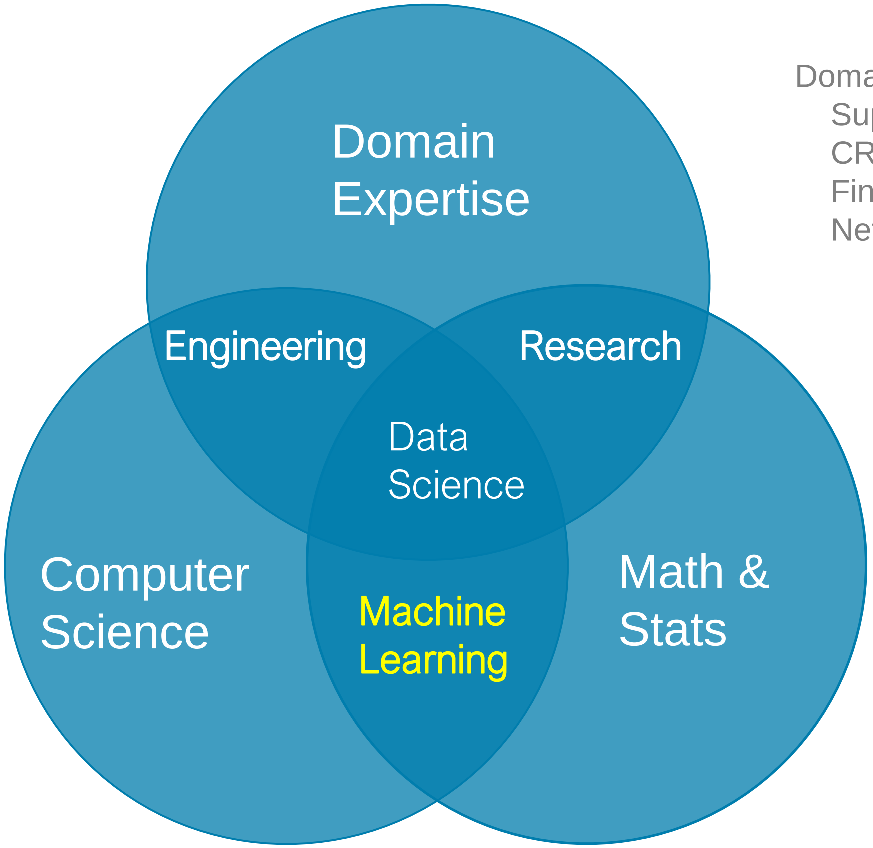
Introduction to Machine Learning

Agenda

8:30am - 9am	Breakfast, Socialize
9:00am – 10:15am	Introduction to Machine Learning Presentation
10:15am – 10:30am	Break
10:30am – 11:30am	Lab 1 - Machine Learning with XGBoost
11:30am – 12:30pm	Lab 2 – Continuous Learning with Watson Machine Learning
12:30pm – 1:30pm	Lunch
1:30pm – 2:30pm	Lab 3 - Neural Network Modeling and Deployment
2:30pm – 4:00pm	Lab 4 – Choice of Labs
4:00pm – 4:30pm	Wrap Up

Machine Learning and Data Science….

Domain Knowledge
- Supply Chain
- CRM
- Financials
- Networking

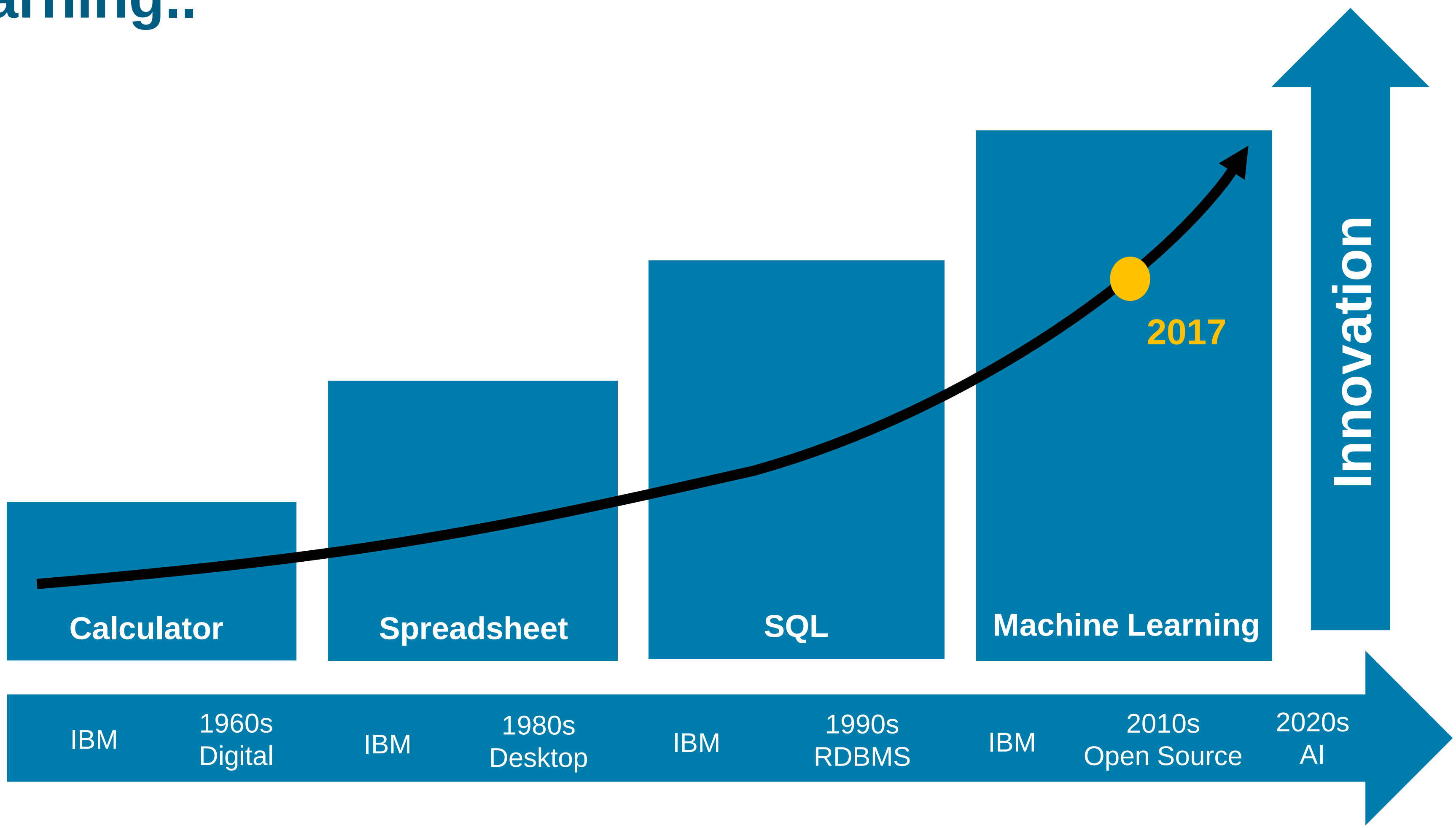


Scripting, SQL
Python, R Scala
Data Pipelines
Big Data/ Apache
Spark

Mathematics
Computational

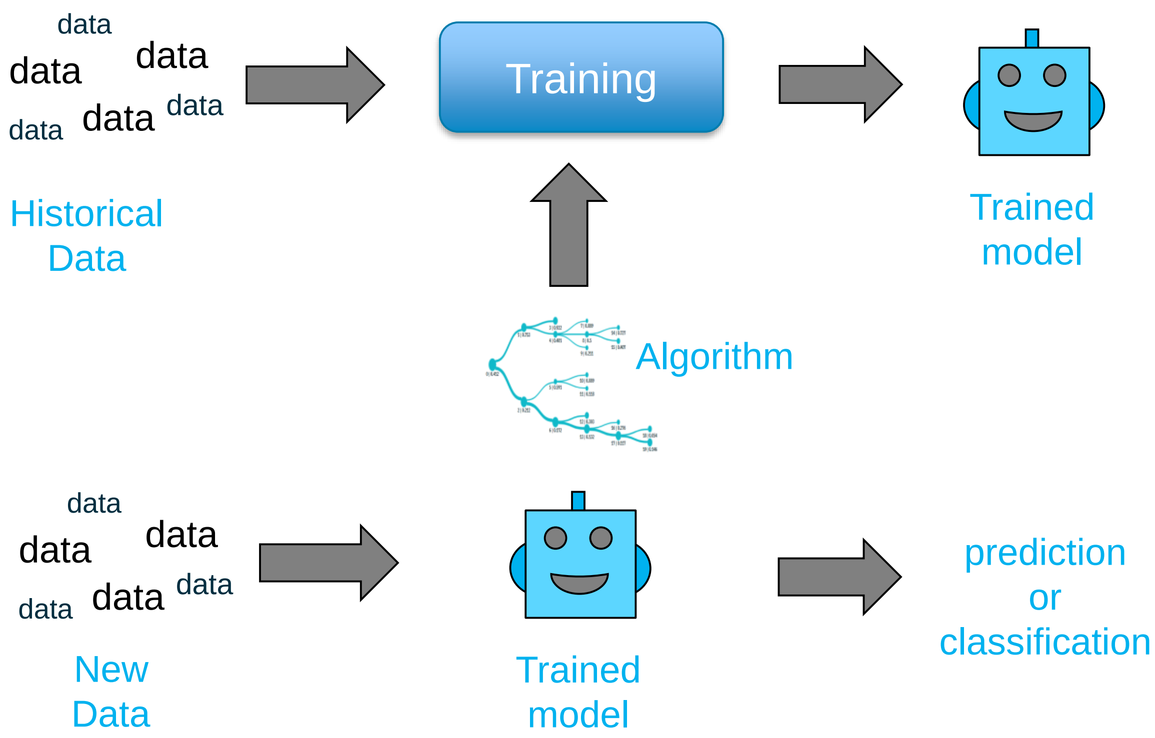
Data Science Projects Require Multiple Skills

Future of Data Science is Democratizing Machine Learning..



But what is Machine Learning?

*"Computers that learn without being **explicitly programmed**"*

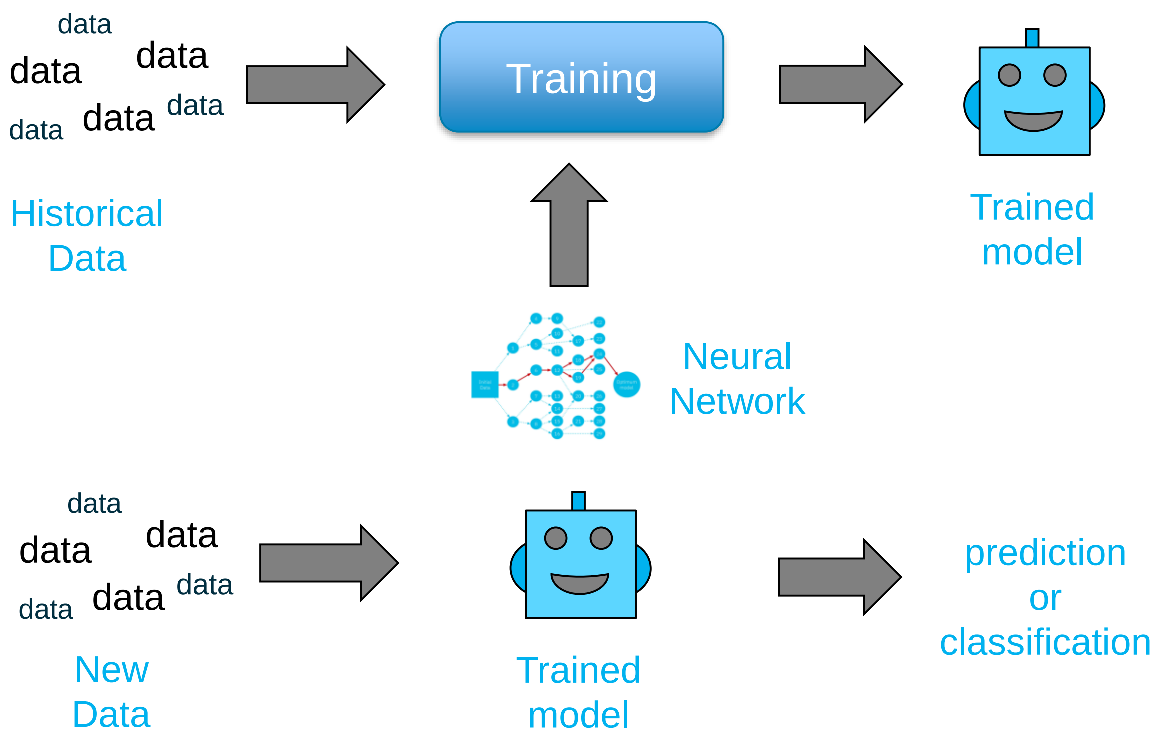


1. A machine learning model is trained to recognize patterns in historical data

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”



1. A machine learning model is trained to recognize patterns in historical data

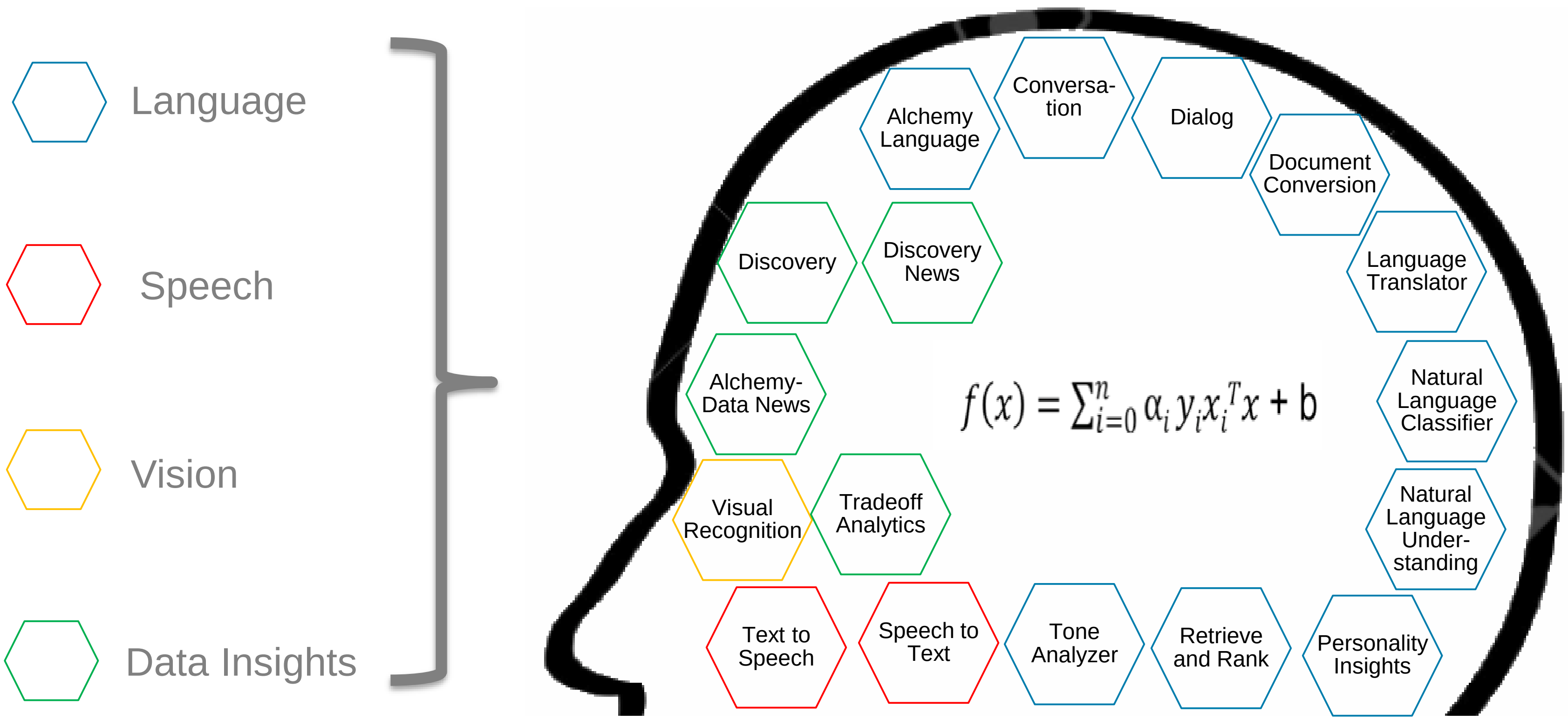
2. The model is then shown new data and asked to predict or classify it. If the patterns In the new data match the training data then the model makes accurate predictions.

But what is Artificial Intelligence?

