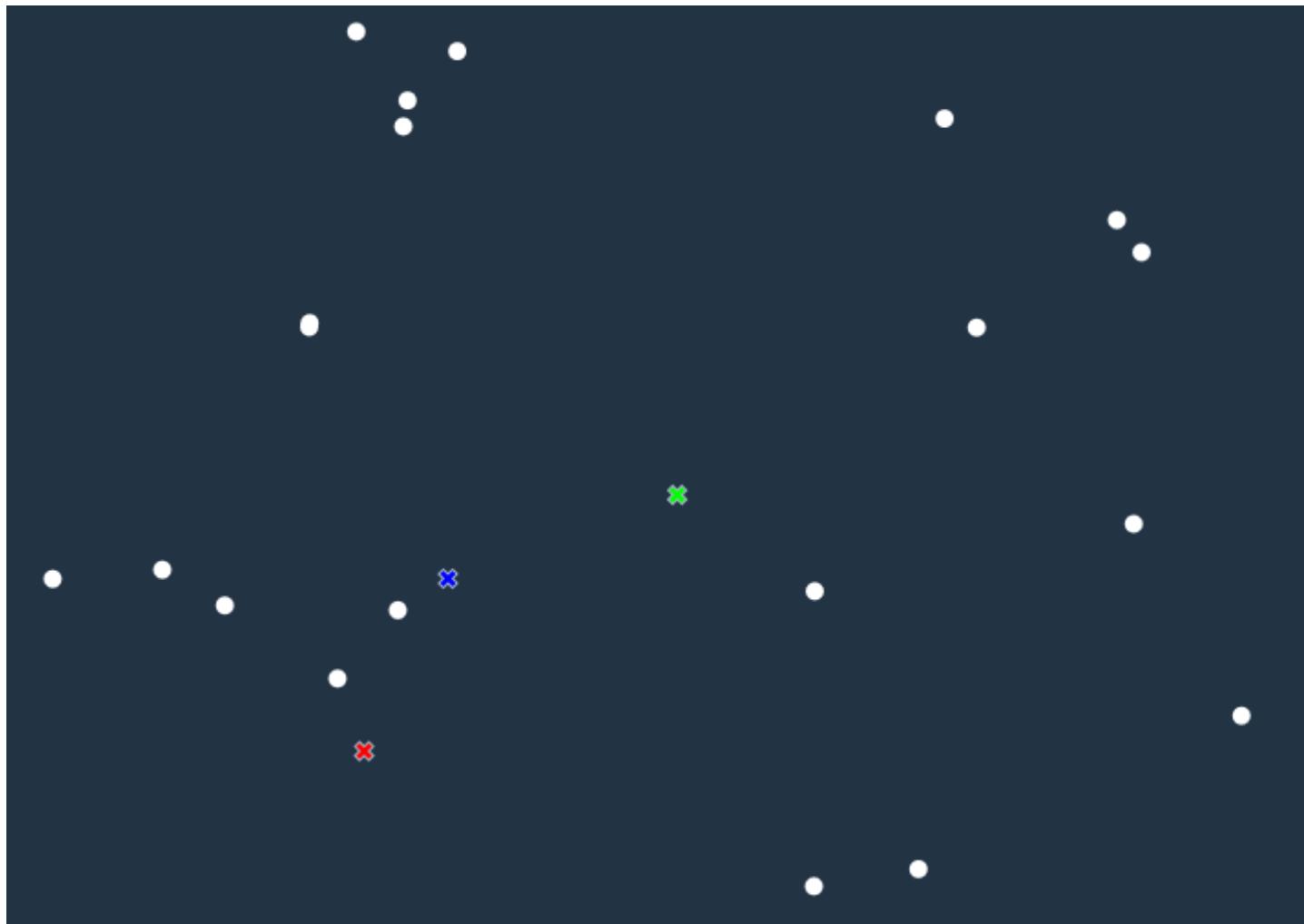
Modeling

- **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”).**



# Clustering – K-means method

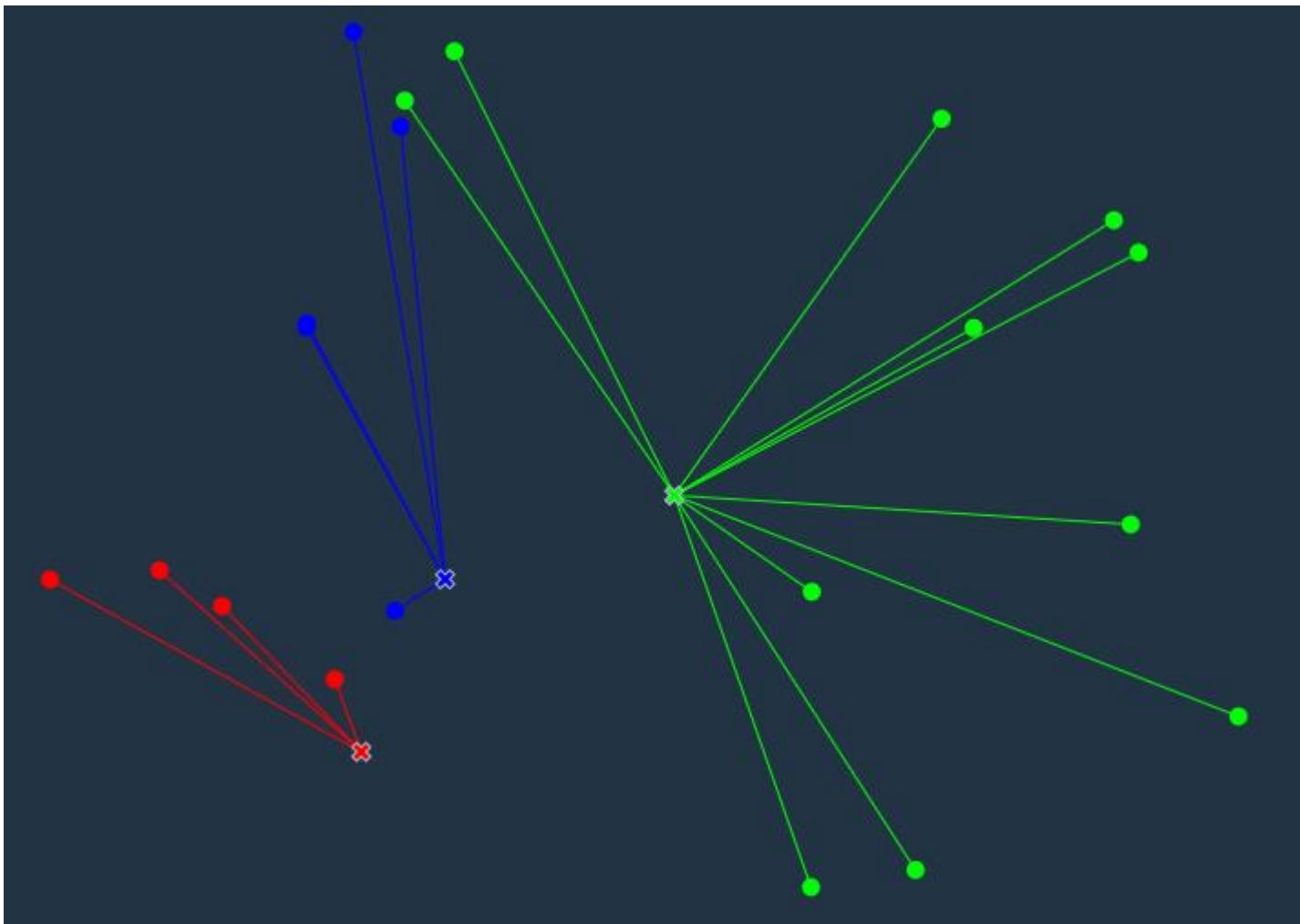
Modeling



Start with 20 data points and 3 clusters

# Clustering – K-means method

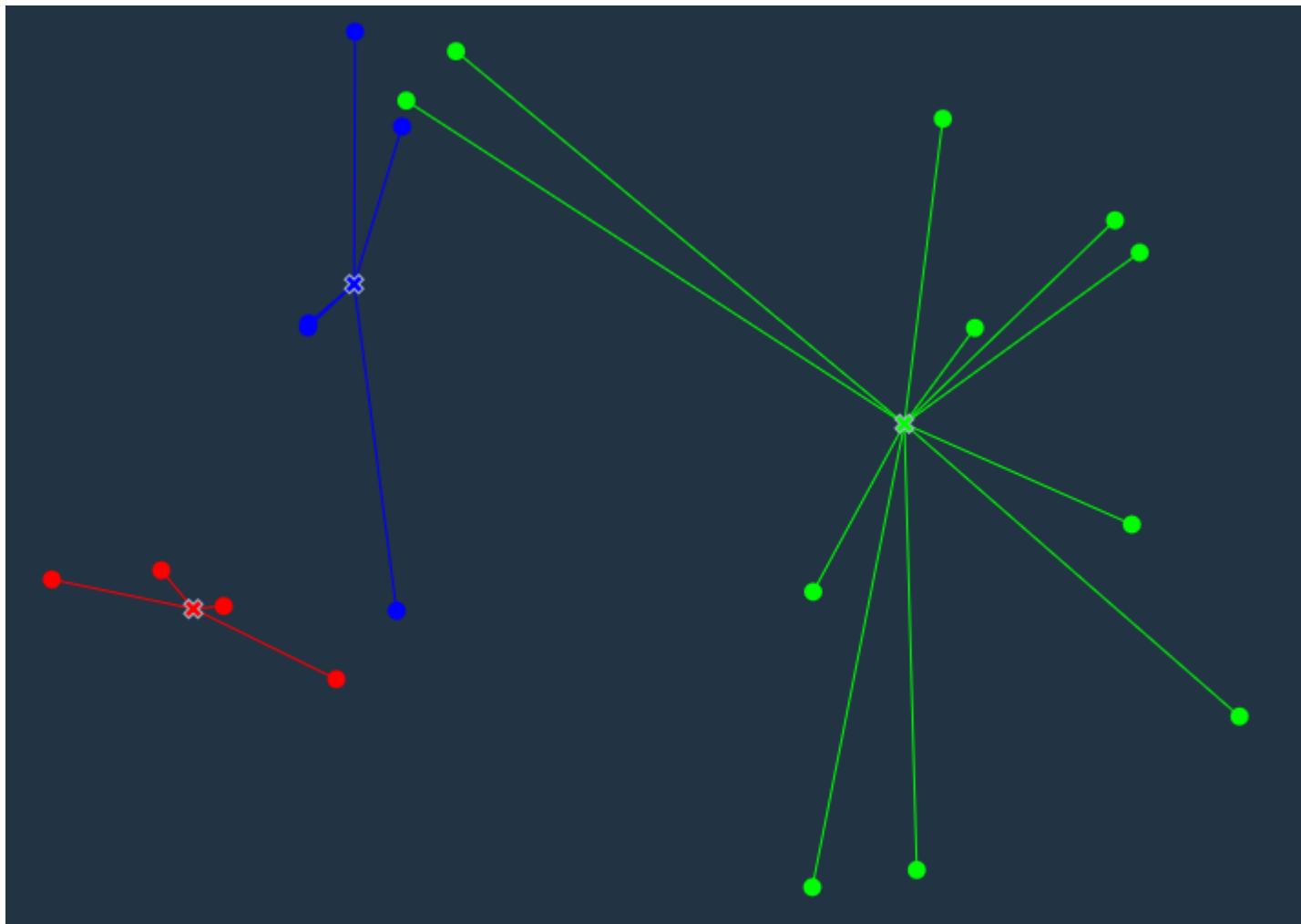
Modeling



Assign each data point to the nearest cluster

# Clustering – K-means method

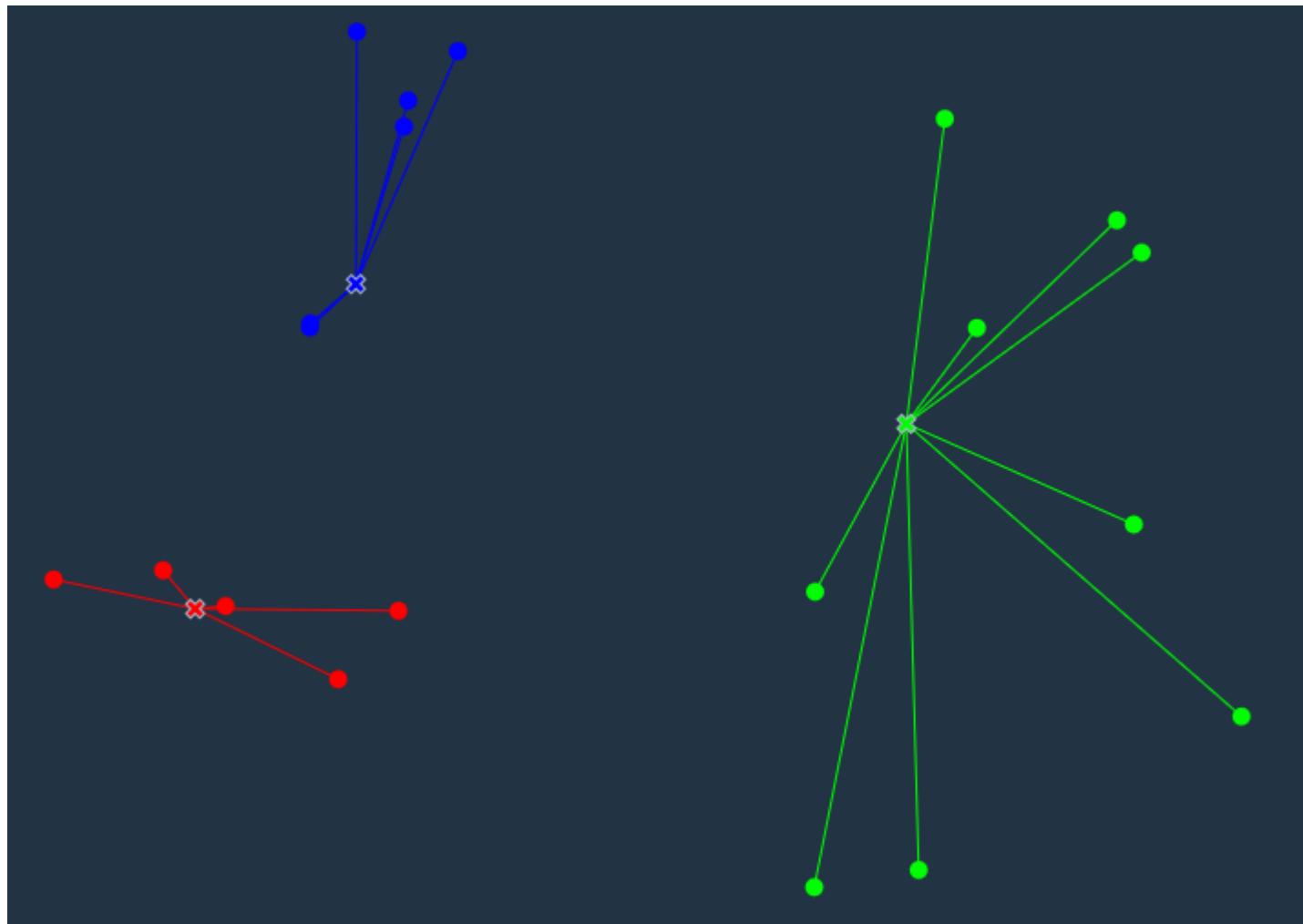
Modeling



Calculate centroids of new clusters

# Clustering – K-means method

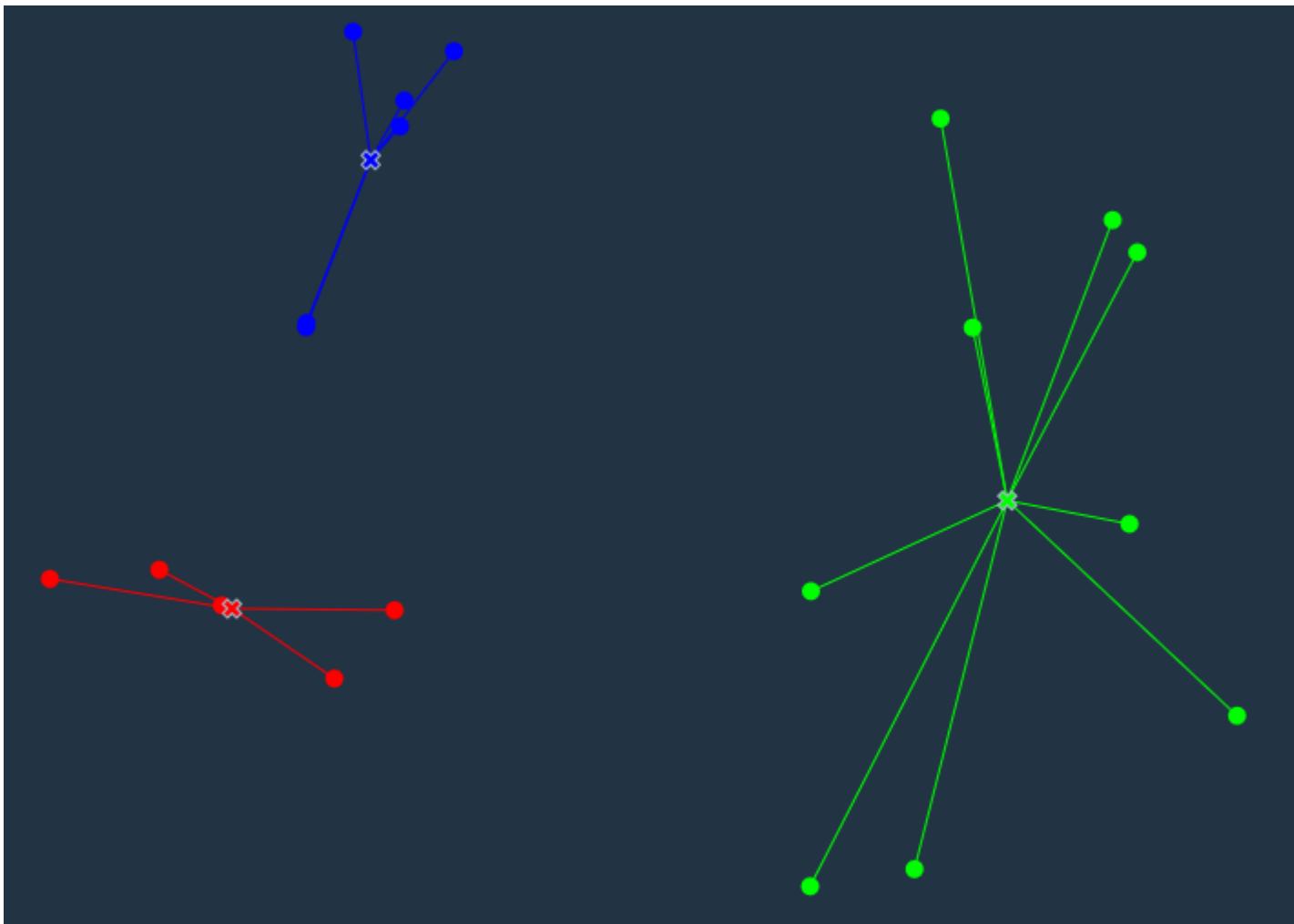
Modeling



Assign each data point to the nearest cluster

# Clustering – K-means method

Modeling



Calculate centroids of new clusters...until convergence