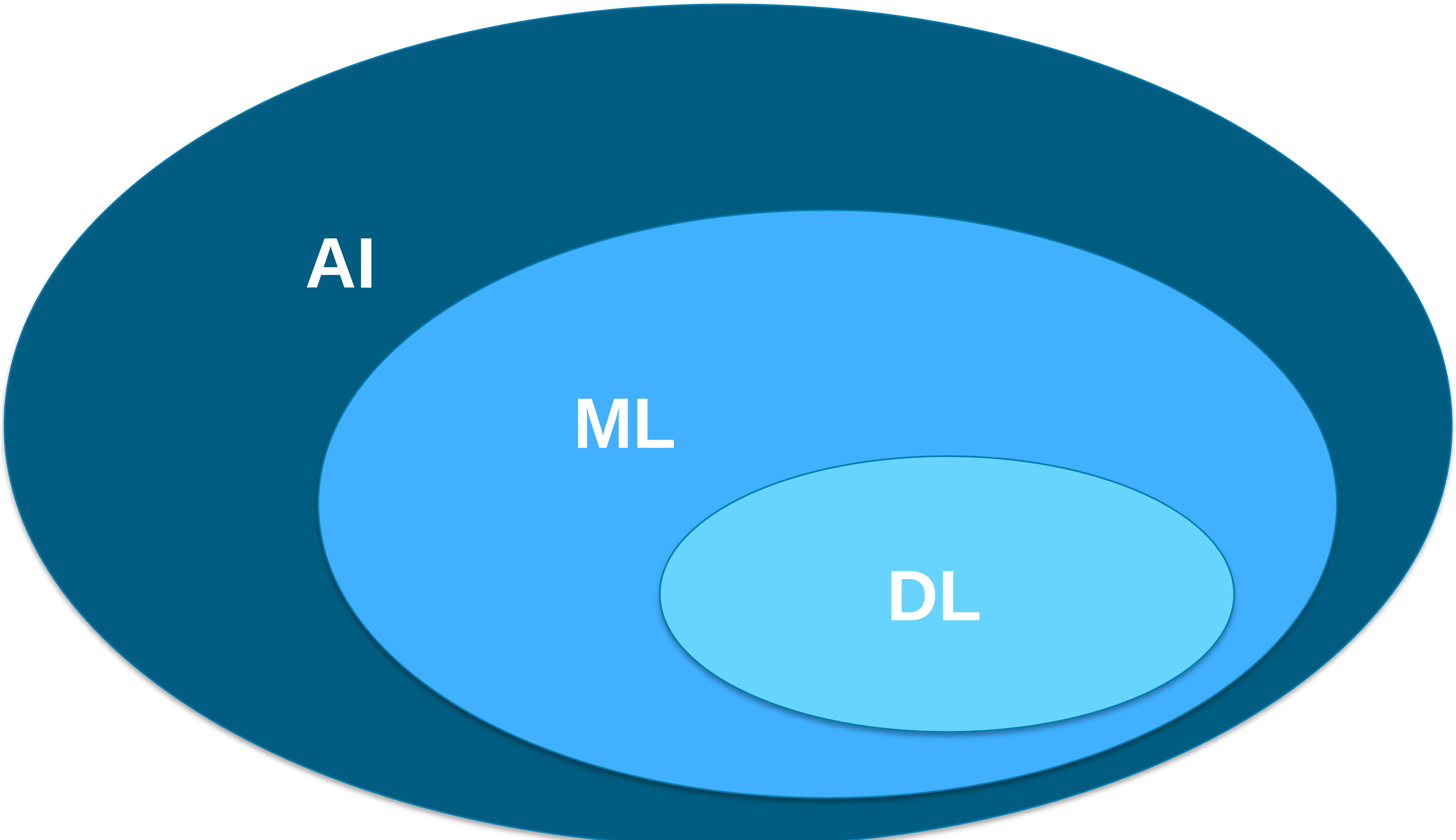
A theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages..

Machine Learning = Artificial Intelligence???

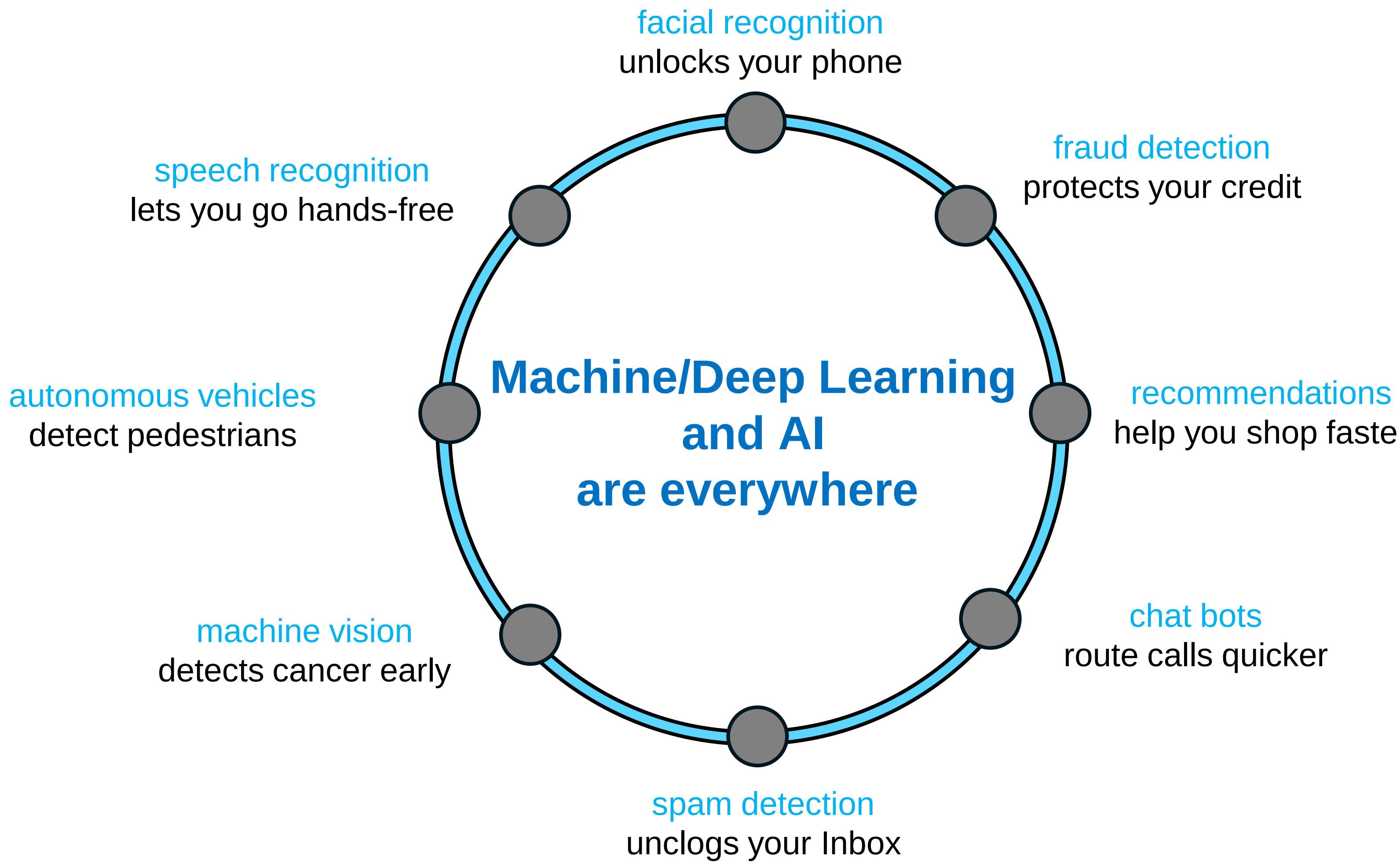
Data + Algorithms = Scored AI Models



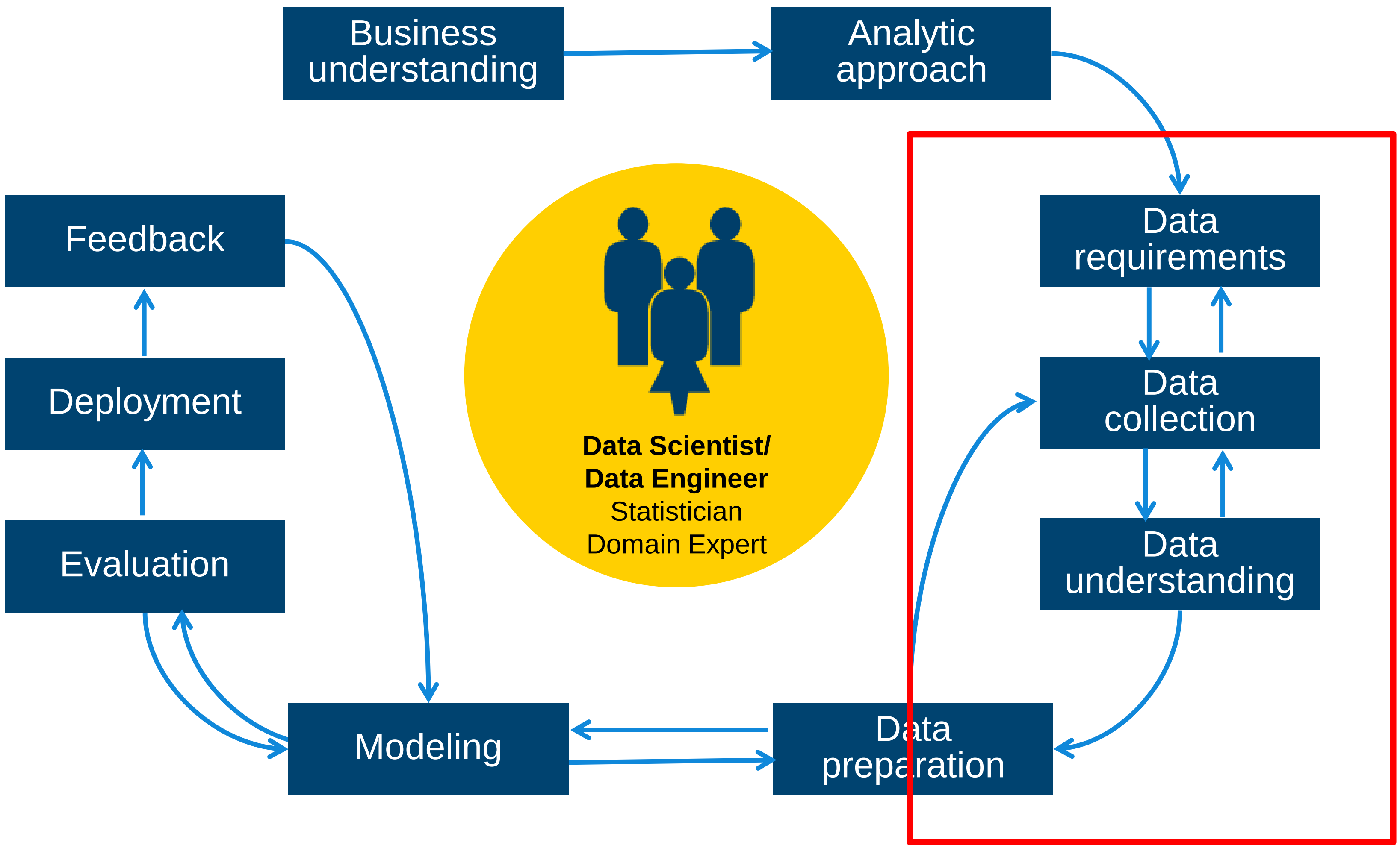
Understanding AI, ML & DL Relationship…



The future is now



Data Science Methodology

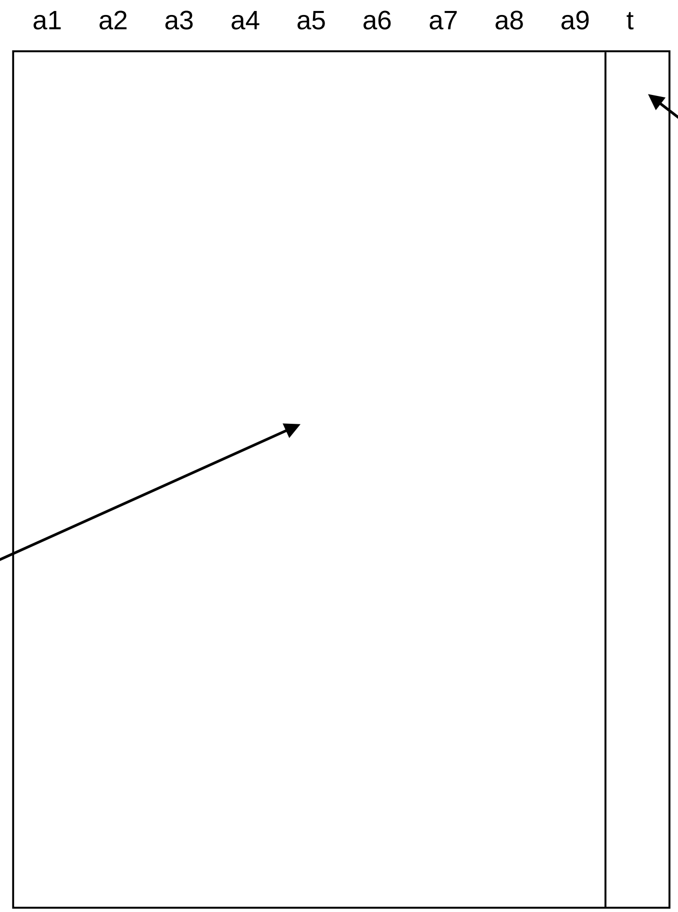


Matrix for Machine Learning

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…

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



Known as:
- Label
- Target variable
- Dependent variable

Scale or Categorical

Data Understanding – Data Audit

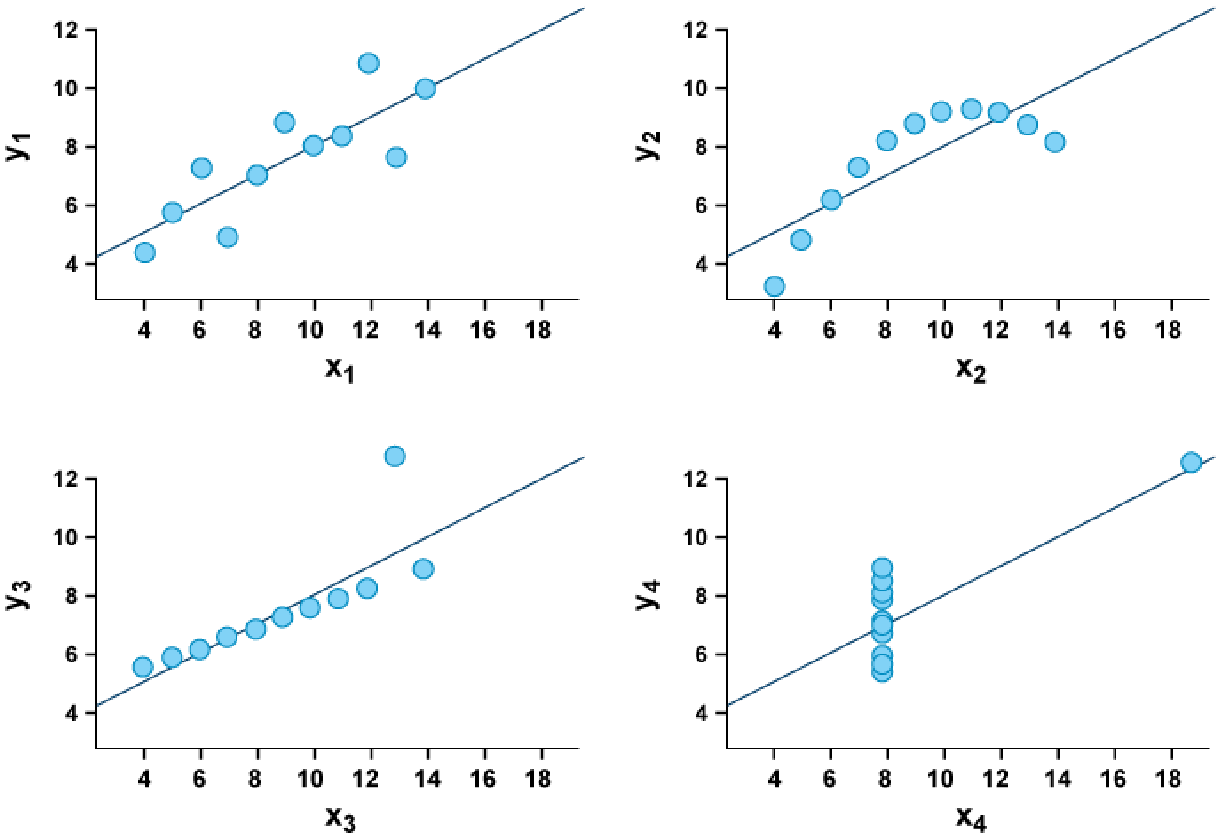
- **Data can be missing values**
  - Blank fields
  - Fields with dummy values (9999)
  - Fields with “U” or “Unknown”

- **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

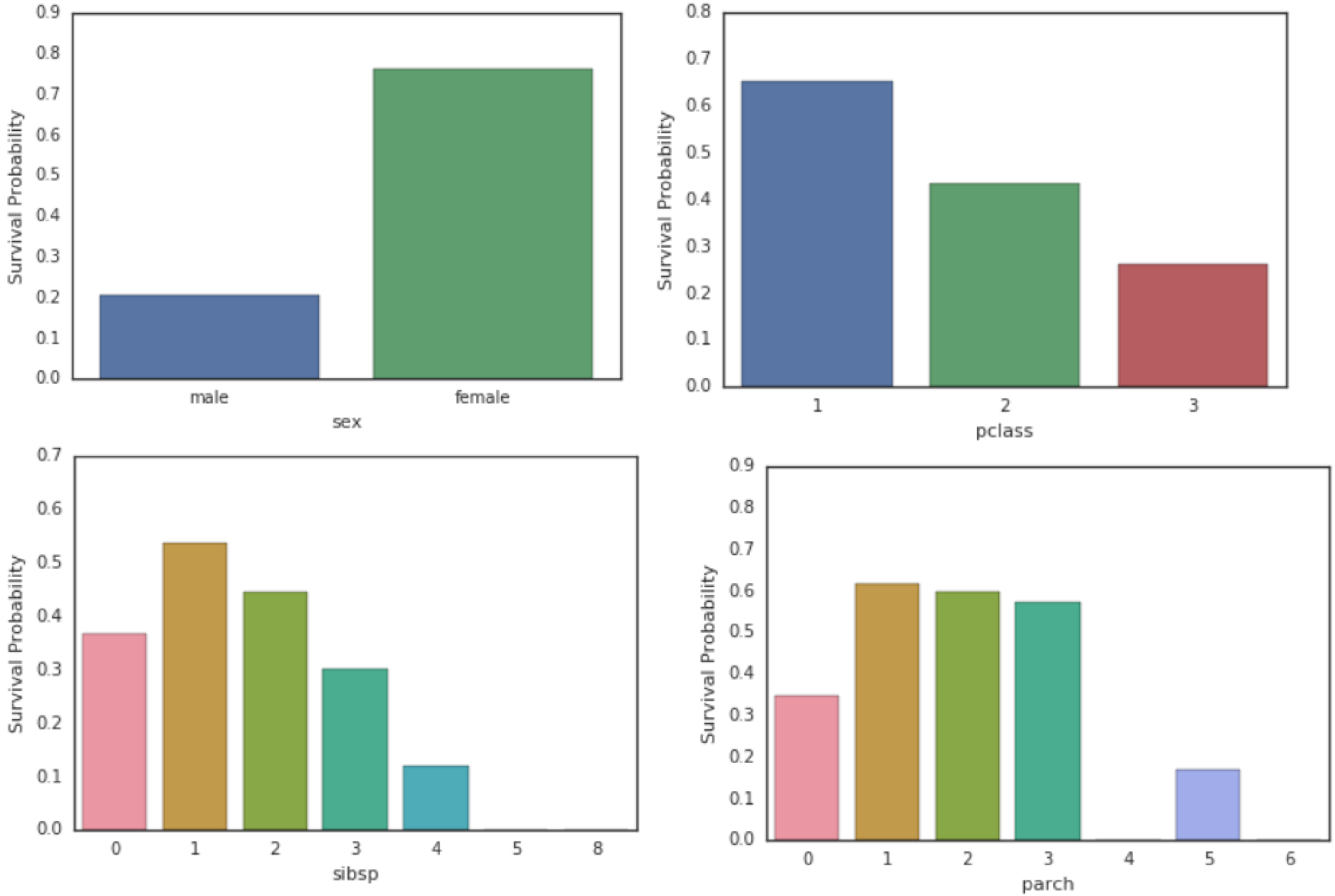


The four data sets have similar statistical properties:
•The mean of x is 9
•The variance of x is 11
•The mean of y is approx. 7.50
•The variance of y is approx. 4.12
•The correlation is 0.816
As shown the linear regression lines are approx. y=3.00+0.500x.