# Ensemble Modeling

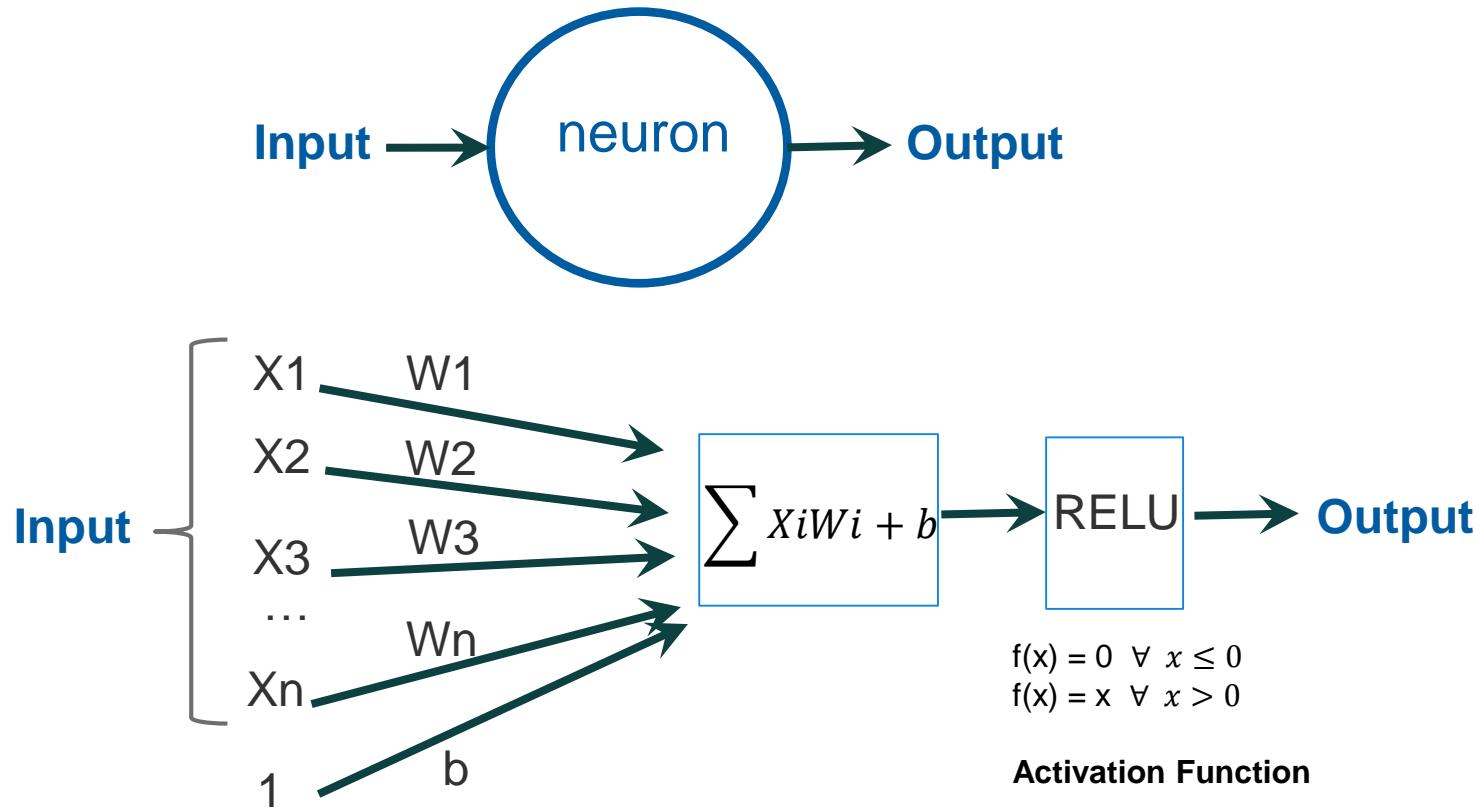
Modeling

- **Use a collection or ensemble of models instead of a single model to create more reliable and accurate predictive models**
- **Bagging**
  - New training datasets are generated based on random sampling with replacement of the original data set
  - Models are constructed for each sample and the results are combined
  - Random Forest is bagging applied to Decision Trees
- **Boosting**
  - Successive models are built to predict observations misclassified from earlier models.
  - Gradient boosting - train each subsequent model on the residuals (error between predicted value and actual value).