- **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

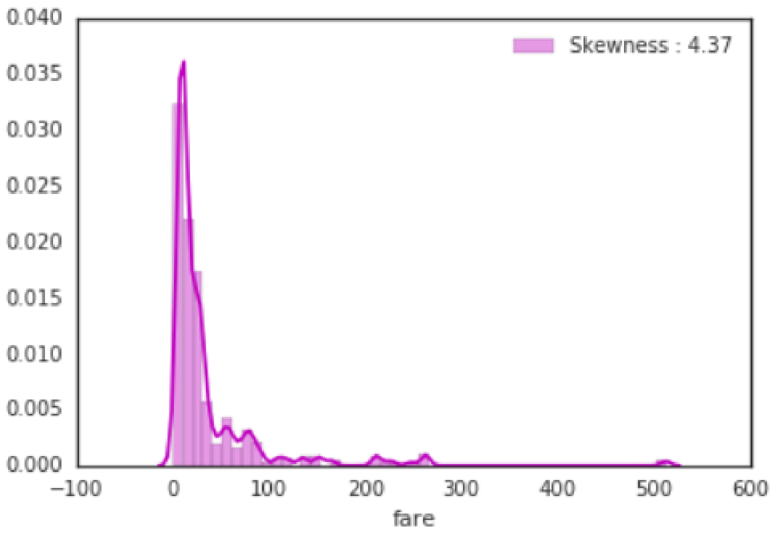
Data Understanding: Visualizations

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

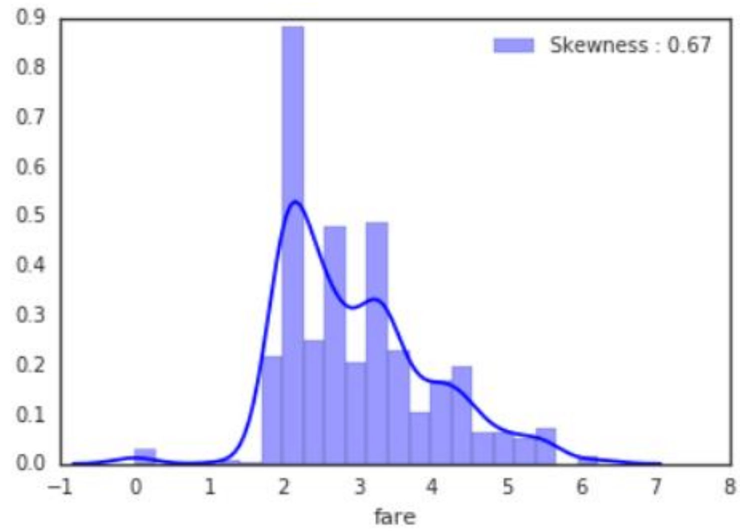


Data Understanding: Visualizations

- **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

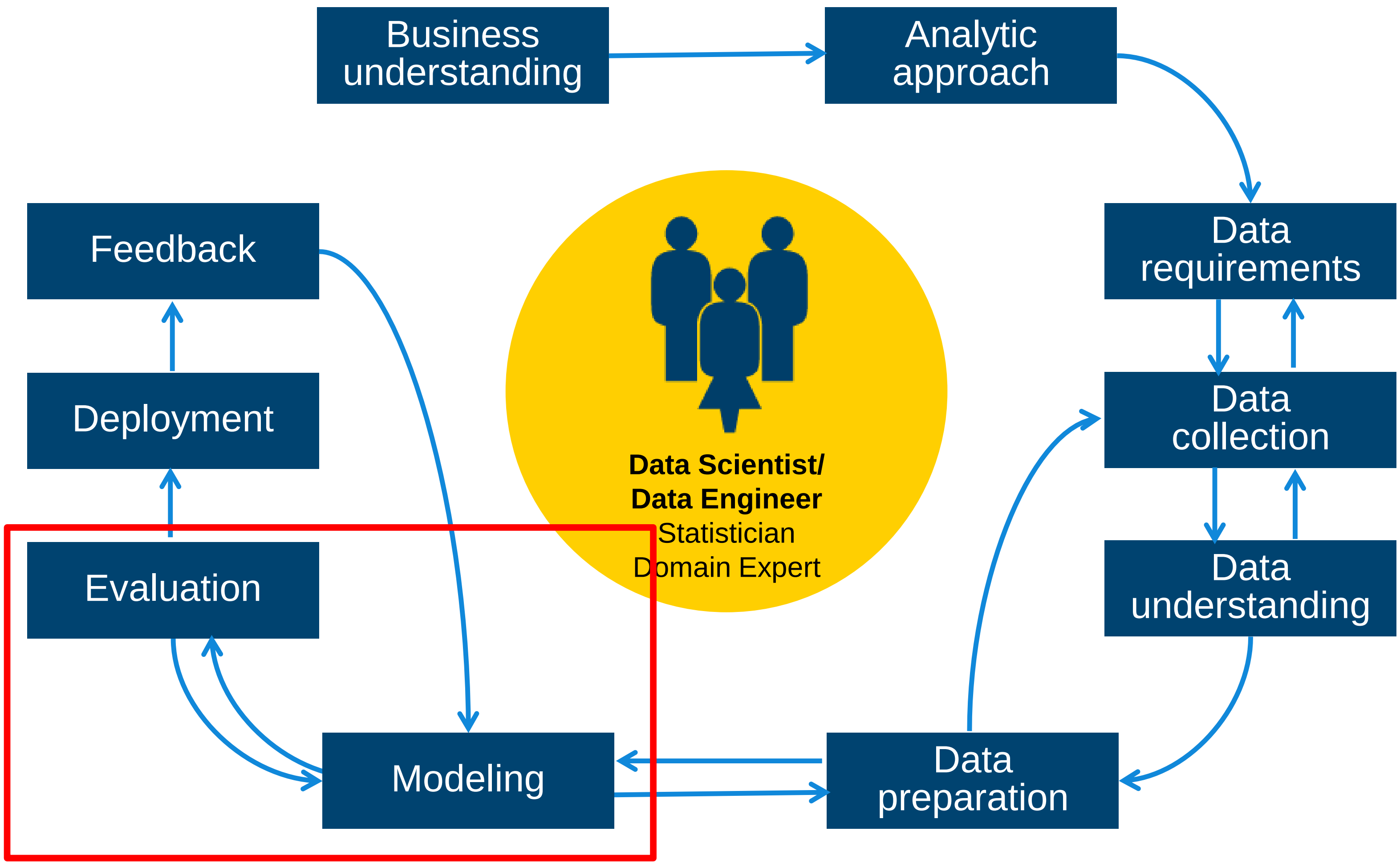
- **Data dimensionality may need to be reduced:**

- **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
    - Autoencoders

Data Science Methodology



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

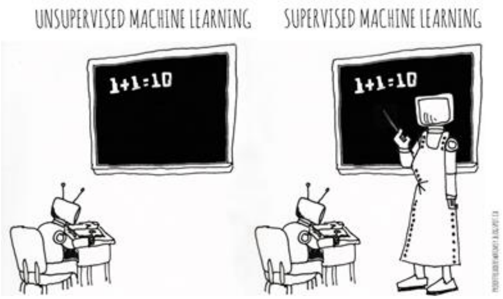
- Is this cancer ? (Medical diagnosis)
- Is this legitimate or fraud (spam) ?
- What is the market value of this house ?
- Which of these people are good friends with each other ?
- Will this engine fail (when) ?
- Will this person like this movie ?
- Who is this ?
- What did you say ? (Speech recognition)

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

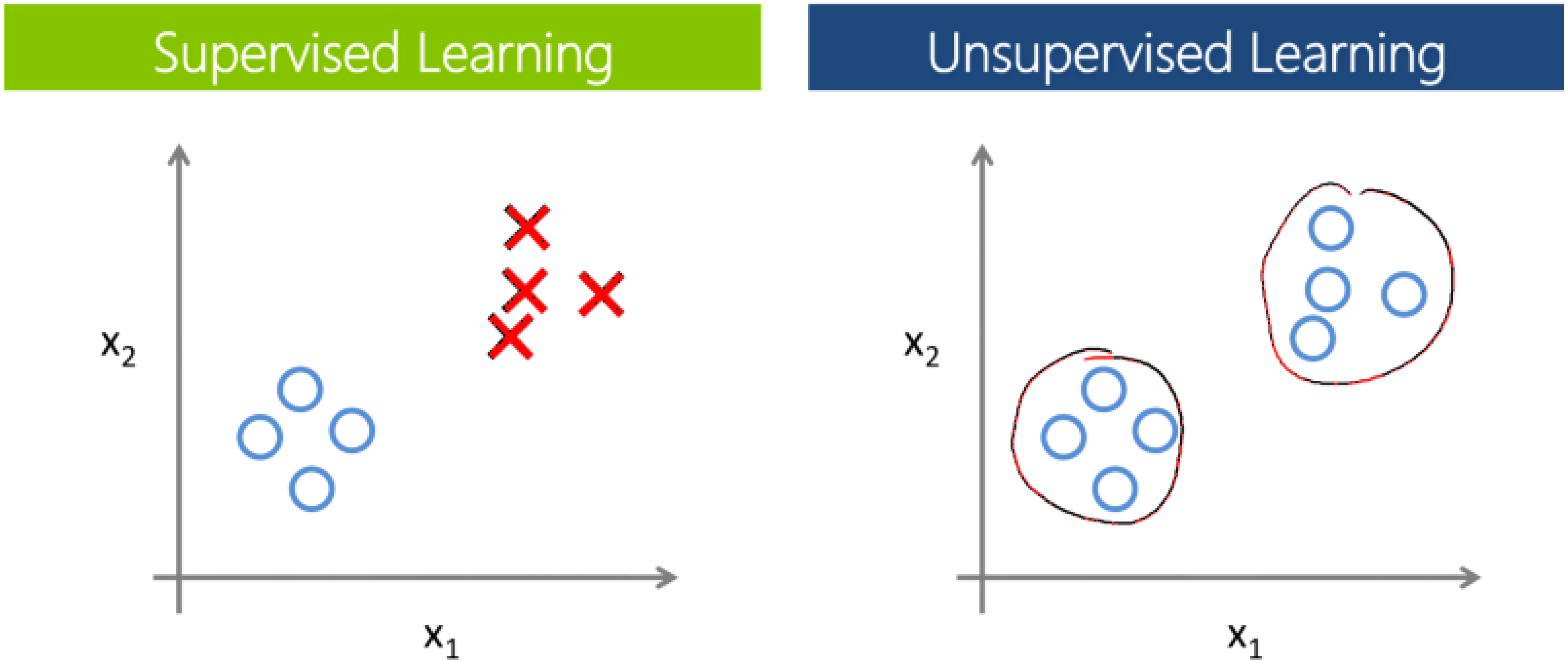
Categories of Machine Learning

- **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

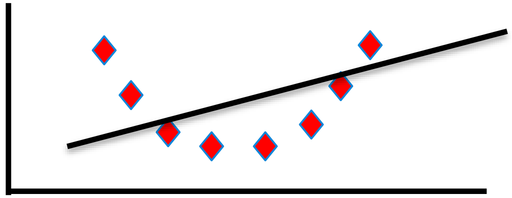
Technique	Usage	Algorithms
Classification (or prediction)	• Used to predict group membership (e.g., will this employee leave?) or a number (e.g., how many widgets will I sell?)	• Decision Trees • Logistic Regression • Random Forests • **Naïve Bayes** • Linear Regression • Lasso Regression etc
Segmentation	• 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

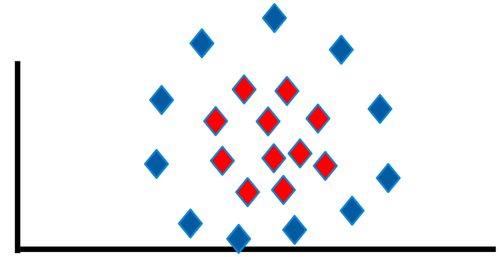
Categories of Machine Learning

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.

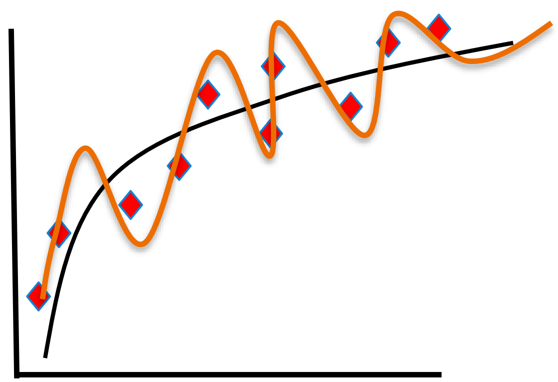
aX + BY + C





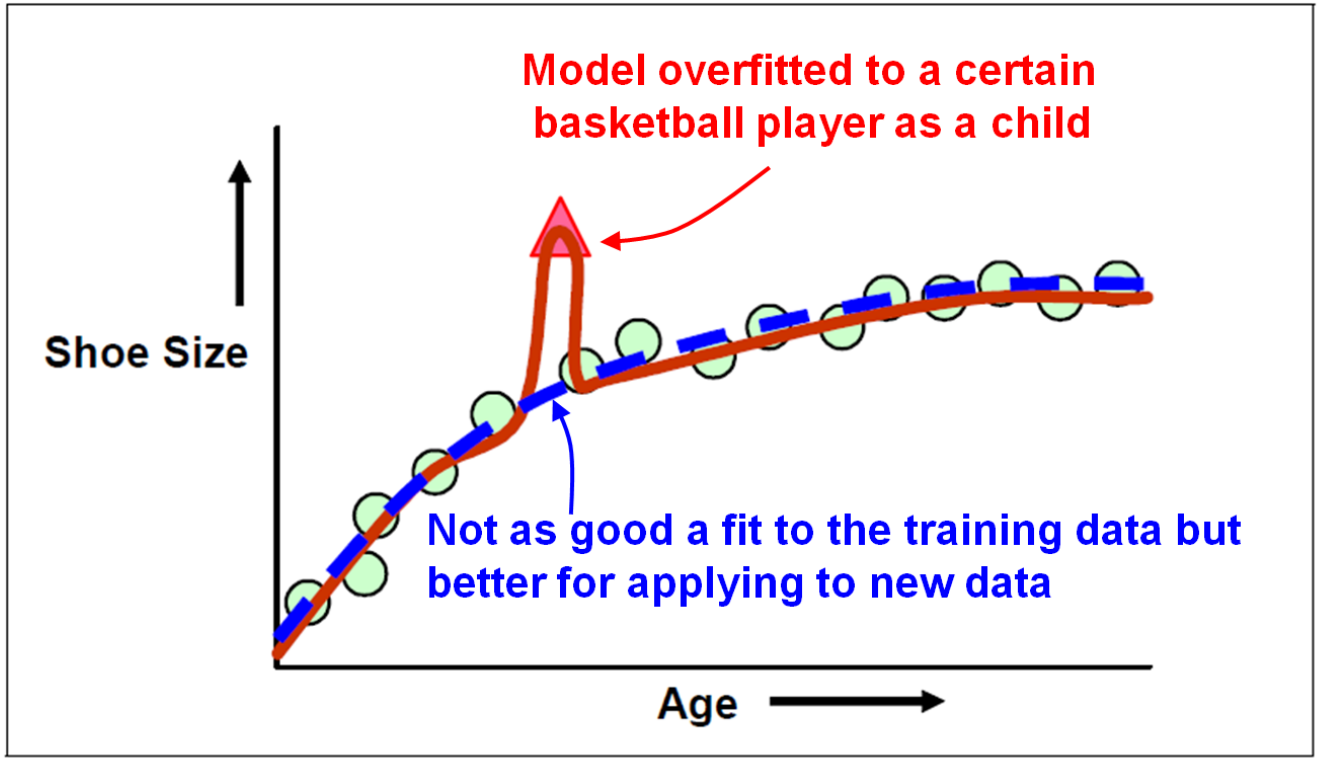
Learning challenges

- **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.