# Neural Network

- **Originated in 1940s**
- **Became very popular this decade**
  - Hardware – GPUs, Storage
  - Availability of Large Datasets for Training
  - Better performing algorithms.
- **Especially useful for human perception type task**
  - Image Classification
  - Object Recognition
  - Speech Recognition
  - Natural Language Understanding
  - Machine Translation
  - ...

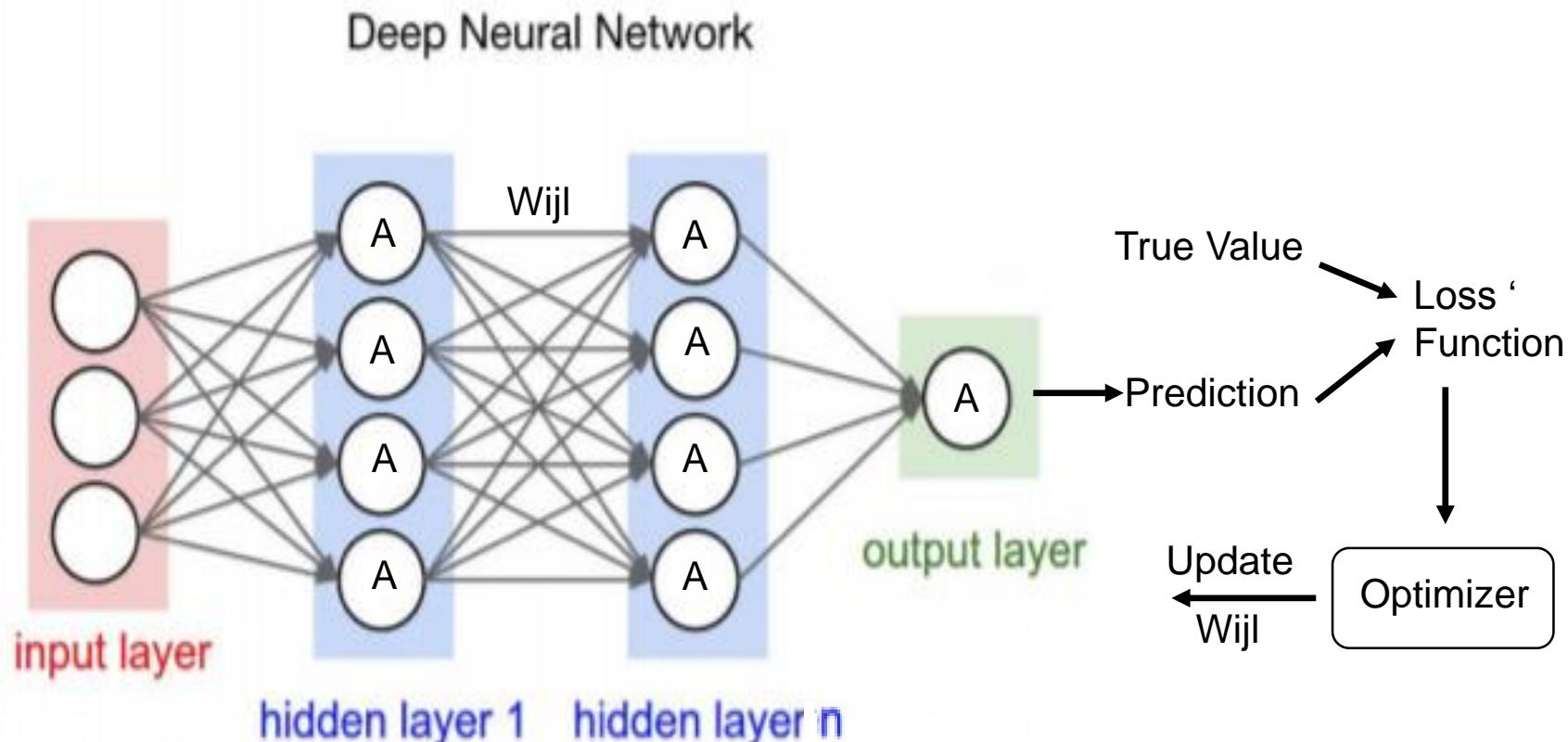
# What is an Artificial Neuron



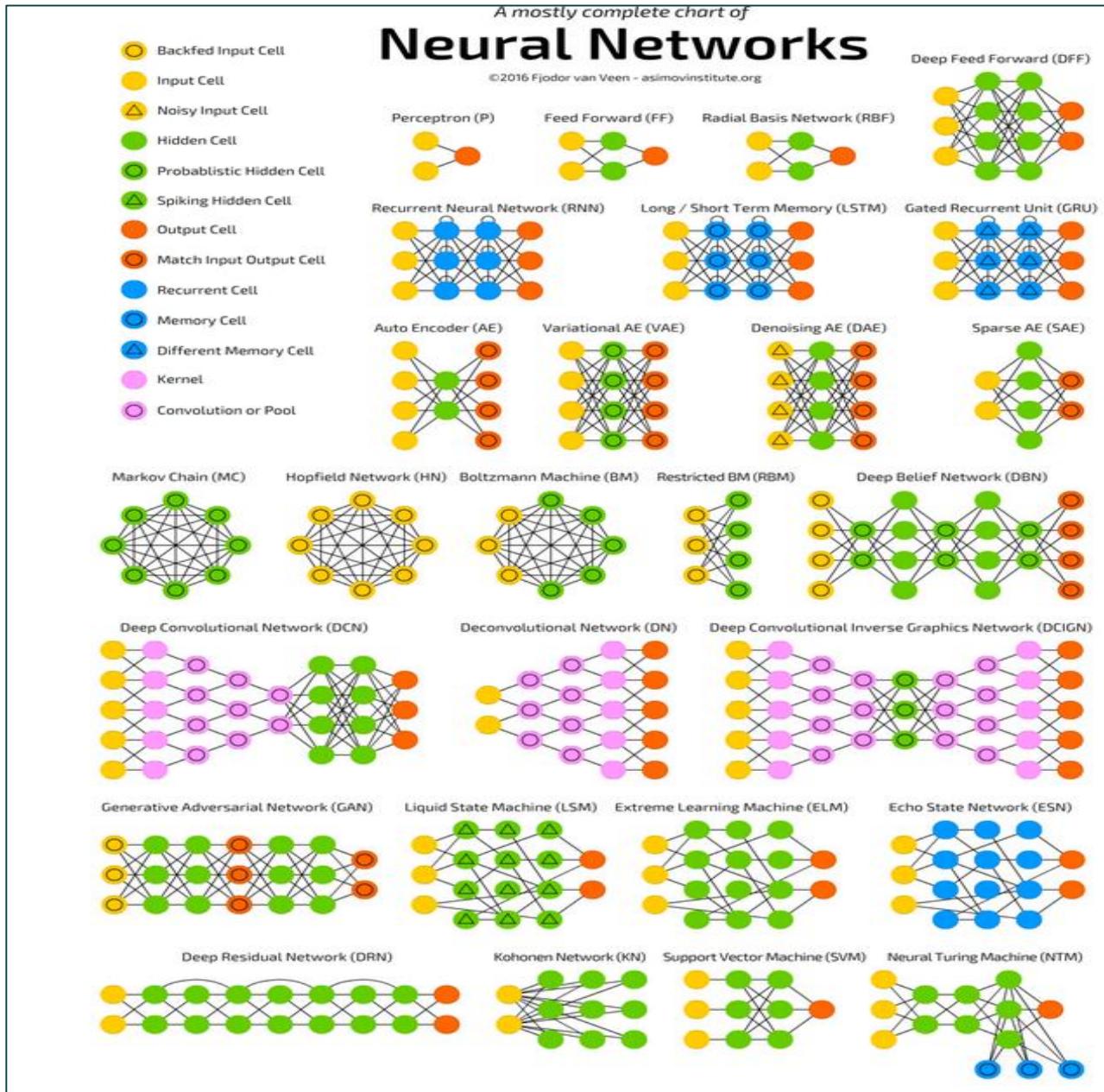
# Neural Network

Modeling

- Inspired by the way the human brain works.



Wijl – weight from neuron (i) in level (l-1) to neuron (j) in level (l)



# Common Types of Neural Networks

- Convolutional Neural Networks
  - Image and Video recognition
  - Recommender systems
  - Natural language processing
  - ..
- Recurrent Neural Networks
  - Speech Recognition
  - Handwriting Recognition
  - Machine Translation
  - ...

# Introduction to Machine Learning

- Overview
- Data Science Methodology
- Data Understanding
- Data Preparation
- Categories of Machine Learning
- Learning Challenges
- Machine Learning Algorithms
- Model Evaluation



# 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

# K-Fold Cross Validation

- Instead of using a separate validation set
- Shuffle Training Samples and sub-divide into “K” folds (groups)
- Train “K” models using K-1 folds as training data and 1 Fold as Test Data
- For example, K=4
  - Model 1 Train on 1,2,3 Test on 4 – calculate and store E1 (Error)
  - Model 2 Train on 2,3,4 Test on 1 – E2
  - Model 3 Train on 3,4,1 Test on 2 - E3
  - Model 4 Train on 4,1,2 Test on 3 - E4
  - $E = (E1+E2+E3+E4)/4$
- A common value for K is 10

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

$$\text{Precision} = \text{tp}/(\text{tp} + \text{fp})$$

$$\text{Recall} = \text{sensitivity} = \text{True Positive Rate} = \text{tp}/(\text{tp} + \text{fn})$$

$$\text{FPR} = \text{fp}/(\text{fp} + \text{tn}) \quad 1 - \text{specificity}$$

ROC = plot of TPR/FPR at different thresholds

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

$$\text{Precision} = \text{tp}/(\text{tp} + \text{fp})$$

$$\text{Recall} = \text{sensitivity} = \text{True Positive Rate} = \text{tp}/(\text{tp} + \text{fn})$$

$$\text{FPR} = \text{fp}/(\text{fp} + \text{tn}) \quad 1 - \text{specificity}$$

ROC = plot of TPR/FPR at different thresholds

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

$$\text{Precision} = \text{tp}/(\text{tp} + \text{fp})$$

$$\text{Recall} = \text{sensitivity} = \text{True Positive Rate} = \text{tp}/(\text{tp} + \text{fn})$$

$$\text{FPR} = \text{fp}/(\text{fp} + \text{tn}) \quad 1 - \text{specificity}$$

ROC = plot of TPR/FPR at different thresholds

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

$$\text{Precision} = \text{tp}/(\text{tp} + \text{fp})$$

$$\text{Recall} = \text{sensitivity} = \text{True Positive Rate} = \text{tp}/(\text{tp} + \text{fn})$$

$$\text{FPR} = \text{fp}/(\text{fp} + \text{tn}) \quad 1 - \text{specificity}$$

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)

**Accuracy = 95%      Recall = 95%      Precision=95%**

# 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)

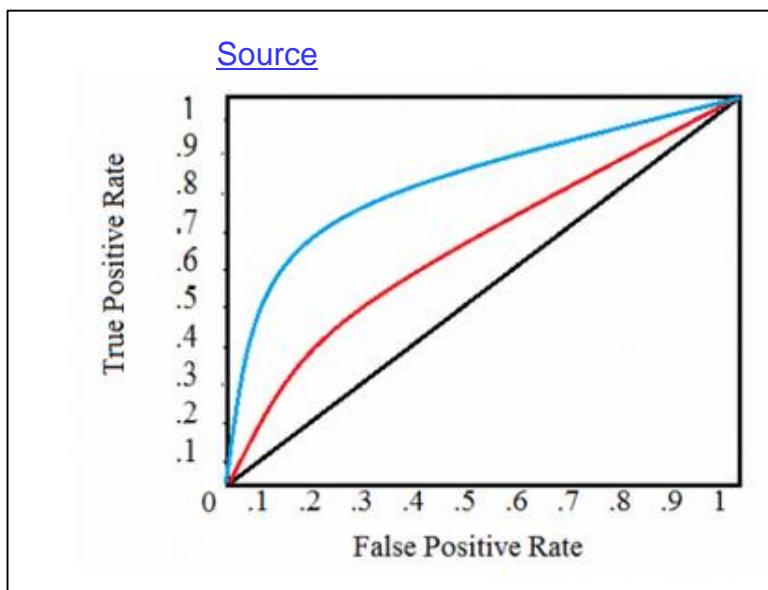
**Accuracy = 95%      Precision=16.1%      Recall = 95%**

What is happening here:

The condition is RARE and the 5% FALSE POSITIVES are still way higher in numbers than the true positives. Need 99% or higher specificity.

## Model Evaluation - Metrics

- **Accuracy** =  $\frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{Tp+Tn}{Tp+Tn+Fp+Fn}$
- $F1 = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$
- **Area under Receiver Operating Characteristic (ROC)**



# Some items to think about ....

## ▪ Business

- What are your goals?
- What are the criteria for success?

## ▪ Data

- Do you need labeled (\$\$) data?
- What is the quality of your data?
- What features are pertinent?
- Do you have enough data?
- How are you going to obtain the data?

## ▪ Models

- What algorithms to use?
- What metrics to evaluate the algorithms?
- Would ensembles help?

## ▪ Implementation

- How quickly does a new instance need to be classified (online/batch)?
- Do you need to scale?
- What resources do you have? Memory?, GPUs?, Compute?
- How are you going to get feedback?

# Watson Studio Platform

# Challenges in delivering value with Data Science

## Data

- Data resides in silos and difficult to access
- Unstructured and external data wasn't considered

## Governance

- Self-service isn't a reality, if the data isn't secure
- Understanding lineage and getting to a system of truth

## Skills

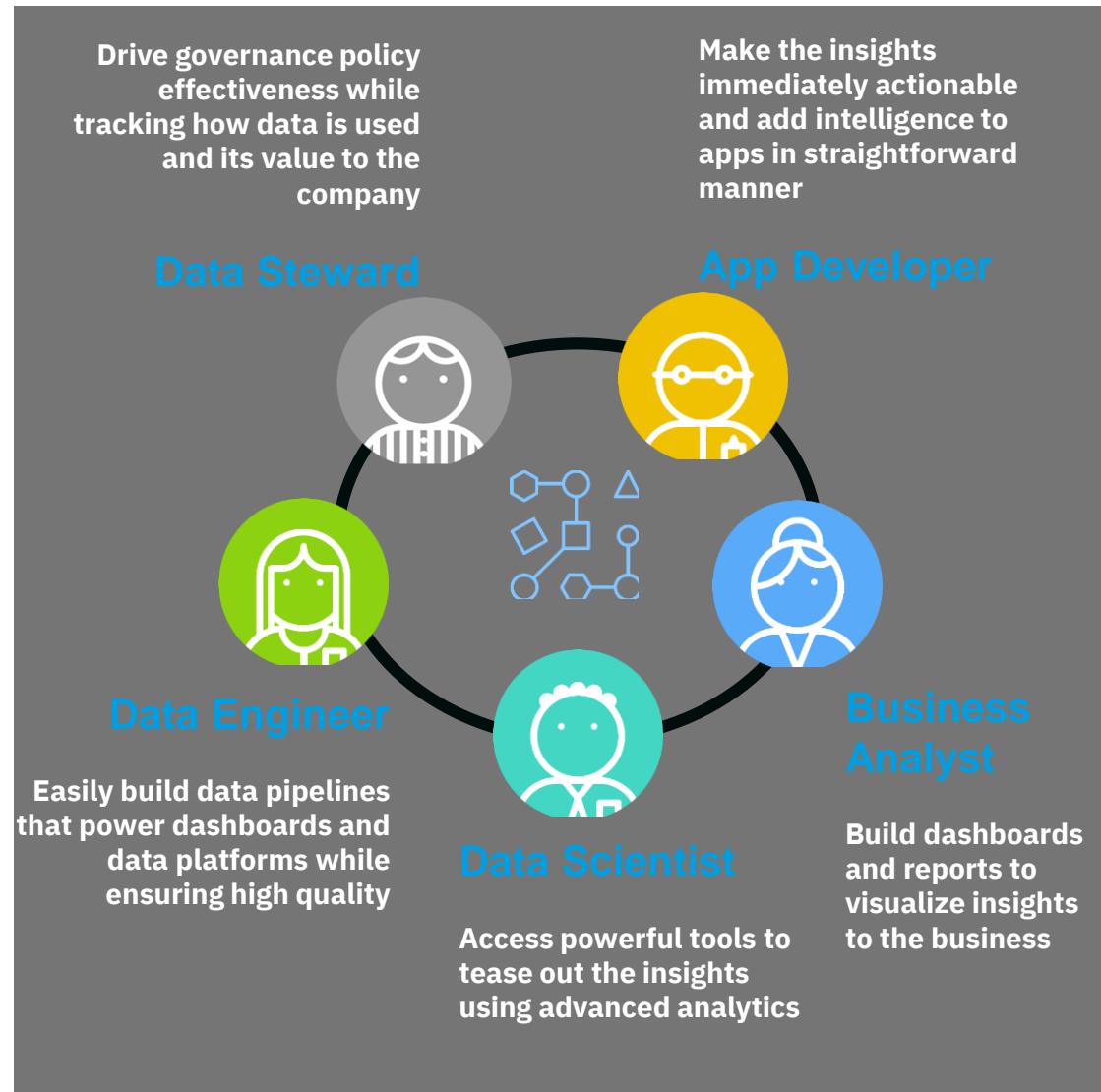
- Data Science skills are in low supply and high demand

## Infrastructure

- Need an environment that enables collaboration and deployment to production
- Discrete tools present barriers to progress

# IBM Watson Studio Platform

An integrated platform of tools, services, and metadata that help companies or agencies accelerate their shift to be data-driven organizations.



# Watson Studio supports end-to-end AI workflow

*Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.*



**Connect** and discover content from multiple data sources in the cloud or on premises. Bring **structured** and **unstructured** data to one toolkit.

**Find** data (structured, unstructured) and AI assets (e.g., ML/DL models, notebooks, Watson Data Kits) in the **Knowledge Catalog** with intelligent search and giving the right access to the right users.

Clean and prepare your data with **Data Refinery**, a tool to create data preparation pipelines visually. Use popular open source libraries to prepare unstructured data.

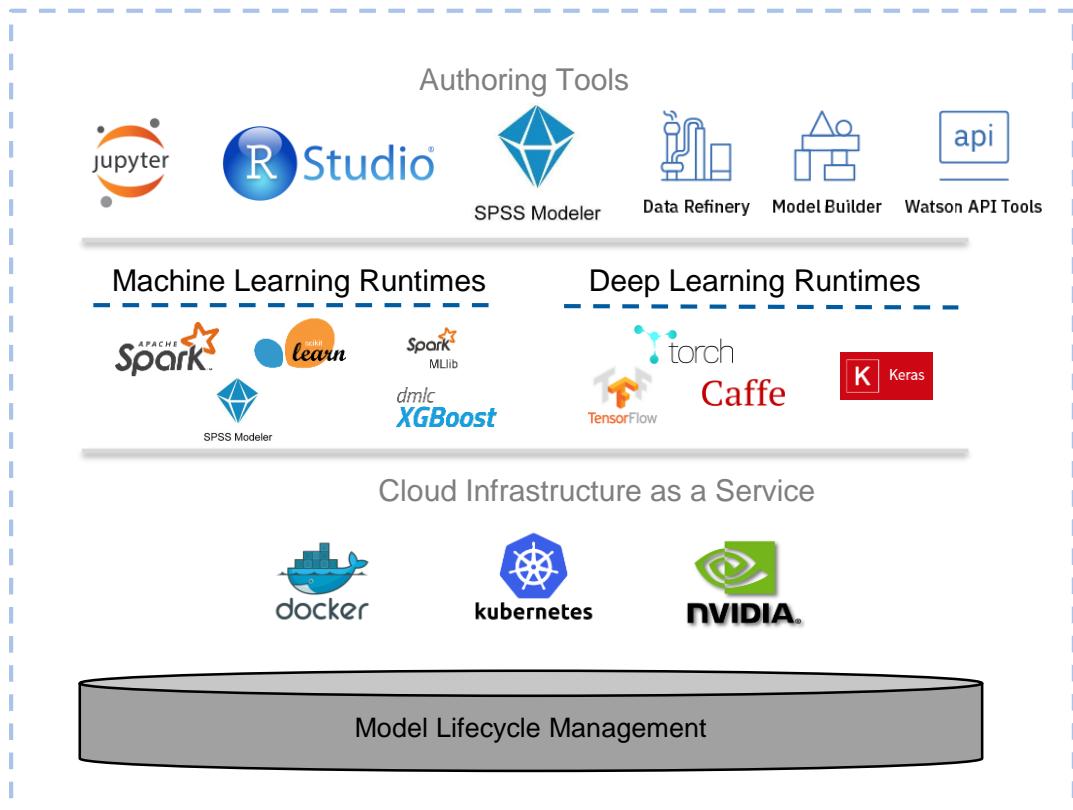
**Democratize** the creation of ML and DL models. Design your AI models **programmatically** or **visually** with the most popular **open source** and IBM ML/DL frameworks. Train at scale on **GPUs** and **distributed** compute

Deploy your models easily and have them **scale automatically** for online, batch or streaming use cases

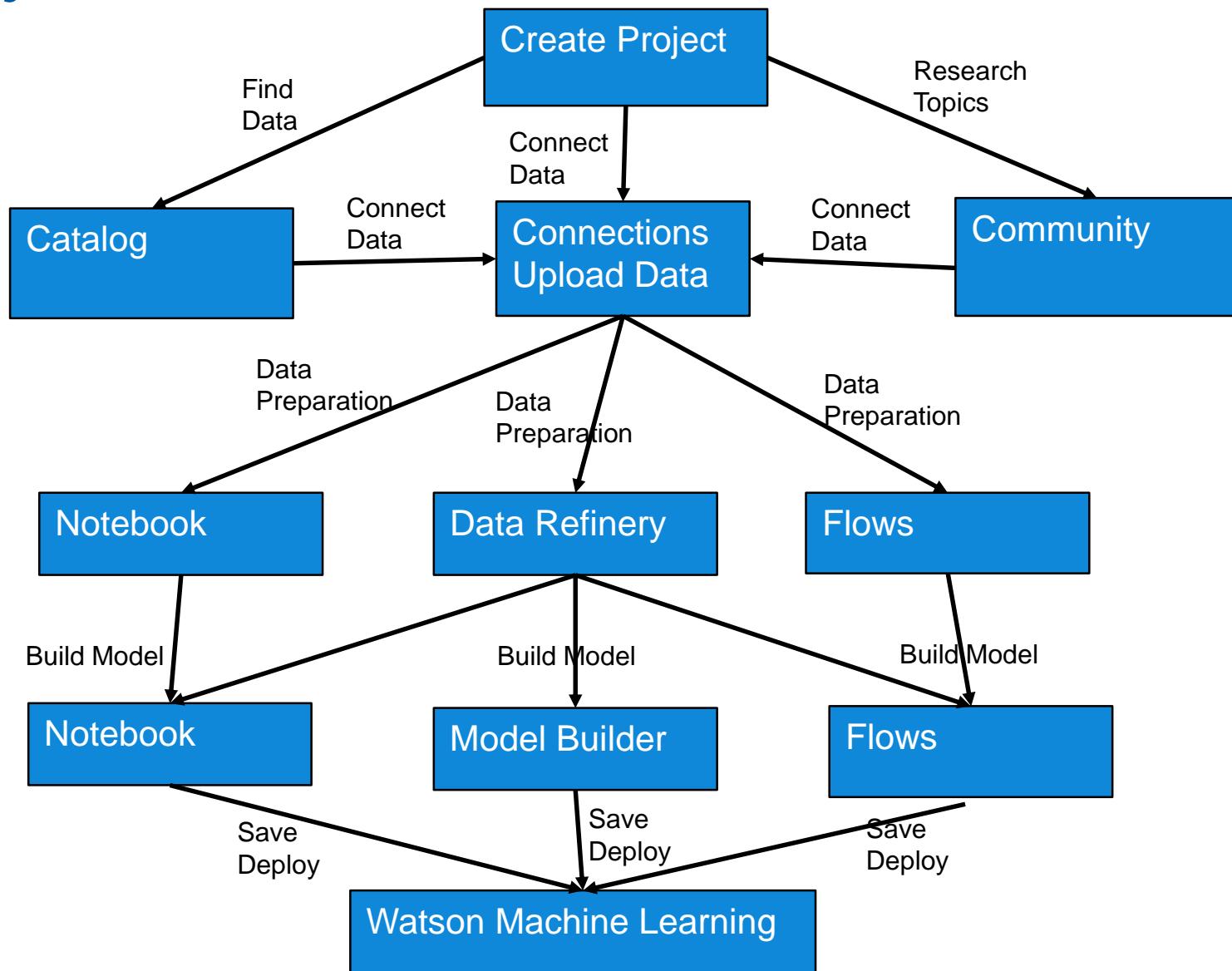
Monitor the performance of the models in production and trigger automatic retraining and redeployment of models.