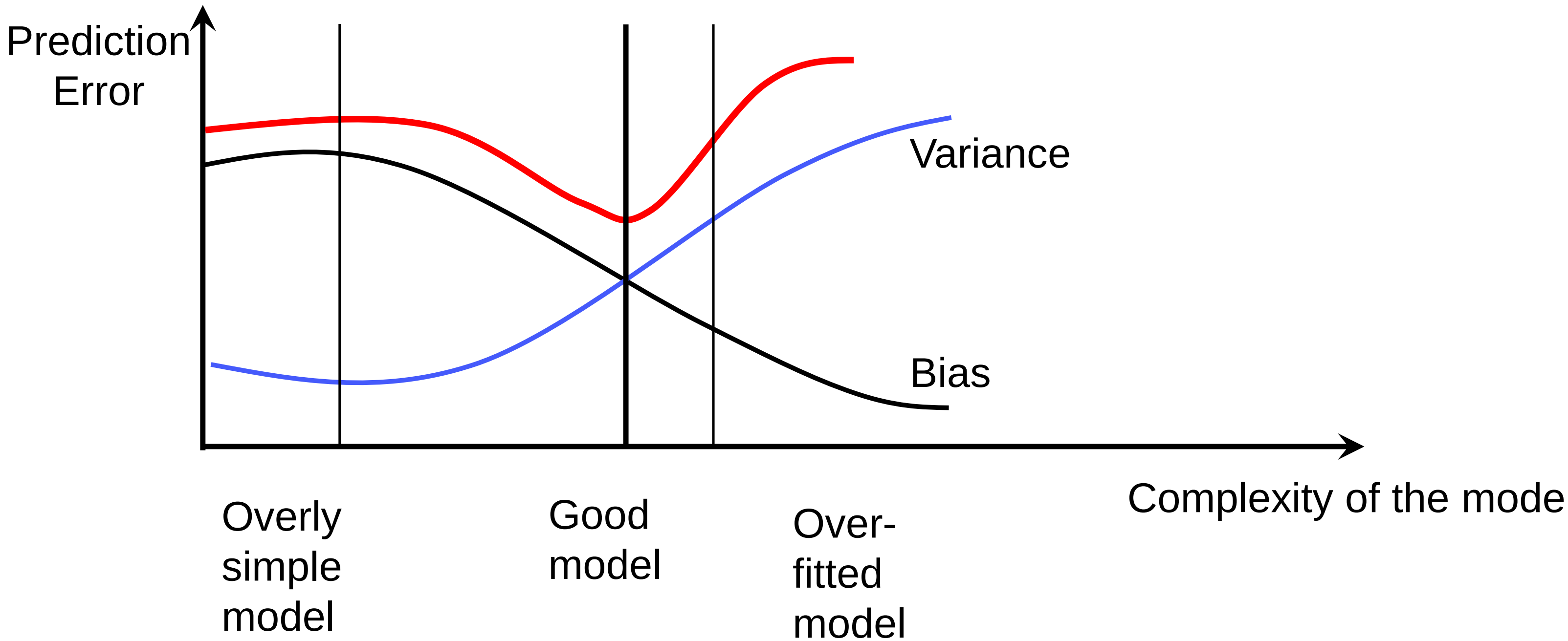
Model overfitting

- 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

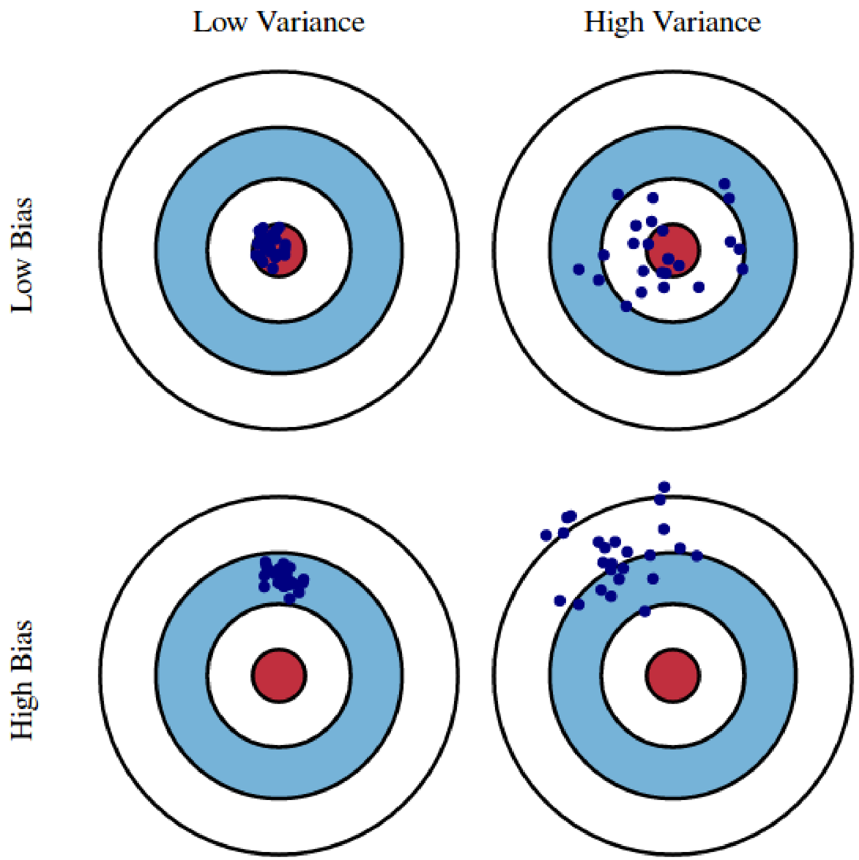


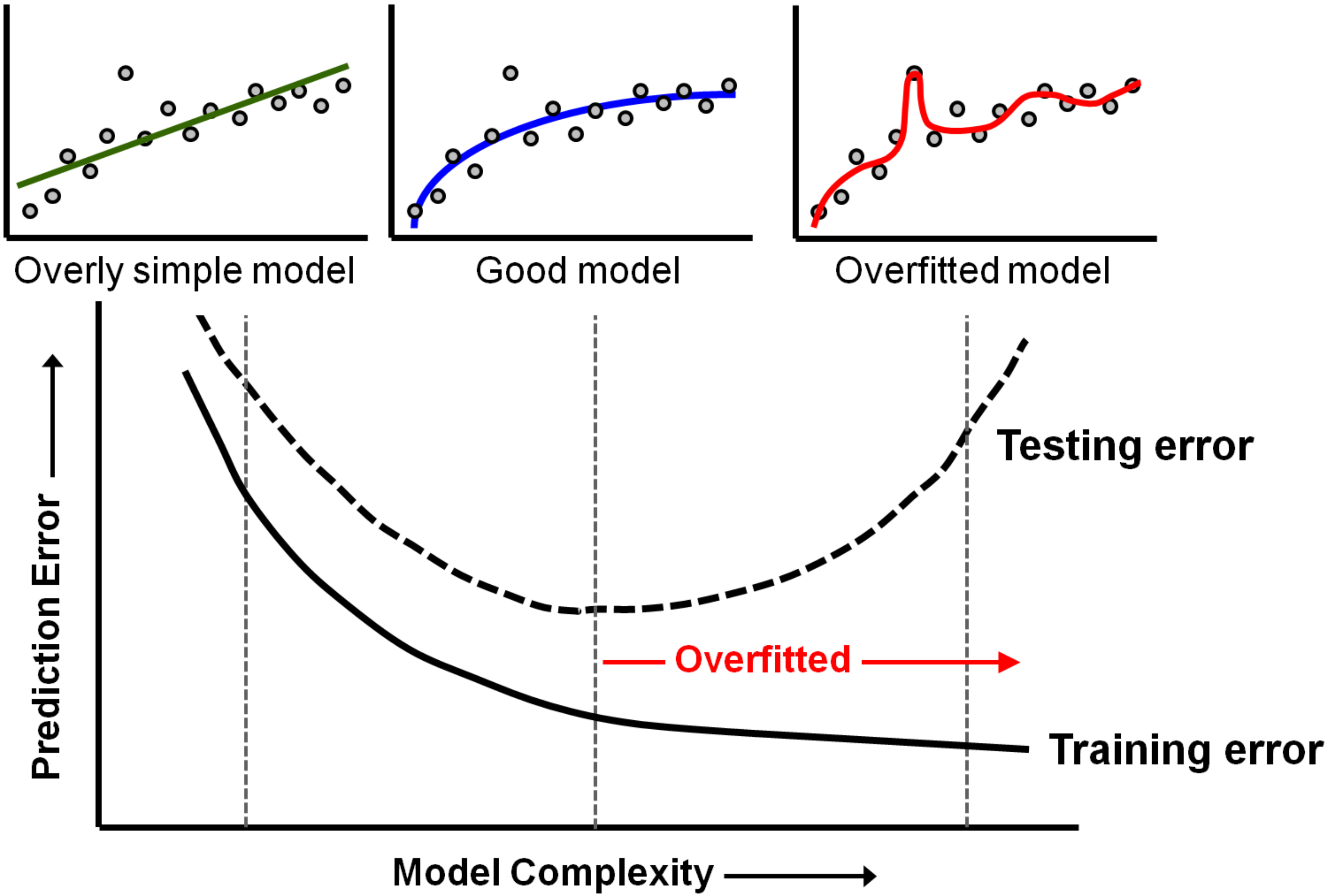
Fig. 1 Graphical illustration of bias and variance.

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



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)
  - 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

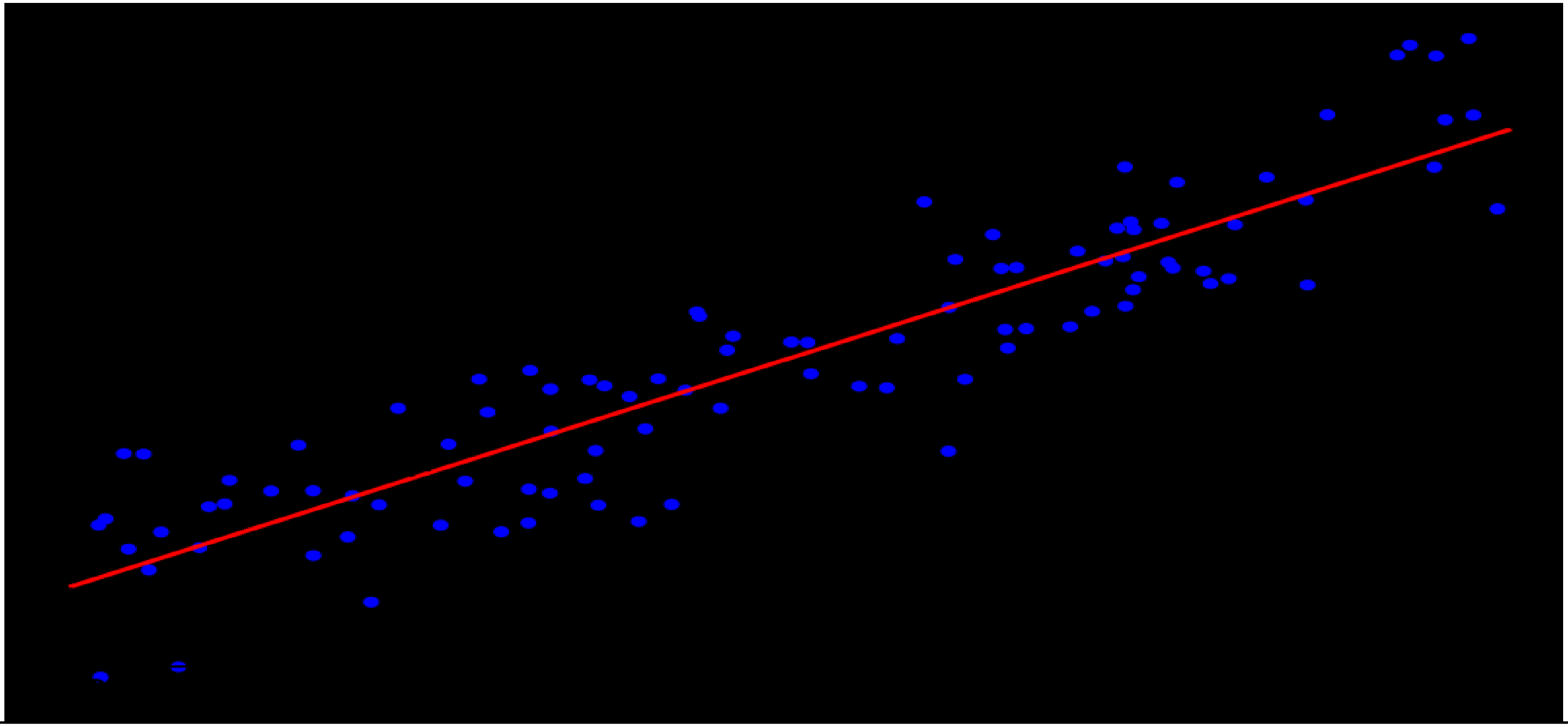
- L(yes) = 2/9 * 3/9 * 3/9 * 3/9 = 0.0082
- L(no) = 3/5 * 1/5 * 4/5 * 3/5 = 0.0577

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

- The decision would be: NO.

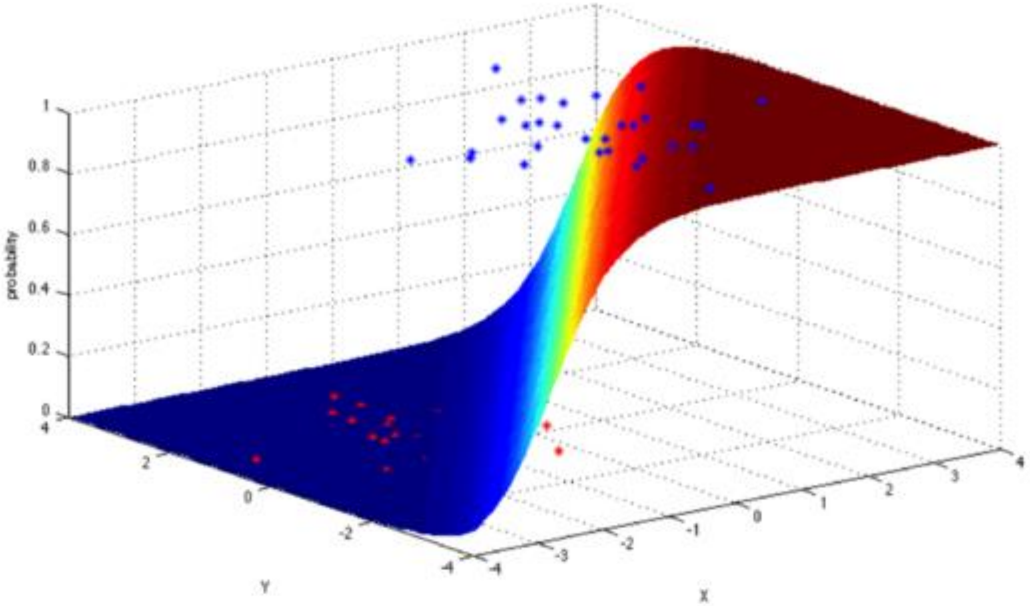
Linear Regression (supervised)

- **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)