# Watson Studio Tools

- Create, collaborate, deploy, and monitor
- Best of breed open source & IBM tools
- Code (R, Python or Scala) and no-code/visual modeling tools
- Open Source and IBM libraries/frameworks
- Fully managed service
- Container-based resource management
- Elastic pay as you go cpu/gpu power



# Project Flow



# Watson Studio – Projects

Making Data Science a Team Sport

Watson Studio Labs

Last Updated: Apr 14 2018

[Readme](#)

Date created  
Apr 10 2018

Description  
No description available

Storage  
0.0% of 25 GB used

Collaborators  
View all (2)

- Bernard Beekman Admin
- DSX26000 User Editor

Bookmarks  
View all (0)

You currently have 0 bookmarks

[IBM Cloud Inform](#)

**Project Overview**

Add to project

Tools

- Data Science
- Visual Recognition
- Deep Learning
- Modeler
- Business Analytics
- Streams Designer
- Data Engineering

Cancel Save

Storage

Type Cloud Object Storage Bucket Name watsonstudiolabs-bdoncodelete-pr-vadimh7wjuw

434.49 KB Used 0.0% of 25 GB used

Associated services

NAME	SERVICE TYPE	PLAN
predictive-modelling-bd	Machine Learning	
Machine Learning	Machine Learning	
DSX-Spark-2.0	Machine Learning	

**Project Settings**

Watson Studio Labs

[Add to project](#)

Find collaborators

Collaborators

NAME	STATUS	ACTIONS
Bernard Beekman	Active	⋮
DSX26000 User	Active	⋮

**Add collaborators**

Watson Studio Labs

Collaborators

Admin (1)

beekman@us.ibm.com

STATUS

Active

ACCESS LEVEL

Enter email address

Cancel Invite

**Add Collaborators**

Add to project

Connected data

Notebook

Connection

Data asset

Model

Experiment

Stream flow

Modeler flow

Dashboard

Data flow

Environment

Visual recognition model

New data asset

New visual recognition model

New notebook

Overview Assets Environments Bookmarks Deployments Collaborators Settings

What assets are you looking for?

**Assets**

0 assets selected.

NAME	TYPE	SERVICE	CREATED BY
titanic.csv	Data Asset	Project	Bernard Beekman
titanic_cleaned.csv	Data Asset	Project	Bernard Beekman

**Visual recognition models**

NAME	MODEL ID	SERVICE INSTANCE	LAST MODIFIED
------	----------	------------------	---------------

you currently have no visual recognition models

**Notebooks**

NAME	CHILDREN	SCENARIOS	STATUS	LANGUAGE	LAST EDITOR	LAST MODIFIED	ACTIONS
Titanic						12 Apr 2018	⋮

**Analytic and Data Assets Organized in Projects**

# Watson Studio – Community Cards

*Built-in learning to get started*

Search results (355)

Sort by: Most Related

Popular filters: Spark, Deep Learning, Brunel

What are you looking for? Article

Articles

ARTICLE	Leaflet: Interactive web maps with R	ARTICLE	Open Sourcing 223GB of Driving Data -...	ARTICLE	Learn TensorFlow and Deep Learning Together...	ARTICLE	sparklyr – R interface for Apache Spark
AUTHOR	RStudio Blog	AUTHOR	Udacity	AUTHOR	Big Data University	AUTHOR	RStudio Blog
DATE	May 20, 2016	DATE	Nov 09, 2016	DATE	May 01, 2017	DATE	Oct 06, 2016
TOPIC	Visualization	FORMAT	Web page	TOPIC	Open Data	FORMAT	Web page
				TOPIC	Deep Learning	FORMAT	Web page
				TOPIC	Analytics +1	FORMAT	Web page

Search results (78)

Sort by: Most Related

Popular filters: Spark, Deep Learning, Brunel

What are you looking for? Notebook

Notebooks

NOTEBOOK	A TensorFlow regression model to predict...	NOTEBOOK	Access Db2 Warehouse on Cloud and Db2 with...	NOTEBOOK	Access MySQL with Python	NOTEBOOK	Access MySQL with R
AUTHOR	IBM	AUTHOR	IBM	AUTHOR	IBM	AUTHOR	IBM
DATE	Apr 06, 2018	DATE	Mar 20, 2018	DATE	Mar 27, 2018	DATE	Mar 27, 2018
TOPIC	Economy & Business	TOPIC	Economy & Business	TOPIC	Transportation	TOPIC	Transportation
NOTEBOOK	Access PostgreSQL with Python	NOTEBOOK	Access PostgreSQL with R	NOTEBOOK	Analyze Facebook Data Using IBM Watson and...	NOTEBOOK	Analyze accident reports on Amazon EMR Spark
AUTHOR	IBM	AUTHOR	IBM	AUTHOR	IBM	AUTHOR	IBM
DATE	Mar 20, 2018	DATE	Mar 20, 2018	DATE	Mar 20, 2018	DATE	Oct 12, 2017
TOPIC	Transportation	TOPIC	Transportation	TOPIC	Transportation	TOPIC	Transportation

Search results (119)

Sort by: Most Related

Popular filters: Spark, Deep Learning, Brunel

What are you looking for? Tutorial

Tutorials

TUTORIAL	What I Learned Implementing a Classifier...	TUTORIAL	Best packages for data manipulation in R	TUTORIAL	Common Excel Tasks Demonstrated in Pandas	TUTORIAL	An Introduction to Stock Market Data...																																																								
AUTHOR	Jean-Nicolas Hould	AUTHOR	DataScience+	AUTHOR	Practical Business Python	AUTHOR	Curtis Miller																																																								
DATE	Apr 17, 2017	DATE	Jul 12, 2016	LEVEL	Intermediate	TOPIC	Data Science	LEVEL	Beginner	TOPIC	Visualization					LEVEL	Beginner	TOPIC	Visualization									TUTORIAL	Pulling and Displaying ETF Data	TUTORIAL	Super Fast String Matching in Python	TUTORIAL	Understanding empirical Bayes estimation...	TUTORIAL	Brunel: interactive visualizations in Jupyter...	AUTHOR	RStudio	AUTHOR	van den Blog	AUTHOR	Variance Explained	AUTHOR	Data Science Experience Blog	DATE	Feb 09, 2017	DATE	Nov 20, 2017	LEVEL	Intermediate	TOPIC		DATE	Mar 13, 2018	TOPIC									
LEVEL	Intermediate	TOPIC	Data Science	LEVEL	Beginner	TOPIC	Visualization																																																								
				LEVEL	Beginner	TOPIC	Visualization																																																								
TUTORIAL	Pulling and Displaying ETF Data	TUTORIAL	Super Fast String Matching in Python	TUTORIAL	Understanding empirical Bayes estimation...	TUTORIAL	Brunel: interactive visualizations in Jupyter...																																																								
AUTHOR	RStudio	AUTHOR	van den Blog	AUTHOR	Variance Explained	AUTHOR	Data Science Experience Blog																																																								
DATE	Feb 09, 2017	DATE	Nov 20, 2017	LEVEL	Intermediate	TOPIC		DATE	Mar 13, 2018	TOPIC																																																					
LEVEL	Intermediate	TOPIC		DATE	Mar 13, 2018	TOPIC																																																									

Search results (295)

Sort by: Most Related

Popular filters: Spark, Deep Learning, Brunel

What are you looking for? Data Set

Data Sets

DATA SET	Adolescent fertility rate (births per 1,000...)	DATA SET	Agriculture, value added (% of GDP) by...	DATA SET	Airbnb Data for Analytics: Amsterdam Calendar	DATA SET	Airbnb Data for Analytics: Amsterdam Listings
AUTHOR	IBM	AUTHOR	IBM	AUTHOR	IBM	AUTHOR	IBM
DATE	May 22, 2016	DATE	May 22, 2016	DATE	Dec 20, 2016	DATE	Dec 20, 2016
TOPIC	Society	TOPIC	Economy & Business	TOPIC	Economy & Business	TOPIC	Economy & Business
DATA SET	Airbnb Data for Analytics: Amsterdam Reviews	DATA SET	Airbnb Data for Analytics: Antwerp Calendar	DATA SET	Airbnb Data for Analytics: Antwerp Listings	DATA SET	Airbnb Data for Analytics: Antwerp Listings...
AUTHOR	IBM	AUTHOR	IBM	AUTHOR	IBM	AUTHOR	IBM
DATE	Dec 20, 2016	DATE	Dec 20, 2016	DATE	Dec 20, 2016	DATE	Dec 20, 2016
TOPIC	Economy & Business	TOPIC		TOPIC		TOPIC	Business

# Watson Studio – Create Assets

The best of open source and IBM Watson tools to create start-of-the-art data products

This screenshot shows the Watson Studio interface under the 'IBM services' section. It lists several data storage and processing options:

- IBM services:** BigInsights HDFS, Cloud Object Storage, Cloud Object Storage (infrastructure), Cloudant, Compose for MySQL, Compose for PostgreSQL, Db2, Db2 for i, Db2 for z/OS, Db2 Hosted, Db2 on Cloud, Db2 Warehouse, Informix, Object Storage OpenStack Swift (infrastructure), PureData for Analytics, Watson Analytics.
- Third-party services:** Amazon Redshift, Amazon S3, Apache Hive, Cloudera Impala, Dropbox, Hortonworks HDFS, Microsoft Azure SQL Database, Microsoft SQL Server, MySQL, Oracle, Pivotal Greenplum, PostgreSQL, Remote file system transfer, Salesforce.com, Sybase, Sybase IQ, Teradata.

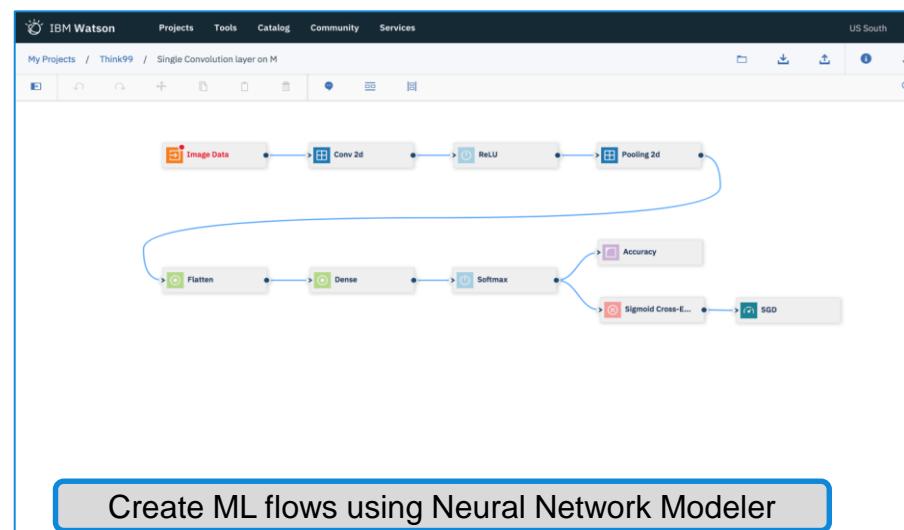
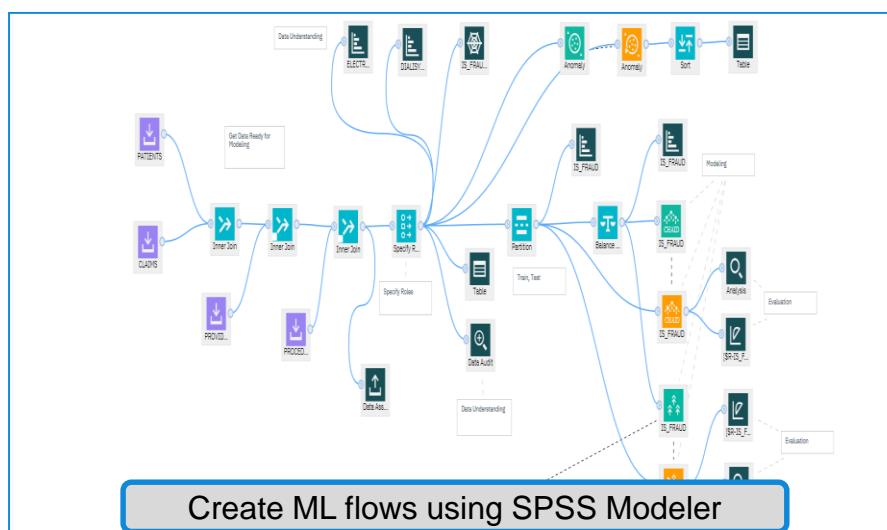
A prominent button at the bottom center says "Connect to Data Sources".

This screenshot shows the Watson Studio interface with two open notebooks:

- RStudio:** A code editor window titled "Format" containing R code to generate a bar chart. The code uses `matplotlib` to plot the number of tweets by country. A preview of the chart titled "Tweets Country Distribution based on the User Profile" shows bars for Germany, Mexico, Canada, India, Japan, and Spain.
- Jupyter:** A code editor window titled "Not Trusted | Python 3.5" containing Python code to draw insights from Twitter data. A preview of the chart is also visible.

Comments from users like "ARMAND RUIZ G... 7:49 AM Great work!" are visible on the right.

A blue box highlights the text "Open Source tools – Jupyter and RStudio".



# Watson Studio - Data Refinery

Making Data fit for use

The screenshot shows the Watson Studio Data Refinery interface. On the left, a list of operations is displayed, each with a step number, ID, and description. The operations include arranging data by product type, counting distinct items, filtering eyewear, grouping by tent, mutating cooking gear, and mutating all rows. A blue box highlights the text "Self-service data refinement and cleaning".

Step	ID	Description
1	TR9304	arrange
2	TR1537	count
3	TR1790	distinct
4	TR3101	filter
5	TR2045	group_by
6	TR3241	mutate
7	TR1764	mutate_all
8	TR3963	mutate_all
9	TR2608	Camping Equipment
10	TR1959	Personal Accessories
11	TR4372	Personal Accessories
12	TR1640	Personal Accessories
13	TR1559	Personal Accessories
14	TR4627	Mountaineering Equ...
15	TR3314	Personal Accessories
16	TR7150	Camping Equipment
17	TR3301	Personal Accessories

The screenshot shows the Watson Studio Data Refinery interface with a focus on profiling. It displays frequency distributions for CUST\_ORDER\_NUMBER, COUNTRY, STATE, and CITY. Below these are statistics for length and unique values. A blue box highlights the text "Comprehensive profiling".

Column	Frequency	Statistics
CUST_ORDER_NUMBER	121098	Maximum length: 6
CUST_ORDER_NUMBER	121098	Minimum length: 6
CUST_ORDER_NUMBER	121098	Mean length: 6.2778
CUST_ORDER_NUMBER	121098	Unique: 100
COUNTRY	121098	Maximum length: 9
COUNTRY	121098	Minimum length: 1
COUNTRY	121098	Mean length: 8.1706
COUNTRY	121098	Unique: 768
STATE	121098	Maximum length: 14
STATE	121098	Minimum length: 5
STATE	121098	Mean length: 10.2778
STATE	121098	Unique: 100
CITY	121098	Maximum length: 25
CITY	121098	Minimum length: 3
CITY	121098	Mean length: 8.1706
CITY	121098	Unique: 768

The screenshot shows the Watson Studio Data Refinery interface with an "Interactive visualization" section. It displays a tree diagram where "COUNTRY, PRODUCT\_LINE" is the source node. The branches lead to "Camping Equipment", "Golf Equipment", "Mountaineering Equipment", and "Personal Accessories", which further branch to specific countries like Australia, Canada, Italy, and Spain. A blue box highlights the text "Interactive visualization".

The screenshot shows the Watson Studio Data Refinery interface with a "Scheduling and monitoring" section. It displays a summary of a data flow from "Great Outdoor Customer Orders.csv" to "Customer Orders\_shaped.csv" through 3 steps. Below this is a table of scheduled runs from March 6 to 11, 2018, with details like date, day, time, start, interval, and end. A blue box highlights the text "Scheduling and monitoring".

Run	Date	Day	Time	Start	Interval	End
1	6 Mar 2018	Tue	9:58 pm	6 Mar 2018 9:58 pm	Every 1 day	11 Mar 2018 9:58 pm
2	7 Mar 2018	Wed	9:58 pm			
3	8 Mar 2018	Thu	9:58 pm			
4	9 Mar 2018	Fri	9:58 pm			
5	10 Mar 2018	Sat	9:58 pm			
6	11 Mar 2018	Sun	9:58 pm			

# Watson Studio – Watson Machine Learning

*Simplifying deployment and management of ML models in production*

The screenshot shows the Watson Studio interface with the 'Watson Studio Labs' project selected. In the 'Train' tab, the 'Select a technique' step is active. It displays a dropdown for 'Column value to predict (Label Col)' set to 'survived (Integer)'. Below it, 'Feature columns' are listed: 'pclass (Integer), sex (String), sibsp (Integer), parch (Integer), embarked (String), Age\_Bucket (Integer)'. Three algorithm options are shown: 'Binary Classification' (for two distinct categories), 'Multiclass Classification' (for multiple categories), and 'Regression' (for continuous values). A callout box highlights the 'Binary Classification' section.

**Manual or Automatic algorithm selection for ML models**

The screenshot shows the Watson Studio interface with the 'AwesomeExperiment' project selected. The 'Training Runs' tab is active, displaying 9 runs in total, 4 GPUs in use, and a total running time of 16 hours 27 mins. Below this, sections for 'Queued', 'In Progress', and 'Completed' runs are shown. A callout box highlights the GPU usage information.

**Train neural network in parallel across NVIDIA GPUs**

The screenshot shows the Watson Studio interface with the 'Watson Studio Labs' project selected. The 'Deployments' tab is active, showing one deployment named 'MNIST Deployed1' with a status of 'DEPLOY\_SUCCESS' and a 'Web Service' type. A callout box highlights the deployment status.

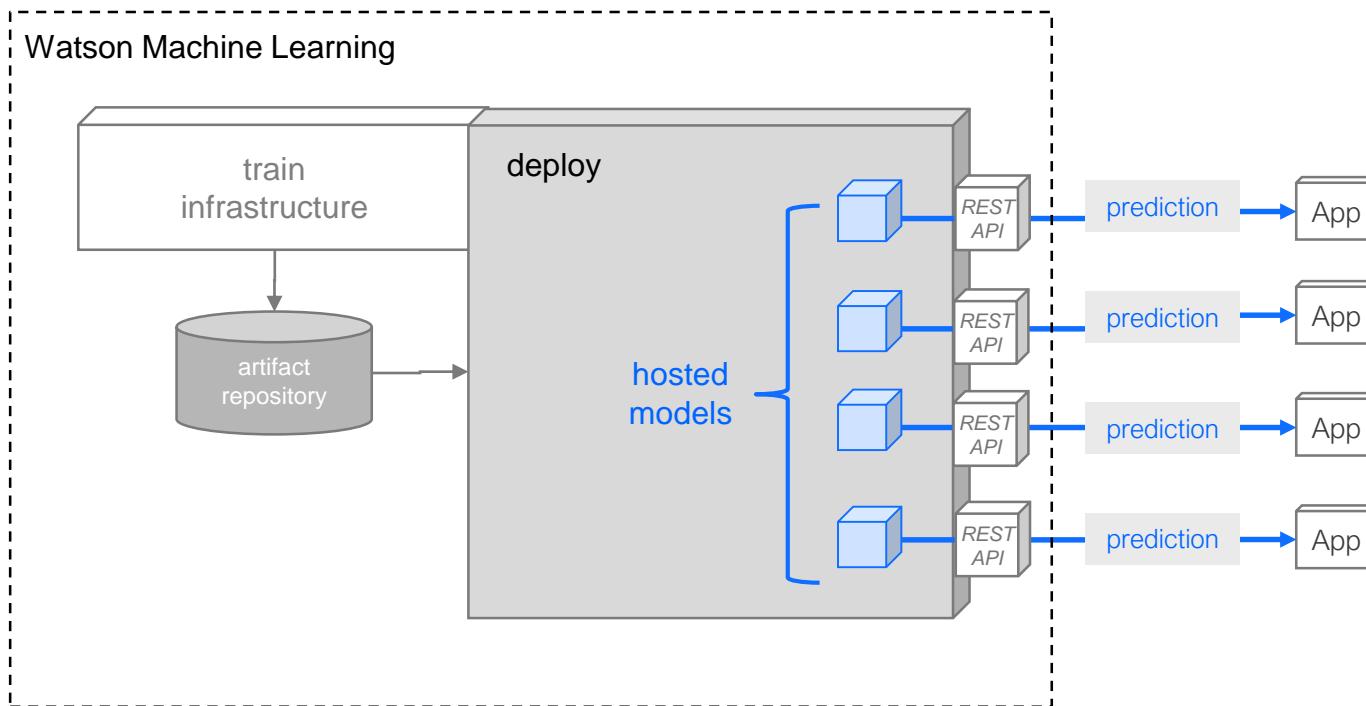
**Deploy models into production**

The screenshot shows the Watson Studio interface with the 'AwesomeProject' project selected. The 'Evaluation' tab is active, displaying evaluation events over the last 60 days. A callout box highlights the evaluation metrics.