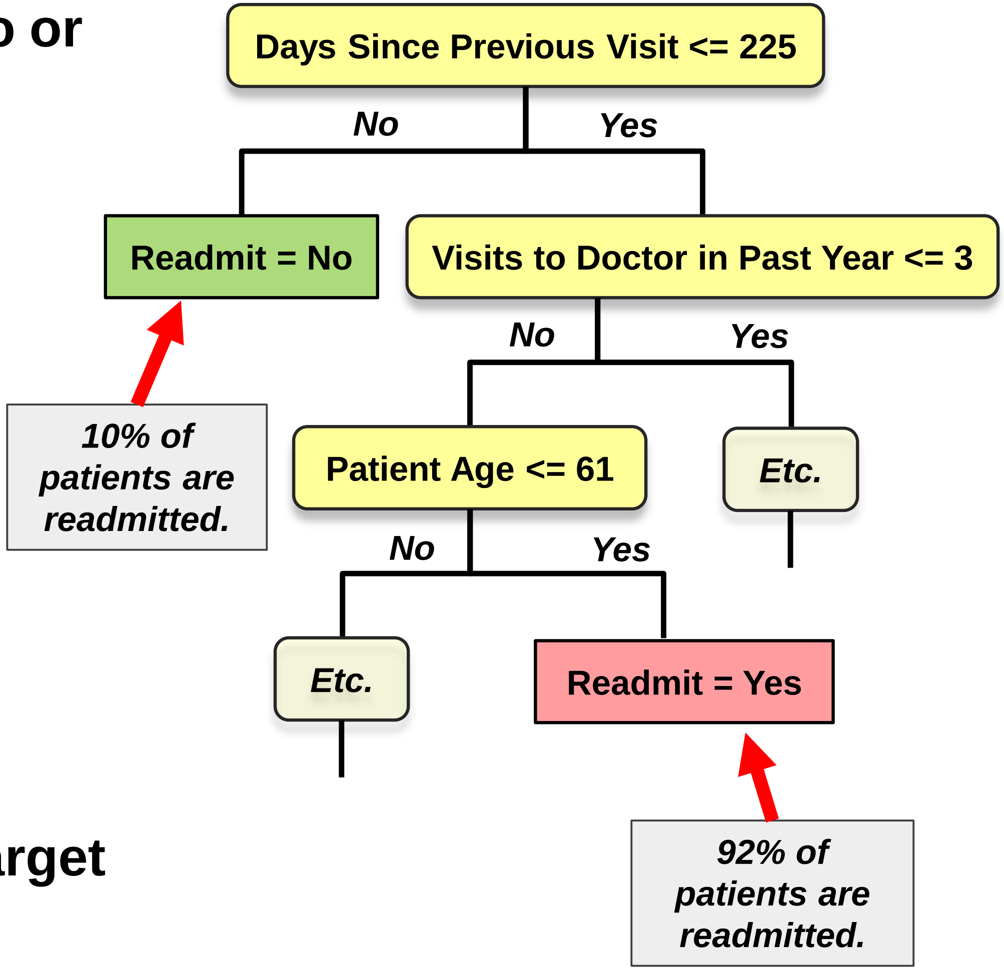
- **Logistic regression is a powerful statistical way of modeling a binomial outcome with one or more explanatory variables. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the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



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").**



Ensemble Modeling

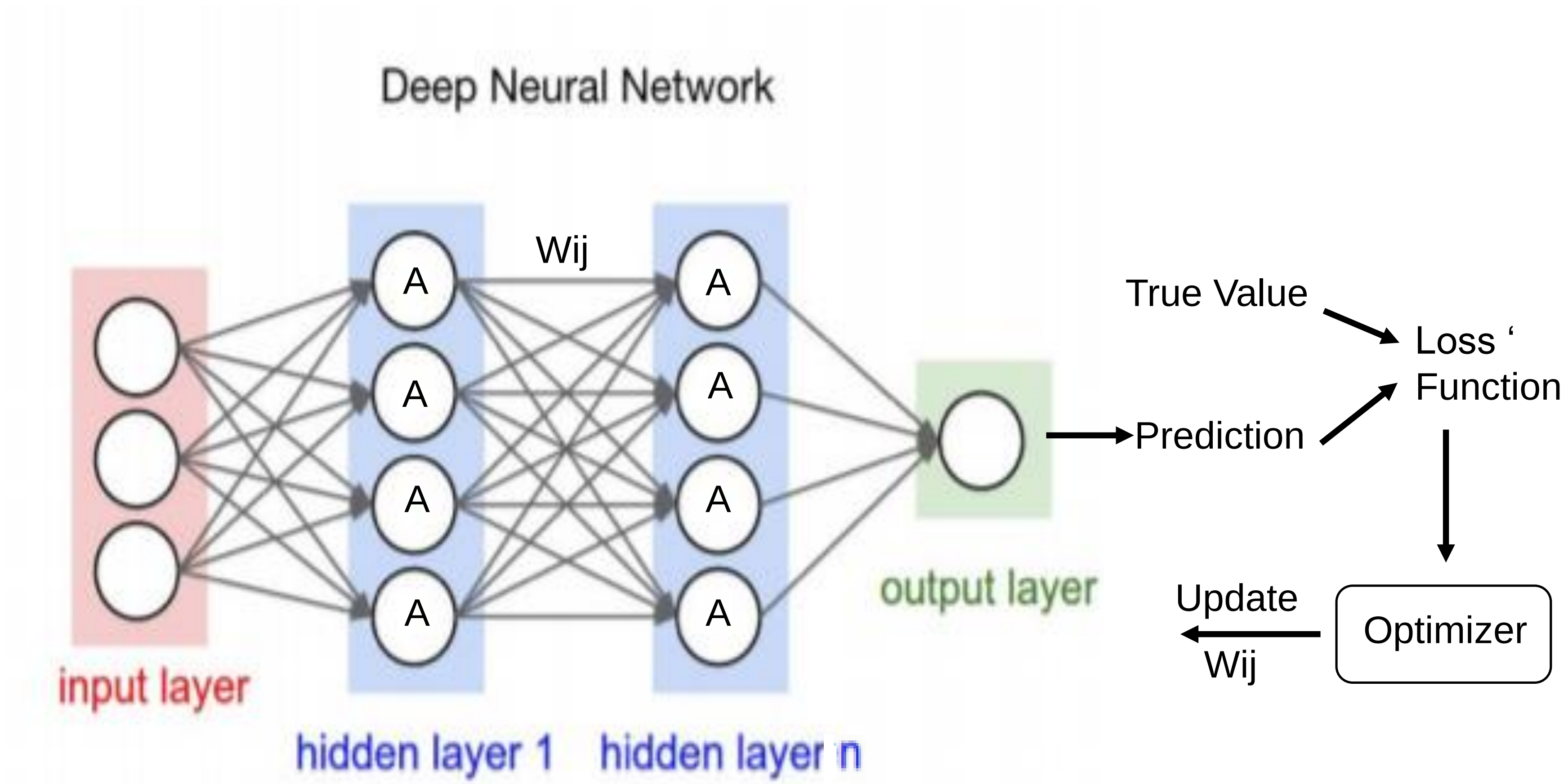
- **Use a collection or ensemble of models instead of a single model to create more reliable and accurate predictive models**