**Monitor deployed models to evaluate performance**

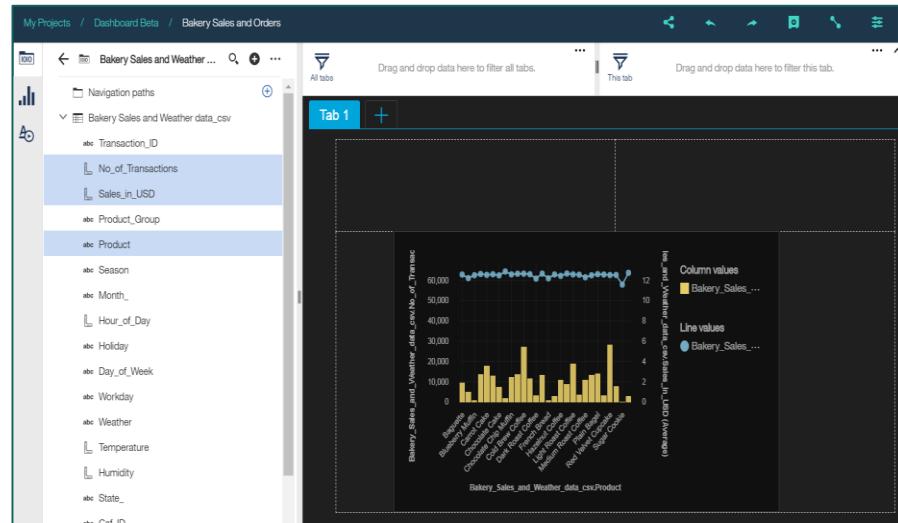
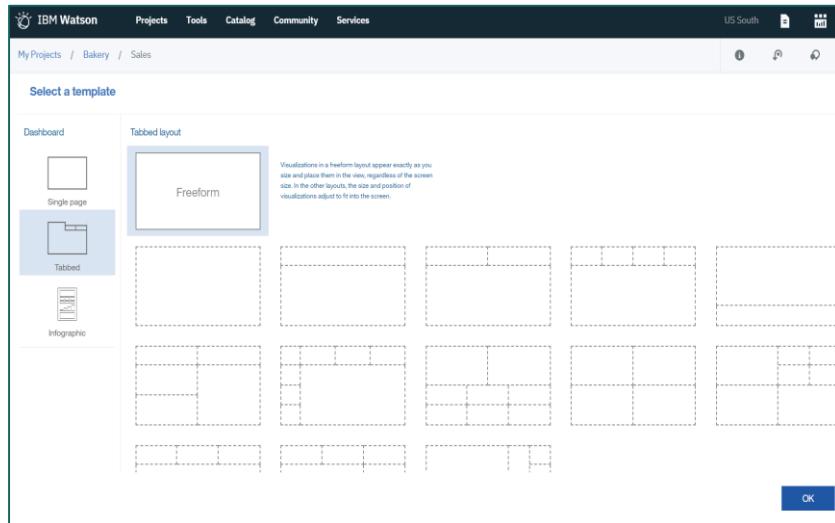
# Watson Studio- Deploying Trained Models

Deploy your models within Watson Machine Learning



# Watson Studio – Dynamic Dashboards

*Making insights available to all*



My Projects / Bakery Sales

Add to project

**Data assets**

0 assets selected.

NAME	TYPE	SERVICE	CREATED BY	LAST MODIFIED	ACTIONS
UNdata_agr_value_add.csv	Data Asset	Project	Alex Jones	7 Mar 2018, 9:37:13 am	⋮
EuropeanCountryStats.csv	Data Asset	Project	Alex Jones	7 Mar 2018, 9:37:12 am	⋮
Bakery Sales and Weather data.csv	Data Asset	Project	Alex Jones	8 Feb 2018, 3:07:05 pm	⋮

**Notebooks**

New notebook

NAME	SHARED	SCHEDULED	STATUS	LANGUAGE	LAST EDITOR	LAST MODIFIED	ACTIONS
Sales Predictions					Alex Jones	7 Mar 2018	⋮

**Streams flows**

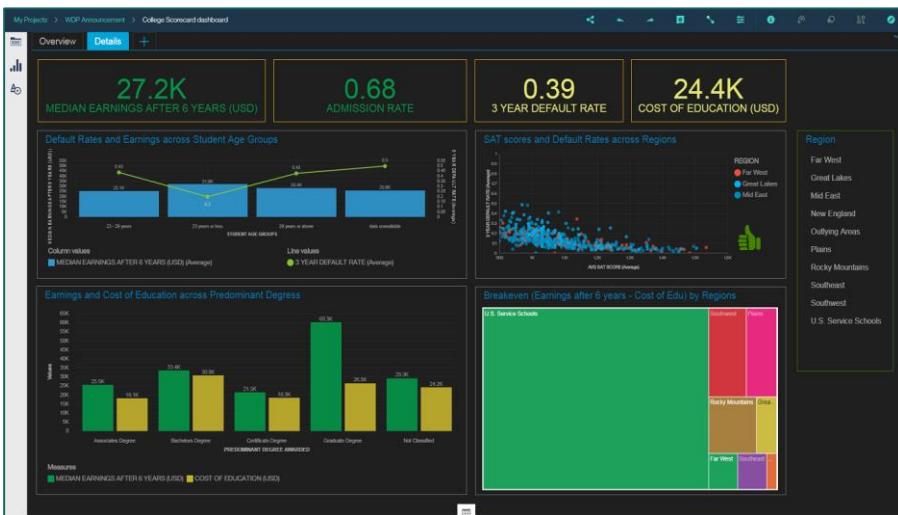
New streams flow

**Dashboard**

0 assets selected.

NAME	SHARED	LAST EDITOR	LAST MODIFIED	ACTIONS
Bakery Dashboard		Alex Jones	9 Feb 2018, 4:58:46 pm	⋮ Share Remove

**Models**



# Watson Knowledge Catalog

*Unlock tribal knowledge and unleash knowledge workers*

Browse Assets Usage Statistics Access control Settings

What assets are you looking for?

Recently Added

Data Asset US Airlines	Data Asset Harry Rosen example	Notebook Machine Learning using R	Data Asset FDIC Failed Bank List	Data Asset 2017 Small Business Banking Loans	Data Asset All US Banks
Owner: Jay Limbум Added: 1 Mar 2018 13:04 Airline	Owner: Michael Tucker Added: 28 Feb 2018 17:30 Sales Forecast	Owner: Jay Limbum Added: 20 Feb 2018 14:55 notebooks	Owner: Jay Limbum Added: 19 Feb 2018 14:25 banking	Owner: Jay Limbum Added: 19 Feb 2018 13:36 banking	Owner: Jay Limbum Added: 19 Feb 2018 13:36 banking

Filter

Asset types

- Asset (56)
- Notebook (4)
- Connector (2)
- Dashboard (2)

Tags

- discovered (24)
- SAMPLES (11)
- untagged (8)
- GOSALES (8)
- sales (5)
- banking (4)
- dsx (4)
- notebook (4)
- GOSALESDW (3)

Browse the Catalog

Browse Assets Usage Statistics Access control Settings

Total assets

Deleted assets

Added assets

Assets accessed

Asset Usage Statistics

Personal data

Data assets containing personal or restricted data

Personally Identifiable Information

10 View all

Operational policies

13 Data Governance Policies

16 Data Governance Rules

Automatic enforcement

1801 Views in March 2018 ▲ 58% from last month

Policy enforcements over time

Sensitive Personal Information

4 View all

MyCo Confidential

7 View all

Most enforced policies

POLICY NAME	ENFORCED	MOST COMMON OUTCOME
Protect PR data	5	Access Deny
Deny access to Sensitive information if Not authorized user	0	
Protect SPI	0	
All Sensitive Data must be restricted to non-US employees	0	
Hide top-secret	0	
Deny Access of confidential data to external users	0	
Data used by Finance must be scored over 80 percent quality	0	
Confidential Data only available to Great Outdoors employees only	0	

Governance and Insight Dashboard

Business Glossary / MyCo Confidential

Overview Related content

MyCo Confidential

All confidential data for our entire knowledge set

Term details

Creator: jay@uk.ibm.com  
Date created: 29 Sep 2017  
Last editor: IBMid-2700028UUU  
Last modified: 7 Mar 2018

Owner: Jay Limbum

Tags: Compliance | Protection

Associated classifier or term: Confidential

Views over time

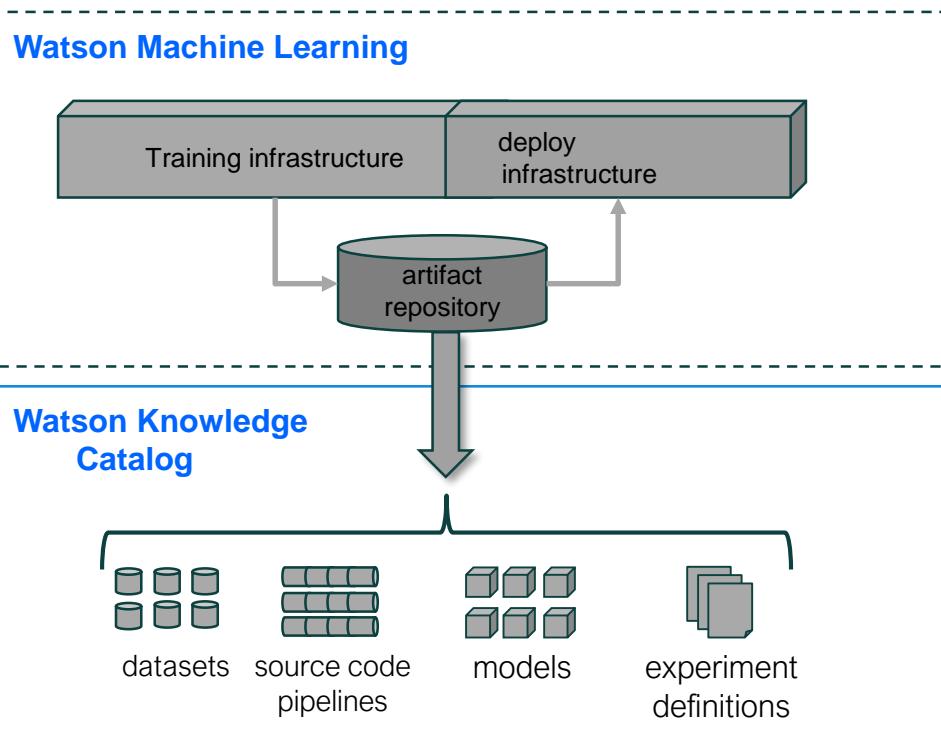
Description

A term used by Great Outdoors to classify any information that should not be disclosed outside of the Great Outdoors company.

Map Business Terms to Technical Assets

# Watson Studio Model Lifecycle Management

Use the Watson Knowledge Catalog and Watson Studio to manage your AI assets or manage them yourself



## Watson Machine Learning

Training infrastructure

deploy  
infrastructure

artifact  
repository

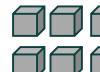
## Watson Knowledge Catalog



datasets



source code  
pipelines



models



experiment  
definitions

## Model Explanations

In May 2018, the General Data Protection Regulation (GDPR) takes effect and grants consumers the legal “right to explanation” from organizations that use algorithmic decision making.

## Audit Trails

Tracking prediction to each model's unique heritage is critical to regulatory compliance. Enforcing access controls for model sharing and deployment ensure ensures data security and application stability.

# Watson Studio Takeaways

## Integrated Collaboration Environment

- Data Scientists, Subject Matter experts, Business Analysts & Developers all in one environment to accelerate innovation, collaboration and productivity
- Built-in learning to get started or go the distance with advanced tutorials

## Choice of Tools for the full AI lifecycle

- Best in-breed open source and IBM tools that support the end-to-end AI lifecycle
- Choice of code or no-code tools to build and train your own ML/DL models or easily train and customize pre-trained Watson APIs

## Support for all levels of expertise

- Use Watson smarts and recommendations for the best algorithms to use given your data, OR
- Use the rich capabilities and controls to fine tune your models

## Experiment centric DL workflow

- Monitor batch training experiments then compare cross-model performance without worrying about log transfers and scripts to visualize results.
- You focus on designing your neural networks. We'll manage and track your assets.