- **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

Modeling

- **Inspired by the way the human brain works.**



Wij - weights   A – Activation Function

Neural Network

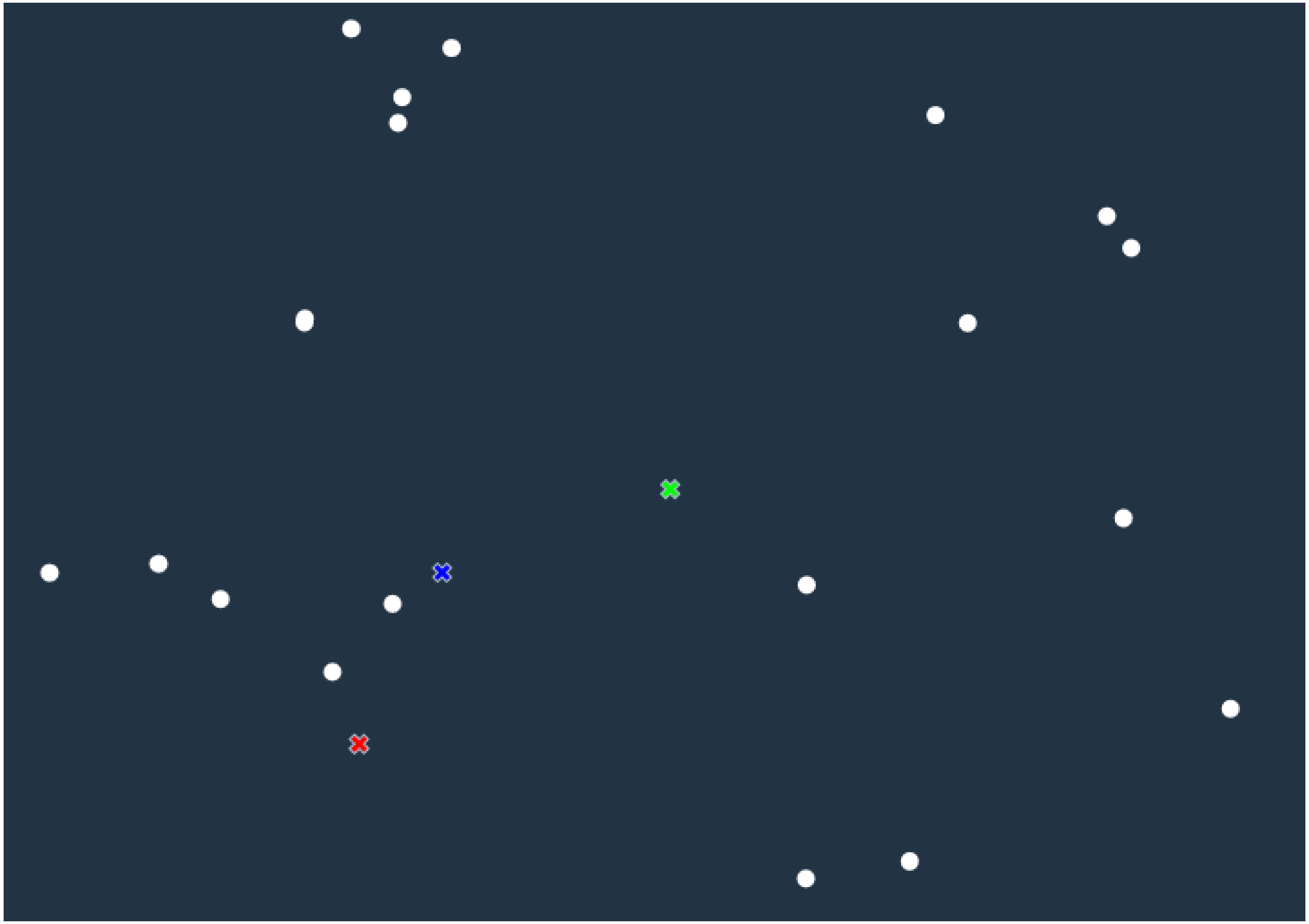
- **Originated in 1940s**

- **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
  - …

Clustering – K-means method

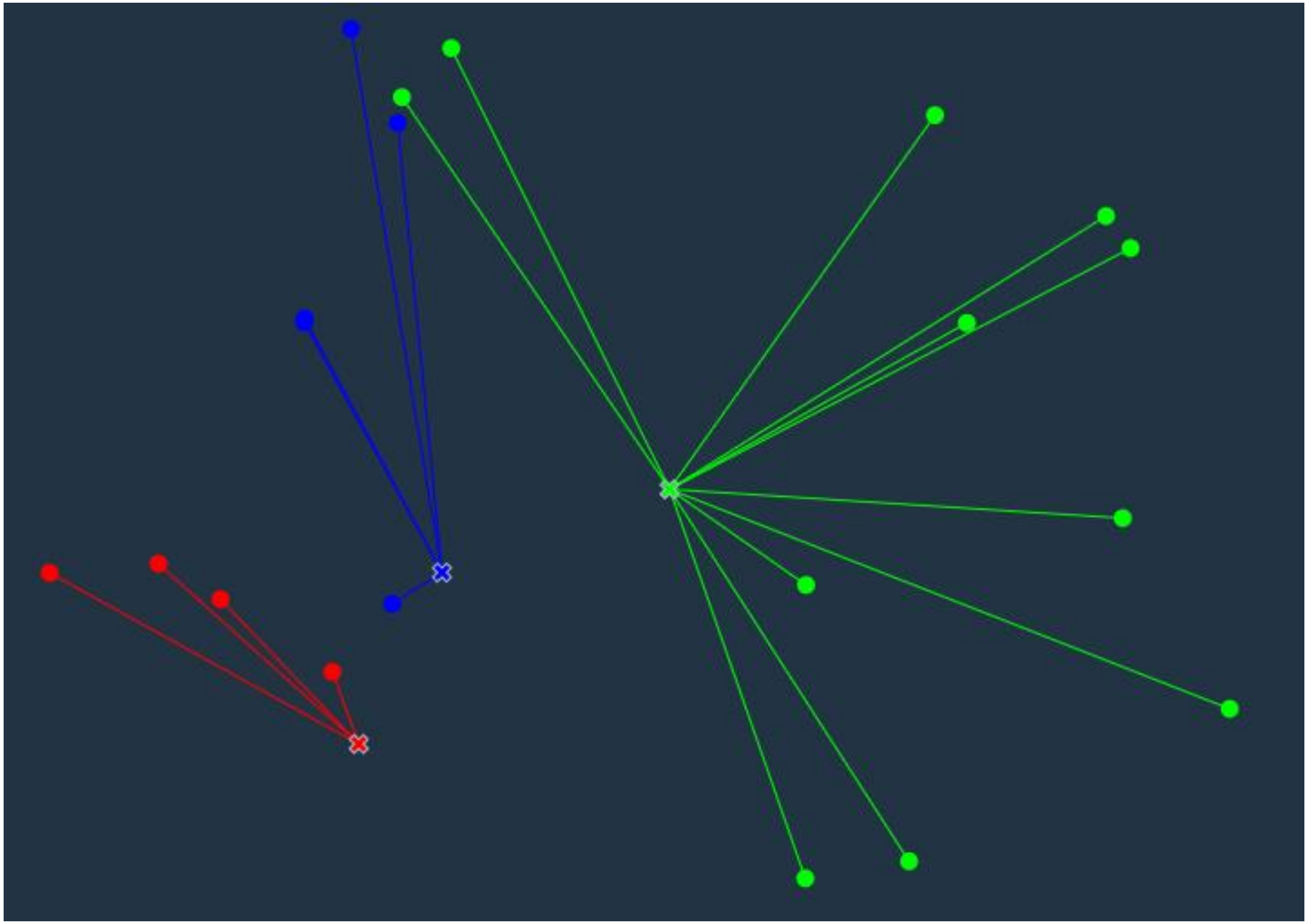
Modeling



Start with 20 data points and 3 clusters

Clustering – K-means method

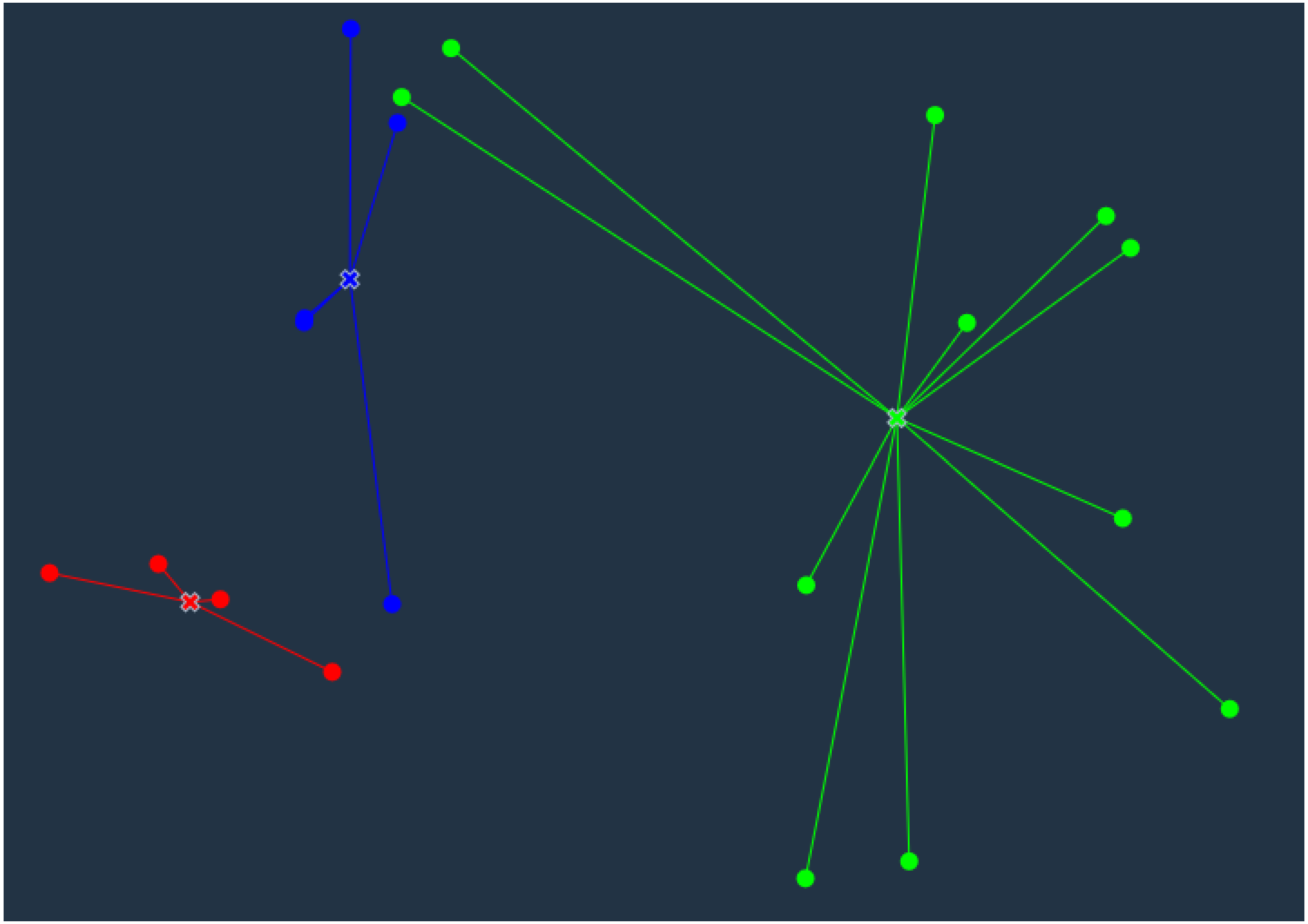
Modeling



Assign each data point to the nearest cluster

Clustering – K-means method

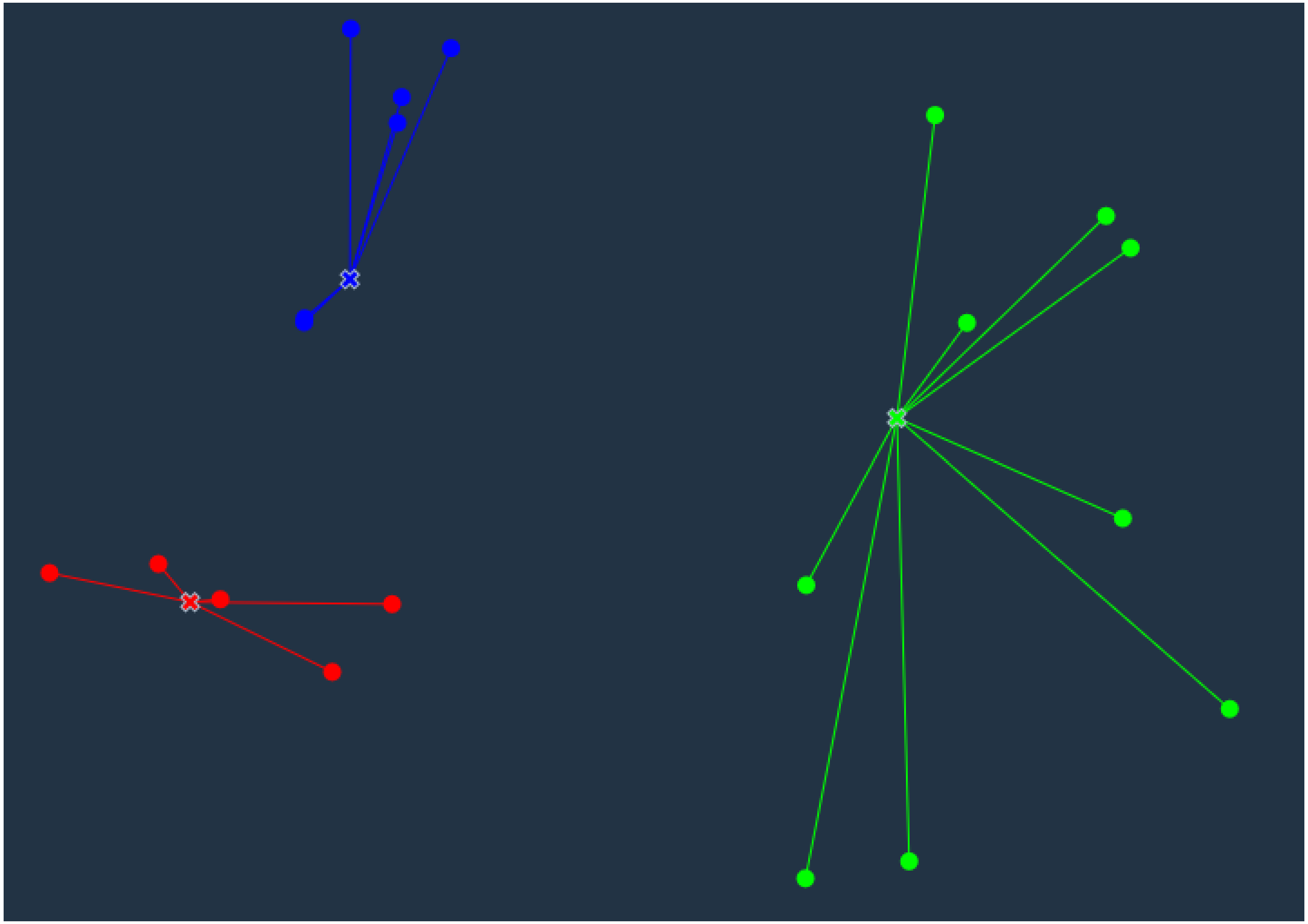
Modeling



Calculate centroids of new clusters

Clustering – K-means method

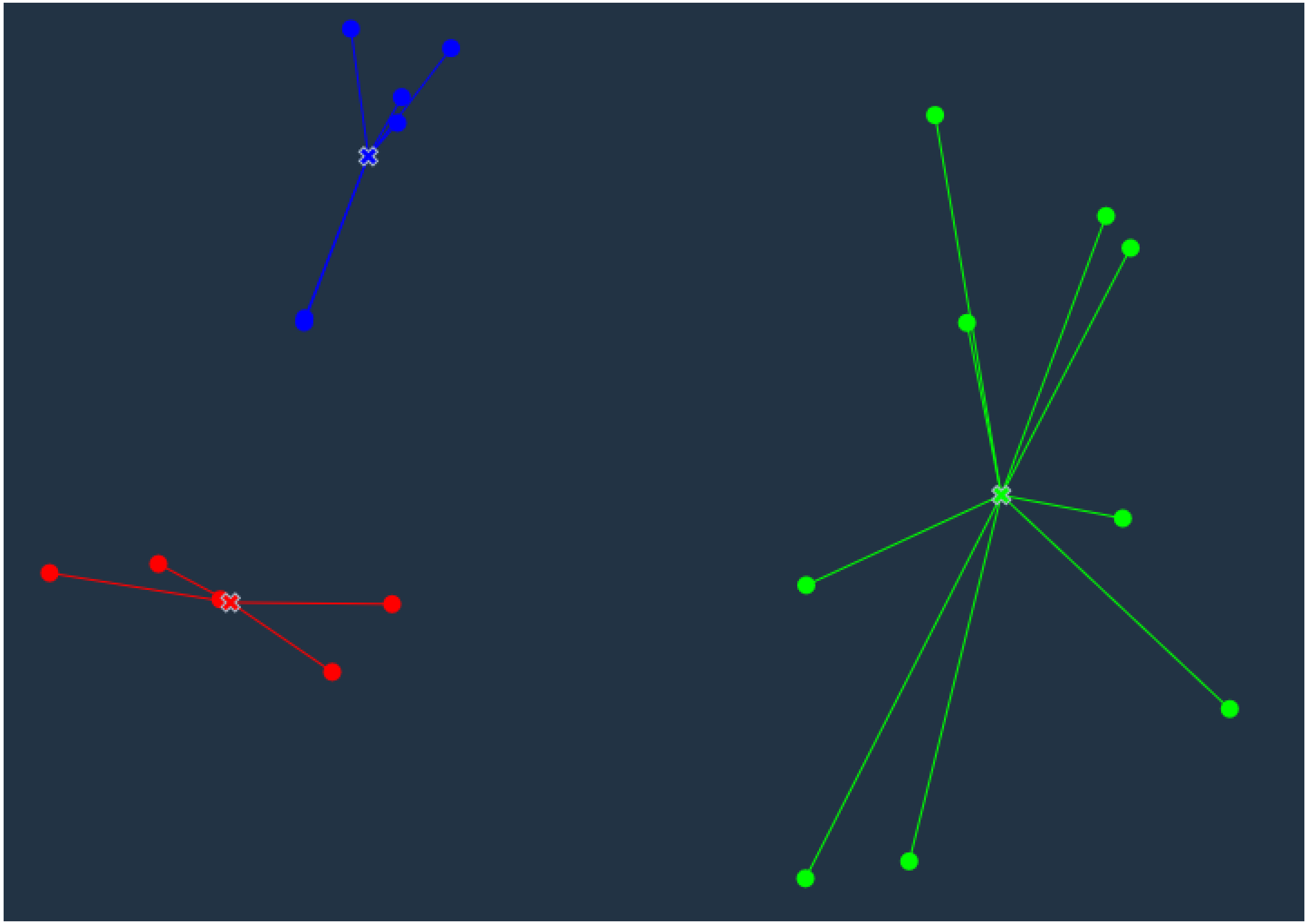
Modeling



Assign each data point to the nearest cluster

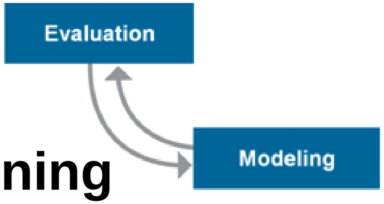
Clustering – K-means method

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	
		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) ✓

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.**

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.**

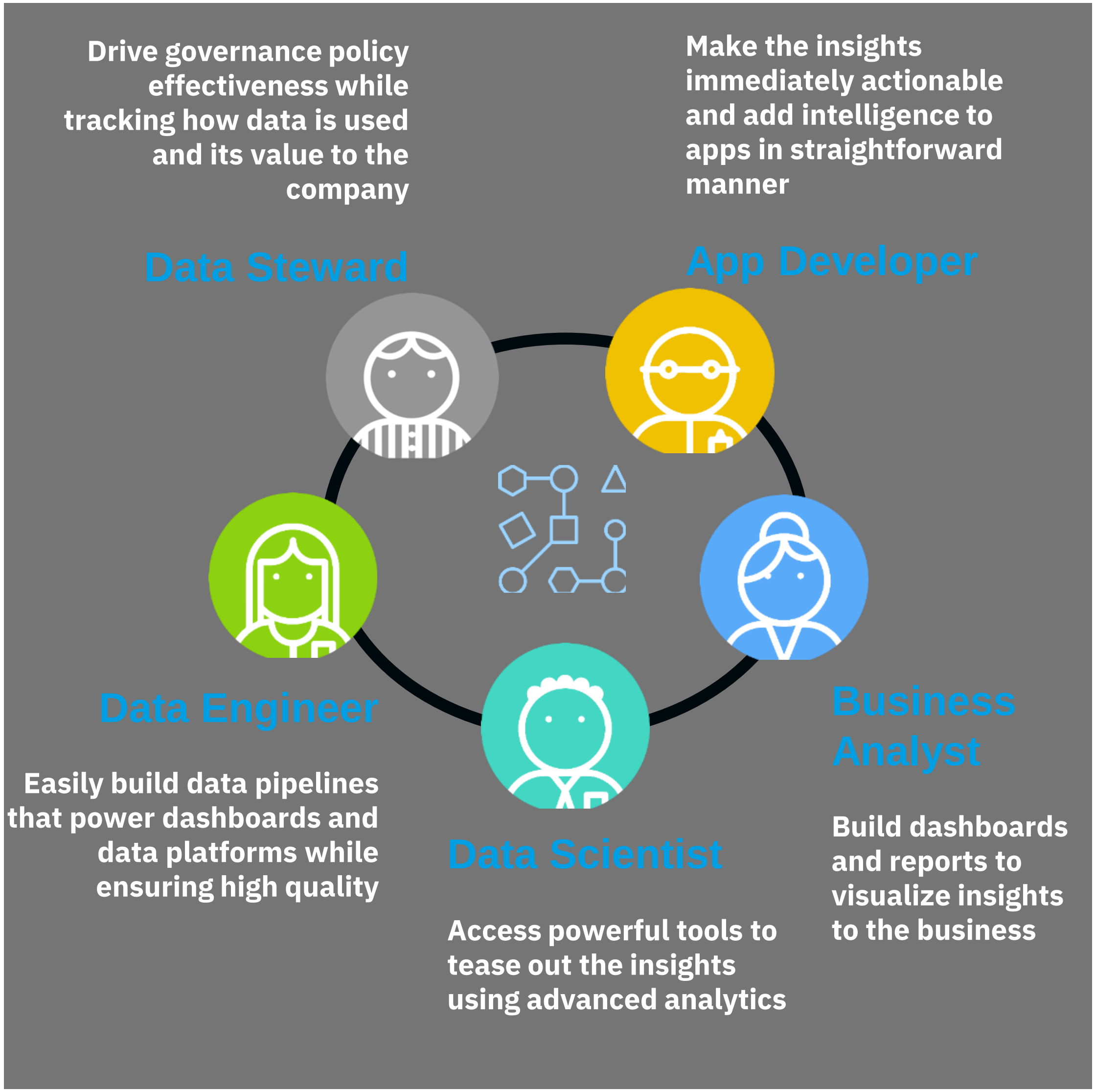
Watson Studio Platform

IBM Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality.
Information about potential future products may not be incorporated into any contract.
The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

IBM Watson Studio Platform

An integrated platform of tools, services, and data 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.

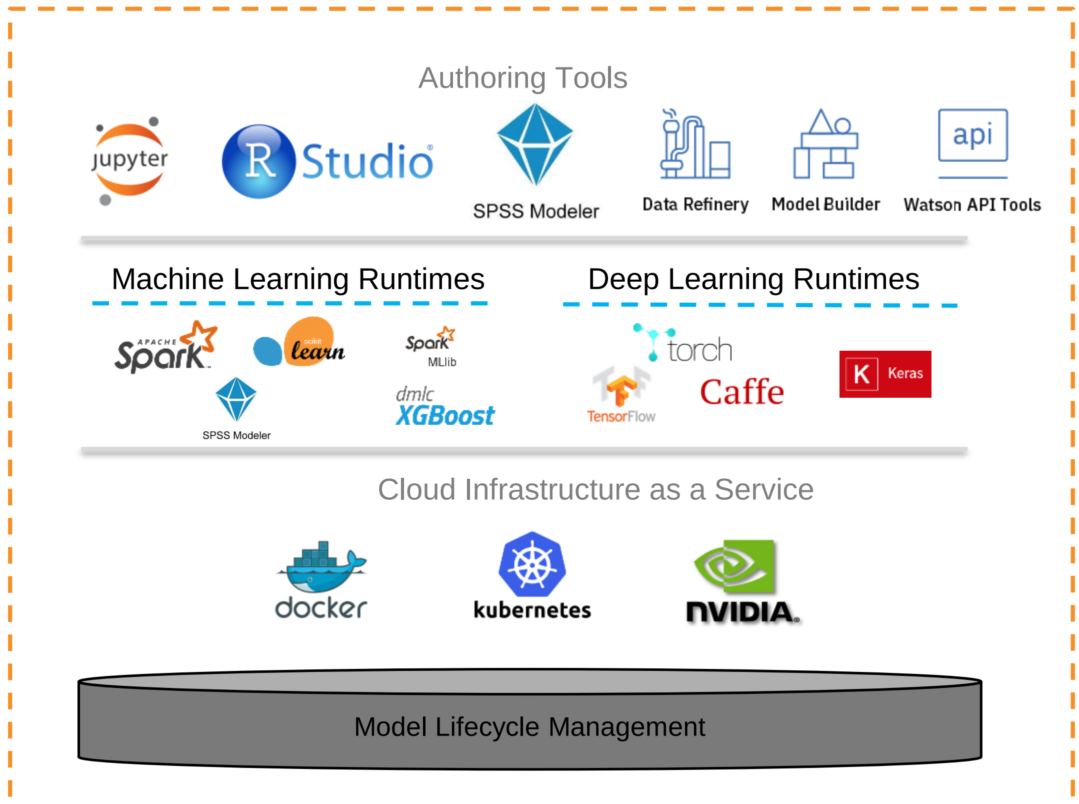
Connect & Access Data	Search and Find Relevant Data	Prepare Data for Analysis	Build and Train ML/DL Models	Deploy Models	Monitor, Analyze and Manage
Connect and discover content from multiple data sources in the cloud or on premises. Bring **structured** and **unstructured** data to one toolkit.	**Find** 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

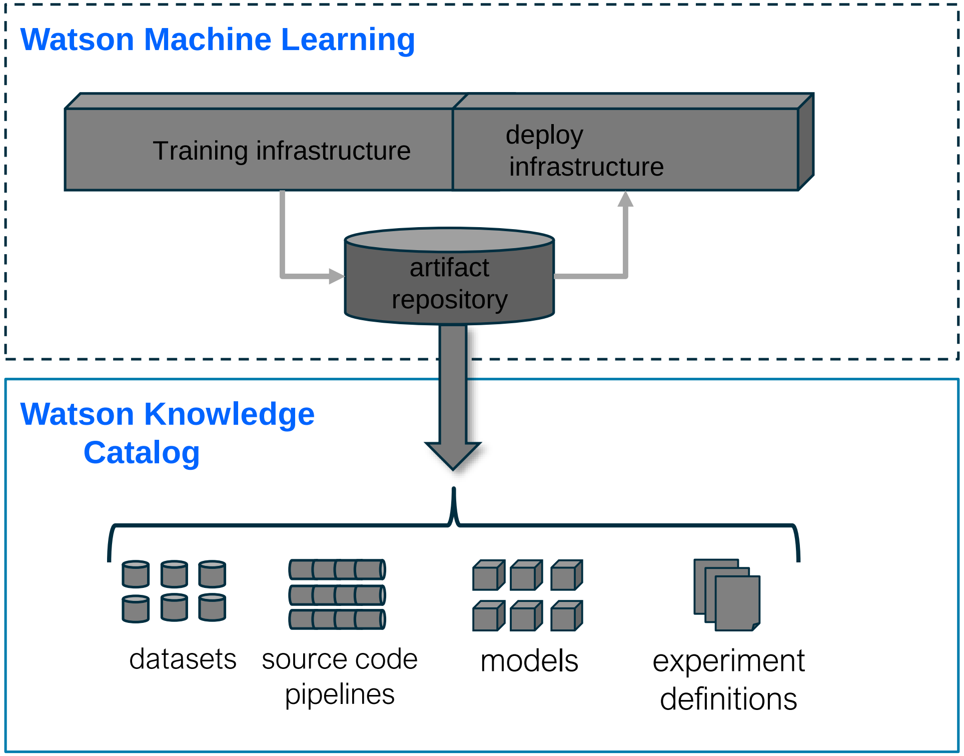
- Open Source and IBM libraries/frameworks

- Fully managed service
- Container-based resource management
- Elastic pay as you go cpu/gpu power



Watson Studio Model Lifecycle Management

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



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

Making Data Science a Team Sport

Making Data Science a **Team Sport**

Learn

Built-in learning to get started or go the distance with advanced tutorials

- Community with free sample Notebooks, Models, Open Data Sets to import and start working right away.
- Learn with hundreds of tutorials and get informed about the latest techniques with articles
- Share and bookmark your favorite community assets

Create

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

- Code in Python, R and Scala in Jupyter Notebooks or Rstudio.
- Access the most popular ML & DL frameworks.
- Create Models in minutes without coding with Visual Modeling tools.
- Start with Watson pre-trained models and customize them with your own data
- Create compute environments on demand and scale/customize them as needed.

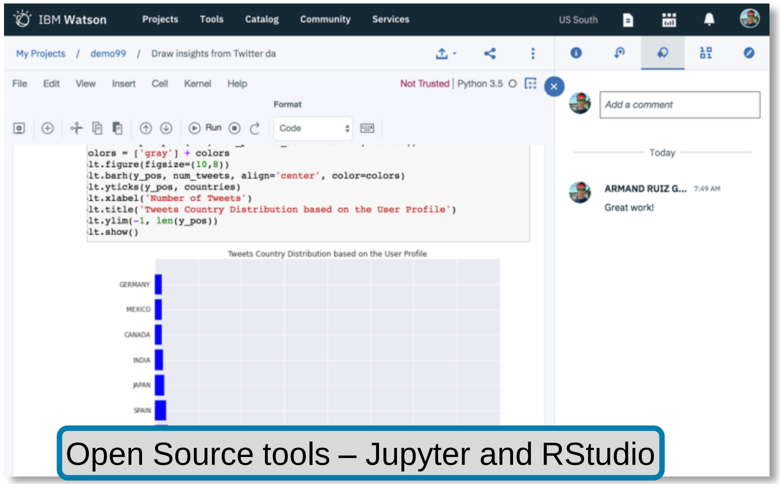
Collaborate

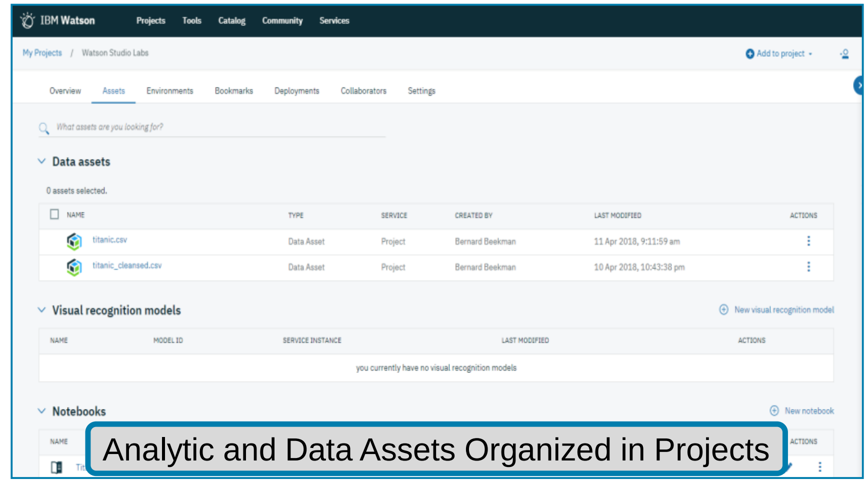
Projects organize resources and are the home base for collaboration

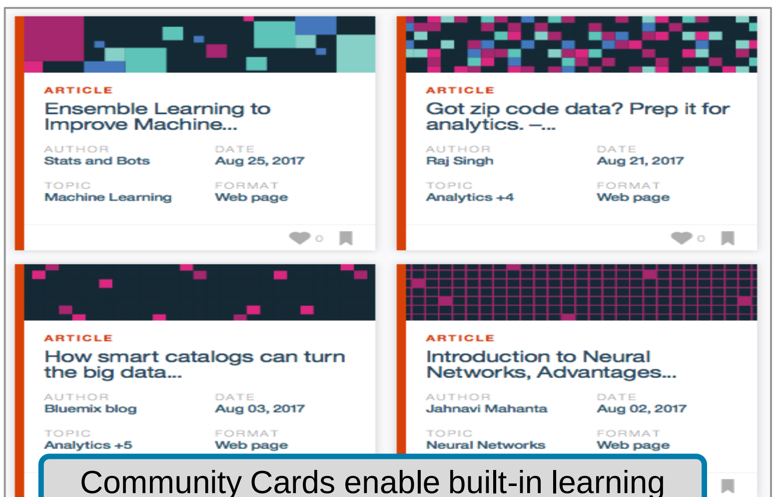
- Create Projects and add your colleagues as collaborators.
- Control the permissions with Admin/Editor/Viewer access
- Add comments and track the activity of your colleagues in Projects
- Version control of your assets
- Data Scientists, Data Engineers, SMEs and Developers all in one environment to accelerate innovation and collaboration

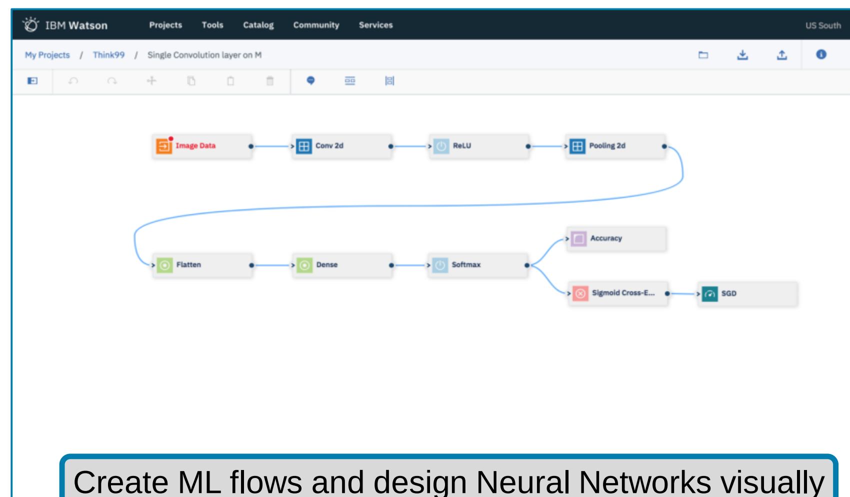
Watson Studio

Making Data Science a Team Sport

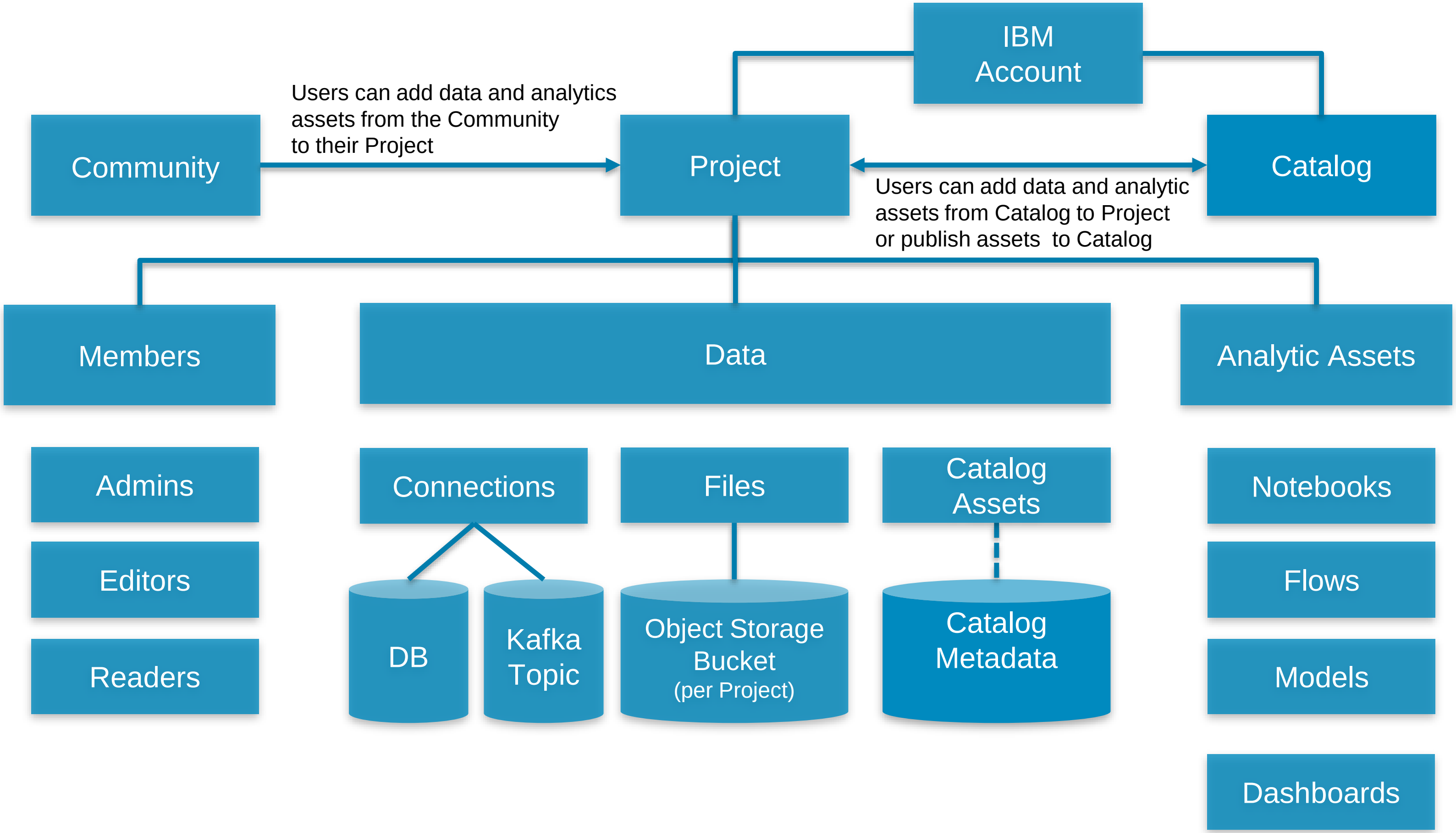






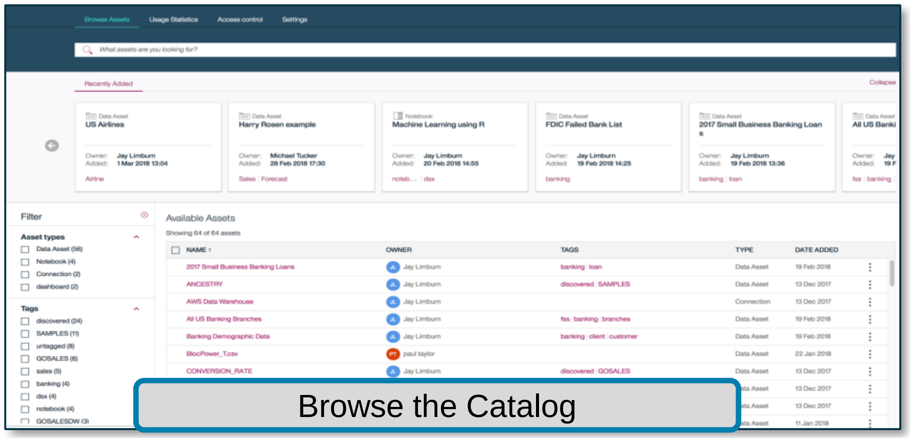


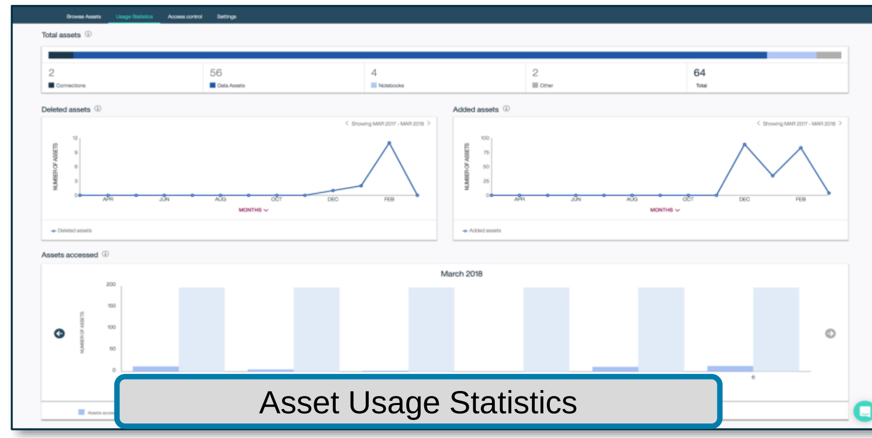
Projects allow users to work and collaborate