## Model lifecycle & management

- Deploy models into production then monitor them to evaluate performance.
- Capture new data for continuous learning and retrain models so they continually adapt to changing conditions.

## Integrated with Knowledge Catalog

- Intelligent discovery of data and AI assets that enables reuse & improves productivity
- Seamlessly integrated for productive use with Machine Learning and Data science
- Powerful governance tools to control and protect access to data

# How does Watson Studio help fulfill the promise of your data?

## Data

Puts every important data source at the fingertips of the teams that need it wherever resides

## Governance

Enforces your policies without getting in the way of delivering insights

## Skills

Makes the most of the data professionals you have and helps them grow and learn from each other as a team

## Infrastructure

Brings all the tools in one place.  
Collaboration capabilities enables Data Science as a team sport.

# Labs

# Lab Overview

Work with IBM's Watson Studio in this Proof of Technology (PoT) to build, train, and test machine learning/deep learning models. Participants will be led through the following four hands-on labs:

- **Lab-1:** The first lab will use Jupyter Notebooks and the XGBoost library to apply machine learning to a classification problem in the healthcare profession. The Watson Machine Learning API will then be used to save and deploy the model.
- **Lab-2:** The second lab will demonstrate Watson Machine Learning capabilities to simplify the building and deployment of machine learning models. The ability to monitor and adjust the deployed model will be demonstrated via the continuous learning capability of Watson Studio.
- **Lab-3:** The third lab will feature the new Watson Studio Neural Network modeler, and Experiment Assistant to build, train, and test a Convolutional Neural Network to classify images.
- **Lab-4:** The fourth lab consists of 4 “sub-labs” each working on the same dataset that demonstrate (4a) Watson Machine Learning deployment of a Machine Learning model, and DevOps to build an application that invokes the deployed model, (4b) Visual Drag and Drop creation of a machine learning model pipeline, (4c) data profiling, visualization, and preparation using the Data Refinery, and (4d) Spark Machine Learning via Jupyter Notebooks.

# Lab Tips

- Labs are all located in [www.github.com/bleonardb3/ML\\_POT\\_9-6](https://www.github.com/bleonardb3/ML_POT_9-6) repository. Environment set up is located in the repository **README** file. We will jointly walk through these steps.
- Instructions for each Lab are in the **README** file in the respective Lab folder.
- Cloud development enables making frequent improvements in the user interface. We reviewed the lab instructions and made screen updates so they should be pretty faithful to the user interface. Small differences may occur but shouldn't get in the way of successfully completing the labs.
- You need to download the pdfs that are linked to the instructions for Lab-2, Lab-3, Lab-4a, Lab-4b, and Lab-4c. Otherwise, the links in the pdf will not work when viewing in the github interface. Please follow the instructions to click on the link and then click on the Download button.
- Do not use Internet Explorer as the browser. For Mac users do not use Safari.
- For the Jupyter Notebook labs, you execute notebook cells by entering <Shift><Enter> when your cursor is in a code cell. Or you can click on the Run icon in the toolbar.
- All of the Labs should be done in the project that you created when following the signup instructions.

# Lab-1 Heart Disease Detection

In this lab, you will use a Jupyter Notebook to train a model using the XGBoost library to classify whether a person has heart disease or not. In addition to training, the notebook also explains how to persist a trained model to the IBM Watson Machine Learning repository, deploy the model as a REST service and then predict using the deployed model.

In this lab we will:

- Use open source data set published in the University of California, Irvine (UCI) Machine Learning Repository.
- Load a CSV file into a Pandas DataFrame.
- Explore data using Pixiedust
- Prepare data for training and evaluation.
- Create, train, evaluate a XGBoost model
- Visualize the importance of features that were used to train the model.
- Use cross-validation to select the optimal hyperparameters based on a parameter grid.
- Persist best model in Watson Machine Learning repository using Python client library.
- Deploy the model for online scoring using the Watson Machine Learning's REST APIs
- Score sample data using the Watson Machine Learning's REST APIs.

# Lab-2 Predict Building Inspection Failure

Using 2017 Chicago building data, we will make Chicago a safer place by building a model to predict when buildings are likely to fail inspection. We can then use our model to find which buildings are most dangerous and attend to those first.

In this lab we will:

- Use Watson Machine Learning to train, compare, and select the best machine learning model for our use case.
- Set up continuous learning capabilities.
- Deploy our machine learning model to make it available to external services.

# Lab-2 Continuous Learning Overview

Continuous Learning in the context of machine learning is the ability to adapt a model to the changing external world through autonomous incremental development. As new data is available, it is useful for a model to automatically retrain to ensure that systems and applications dependent on our model stay as up to date as possible.

My Projects / Continuous Learning Lab / Building\_Violations\_Chicago\_2017

Last Evaluation Result

Version	e2a674a9-e916-4eb0-81d3-702ce33fa9ba
Phase	training
AreaUnderPR	0.738

Performance Monitoring [Edit configuration](#)

Performance Metrics (Threshold)	areaUnderPR (0.8)
Feedback Data Source	dashdb: BLUDB / New2017Table
Record Count Required For Re-Evaluation	500
Auto Re-Train	conditionally
Auto Re-Deploy	never

Versions

TIME	VERSION	DEPLOYED	AREAUNDERPR	ACTIONS
16 Apr 2018 03:03pm	e2a674a9-e916-4eb0-81d3-702ce33fa9ba		0.738	
16 Apr 2018 02:57pm	4cc6abb1-f3be-4e3b-b26d-b9c2dc67abec		0.708	
16 Apr 2018 02:51pm	66245399-4be3-470d-b190-c849a076947a		0.851	 

# Lab 3 – Recognizing Handwritten Digits

This lab will use the [MNIST](#) computer vision data set to train a convolutional neural network (CNN) model to recognize handwritten digits. The Watson Studio neural network flow editor, Watson Studio experiment builder and the Watson Machine Learning component will be used to build, train, and save the trained model.

## Objectives:

- Upon completing the lab, you will know how to:
  - Create Cloud Object Storage buckets to contain the input and result files
  - Create a neural network design from an example using the flow editor
  - Use the experiment builder used to set up a training definition to train the neural network model
  - Monitor the training progress and results.
  - Save and Deploy the trained model.
  - Test the model

# Lab-3 Neural Network Modeler

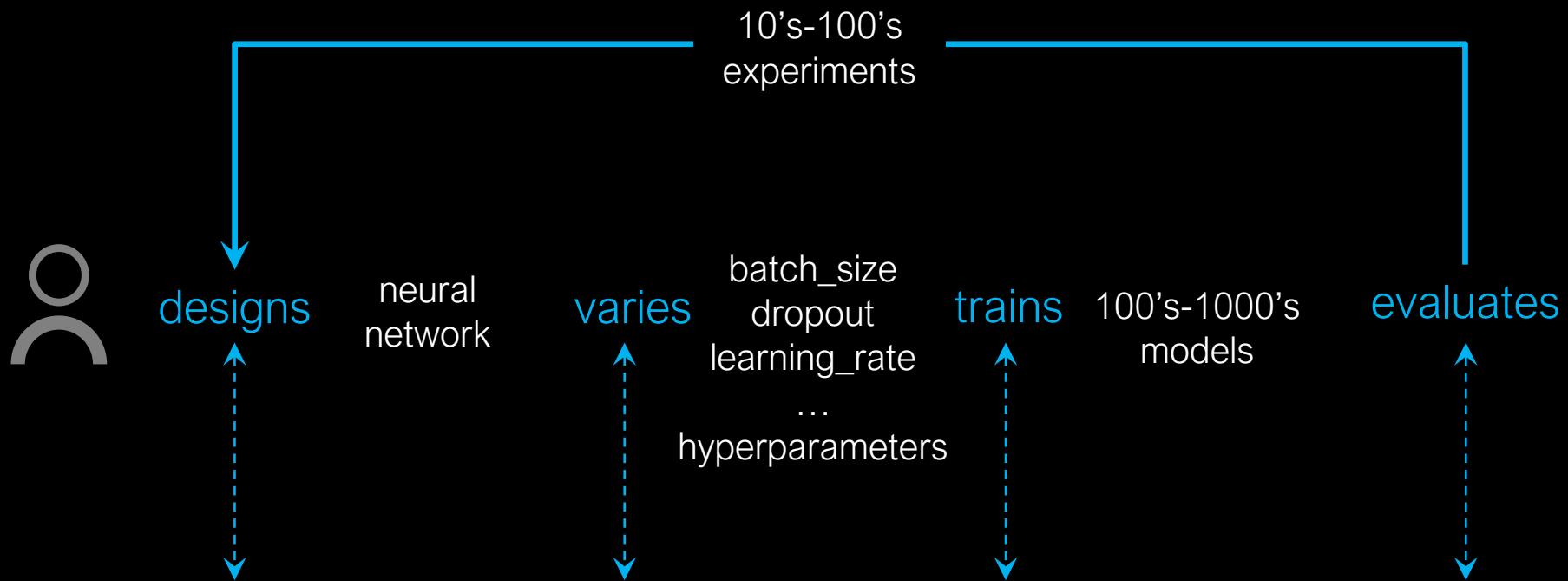
An intuitive drag-and-drop, no-code interface for designing neural network structures using the most popular deep learning frameworks. Quickly capture your network design then single click export for experimental optimization.



The screenshot displays the Deep Learning Editor interface. On the left, a sidebar lists nodes: Input, Activation, Convolution, Core (Flatten, Dense), Metric, and Loss. The main area shows a neural network flowchart starting with 'Image Data' input, followed by 'Conv 2d', 'ReLU', and 'Pooling 2d' layers, which then branch into two parallel paths. Each path contains 'Conv 2d', 'ReLU', and 'Pooling 2d' layers. The final output of the bottom path is 'Softmax With L...'. A callout 'Real-time validation of network flow' points to the flowchart. Another callout 'Drag-and-drop network layers' points to the sidebar. A detailed configuration panel on the right is shown for a 'Dense' layer, featuring sections for Weight Regularizer (L1, L2, L1-L2, null), Weight LR Multiplier (set to 1), Bias Constraint (maxnorm, nonneg, unifnorm, null), Bias Regularizer (L1, L2, L1-L2, null), Bias LR Multiplier (set to 1), Activity Regularizer (L1, L2, L1-L2, null), and a 'Save' button. Callouts point from the configuration panel to the 'Define layer configuration' and 'Choose optimizer params' sections below. A final callout points to a list of export options at the bottom.

- Generate CPU or GPU compatible code
- Save as popular framework code
- Export as a python notebook
- Execute as batch experiment

# Lab-3 Experiment Assistant



Experiment Assistant  
supports the end-to-end workflow

# Lab-4 Predict Passenger Survival on the Titanic

For Lab-4, there are 4 “sub-labs” that you can select that show other features of Watson Studio. Each lab is based on the Titanic data set, often used in Kaggle competitions.

- [Lab-4a](#) - The first lab will use the Watson Machine Learning Model Builder capability to create a machine learning model based on the Titanic data set. The model will be deployed in the IBM Cloud, and an application will be built that uses the deployed machine learning model to predict survivability given passenger characteristics.
- [Lab-4b](#) - The second lab will guide participants in using the Watson Studio SPSS Modeler capability to explore, prepare, and model passenger data from the Titanic. The SPSS Modeler is a drag and drop capability to build machine learning pipelines.
- [Lab-4c](#) - The third lab features the Data Refinery tool, a fully managed self-service data preparation facility.
- [Lab-4d](#) - The fourth lab will leverage Spark machine learning (SparkML) in a Jupyter notebook to predict survivability using pyspark and a supervised learning model.