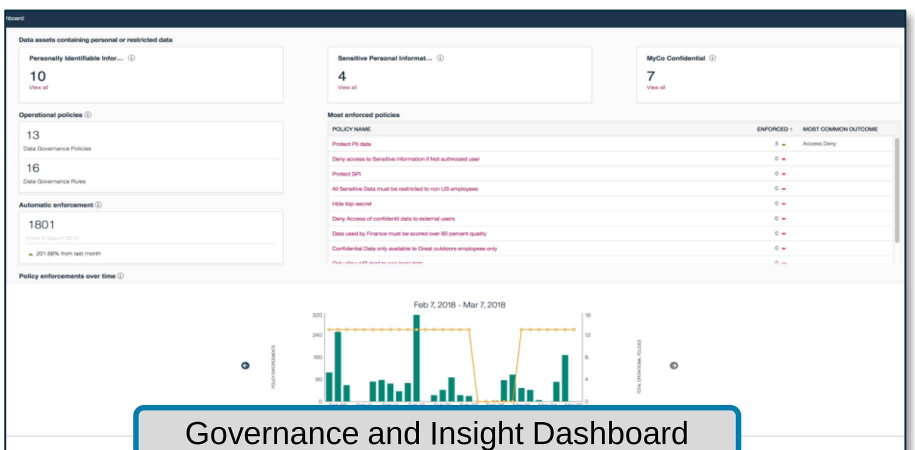


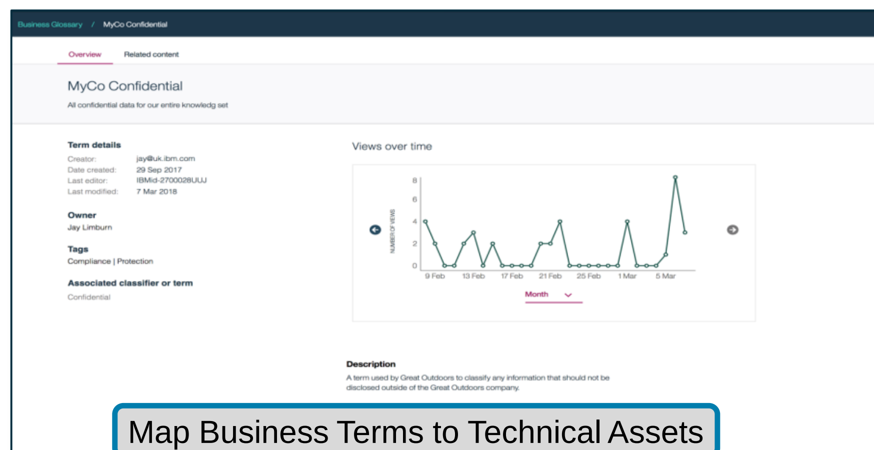
Watson Knowledge Catalog

Unlock tribal knowledge and unleash knowledge workers



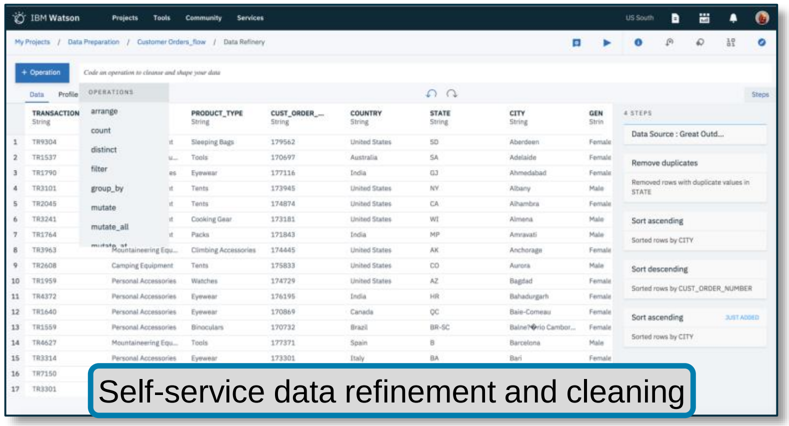


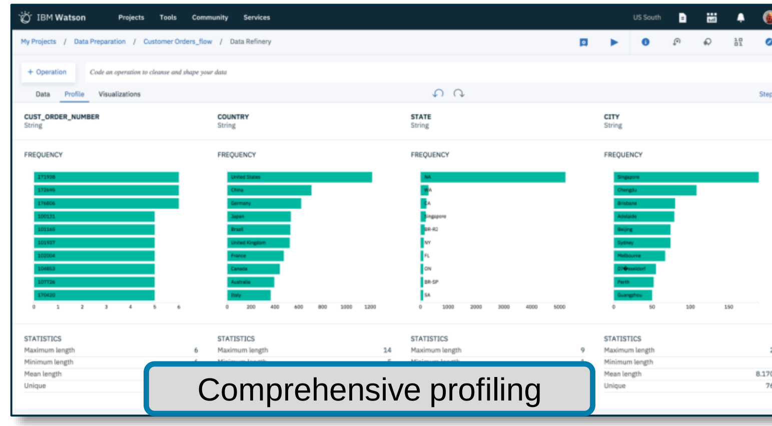


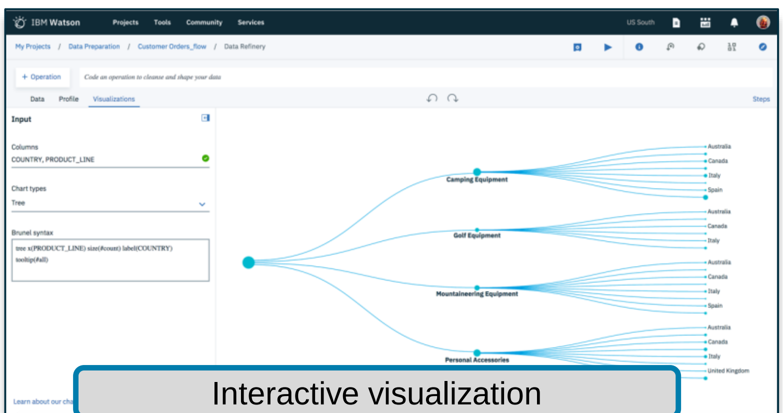


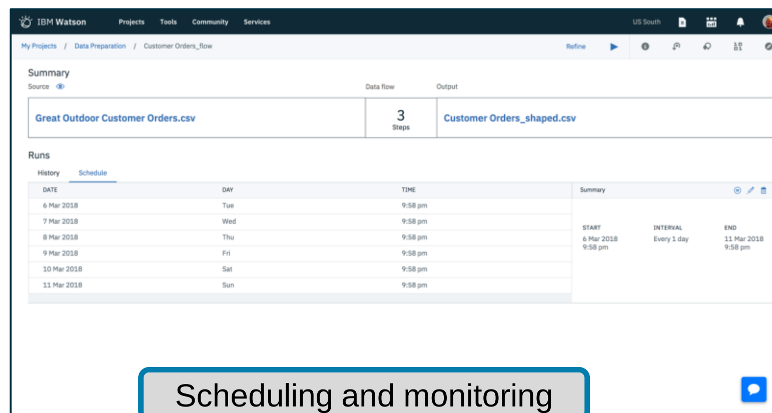
Data Refinery

Making Data fit for use



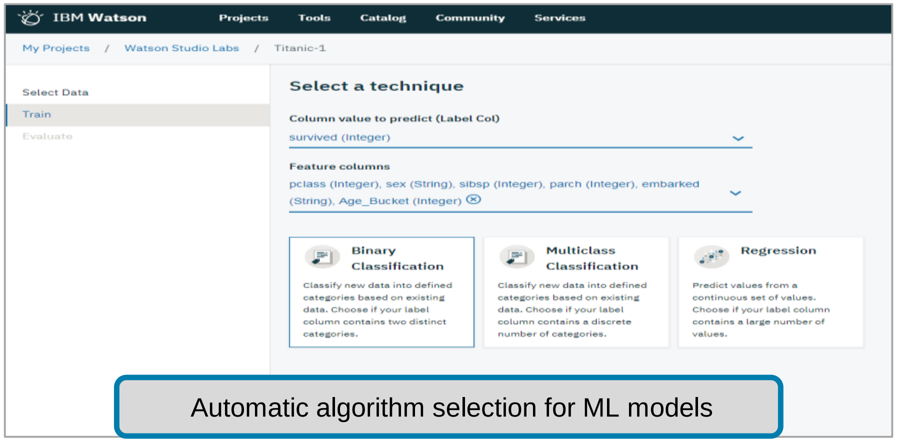




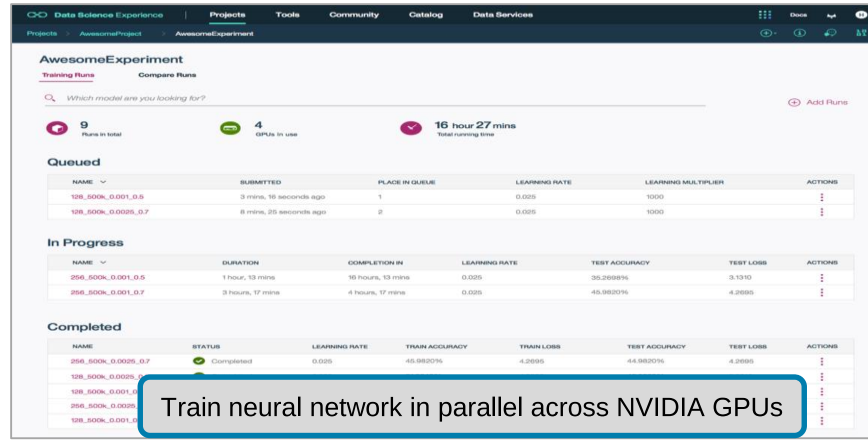


Watson Machine Learning

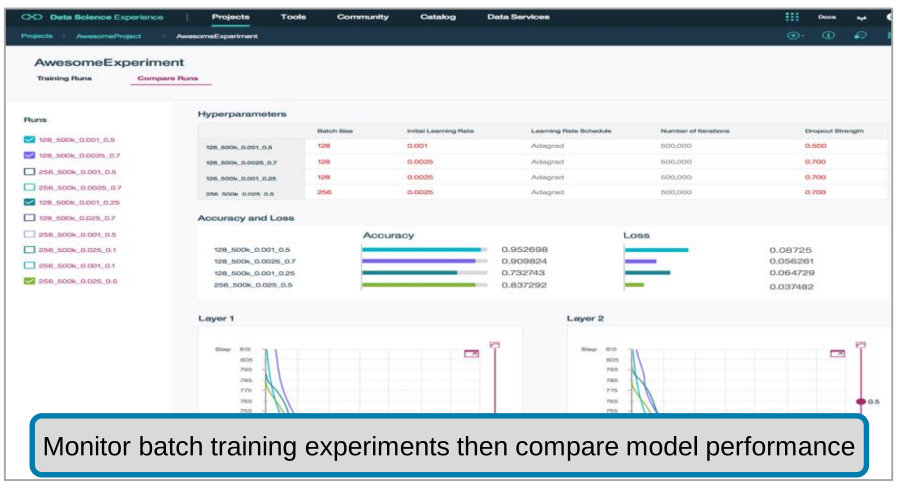
Simplifying deployment and management of ML models in production



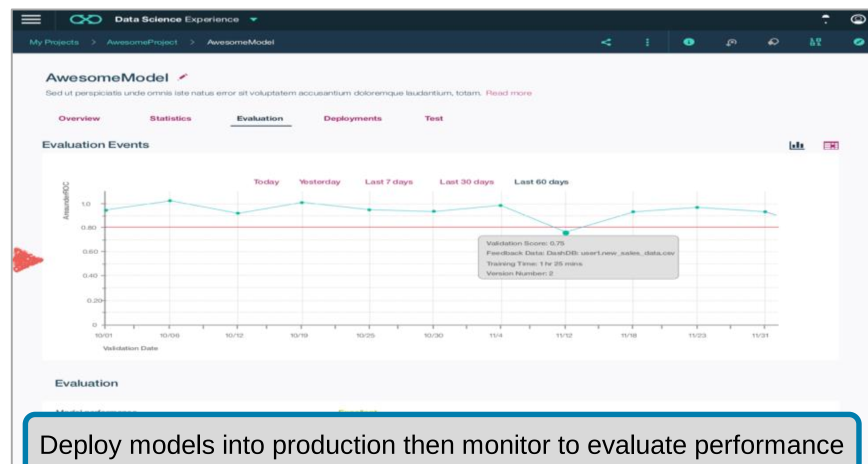
Automatic algorithm selection for ML models



Train neural network in parallel across NVIDIA GPUs



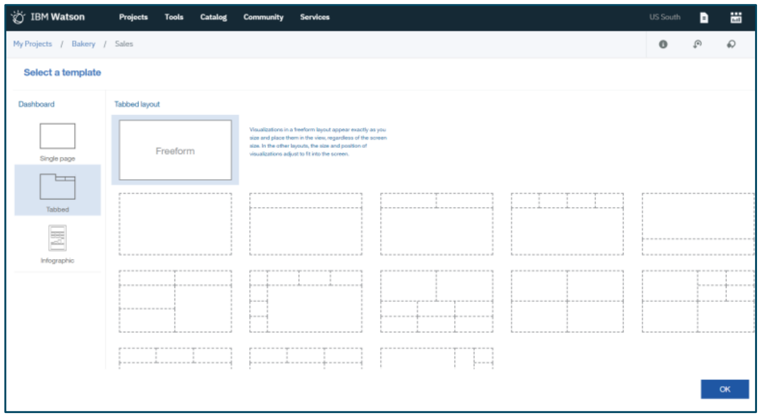
Monitor batch training experiments then compare model performance

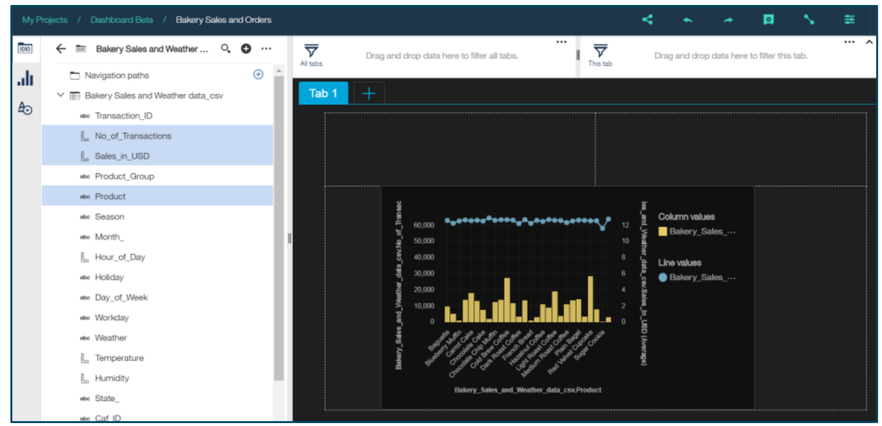


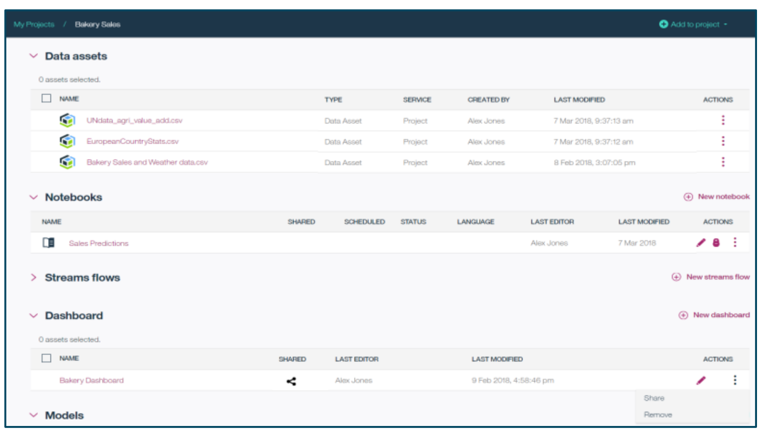
Deploy models into production then monitor to evaluate performance

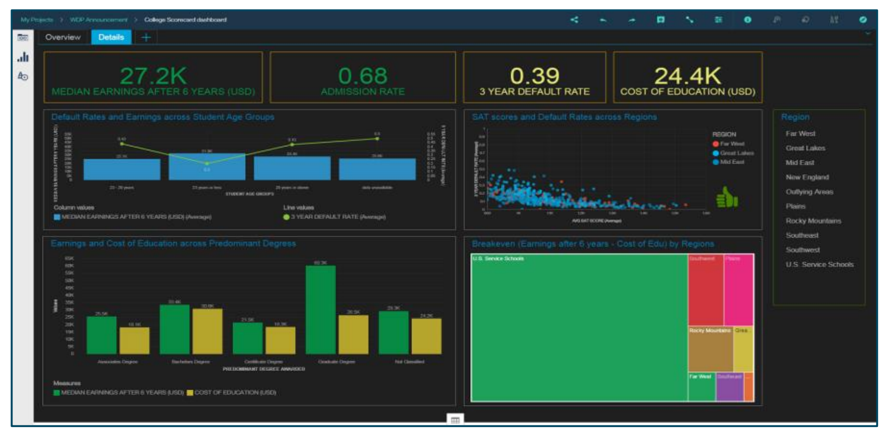
Dynamic Dashboards

Making insights available to all









3

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: For the 4th lab, select one or more of 3 optional labs that demonstrate (a) Watson Machine Learning deployment of a Machine Learning model, and DevOps to build an application that invokes the deployed model, (b) Visual Drag and Drop creation of a machine learning model pipeline, or (c) Spark Machine Learning via Jupyter Notebooks.

Lab Tips

- Labs are all located in www.github.com/bleonardb3/ML-POT repository
- 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 in the instructions for Lab-2, Lab-4a, and Lab-4b. 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 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

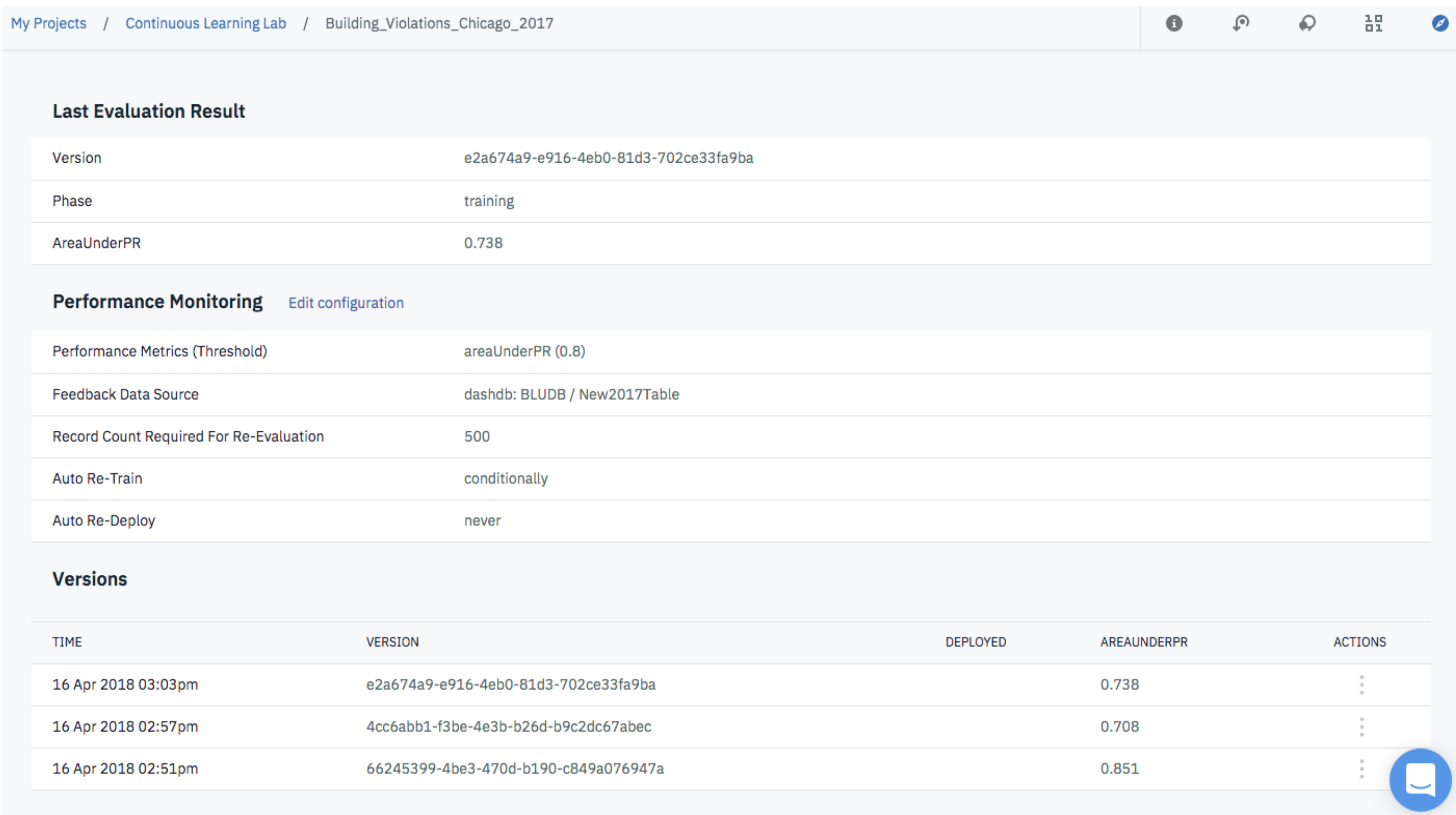
Using 2017 Chicago building data, we will make Chicago a safer place by building a model to predict when buildings are likely to fail inspection. We can then use our model to find which buildings are most dangerous and attend to those first.

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 Image Classification using a Neural Network

This lab features the new Watson Studio Neural Network modeler, and Experiment Assistant to build, train, and test a Convolutional Neural Network to classify images. The dataset used is the CIFAR-10 dataset (Canadian Institute For Advanced Research), a collection of images that are commonly used to train machine learning and computer vision algorithms.

In this lab we will:

- Create the neural network design (from example)
- Load the input data into object storage
- Configure the input data in the model
- Run an Experiment on the Training Definition
- Save and deploy the Model
- Test the Model

Lab-4 Predict Passenger Survival on the Titanic

For Lab-4, there are 3 alternative labs you can select, or select multiple if you have time. Each lab is based on the Titanic data set, often used in Kaggle competitions.

- Lab-4a - This lab will use the Watson Machine Learning 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 - This 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 - This lab will leverage Spark machine learning (SparkML) in a Jupyter notebook to predict survivability using pyspark and a supervised learning model.