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

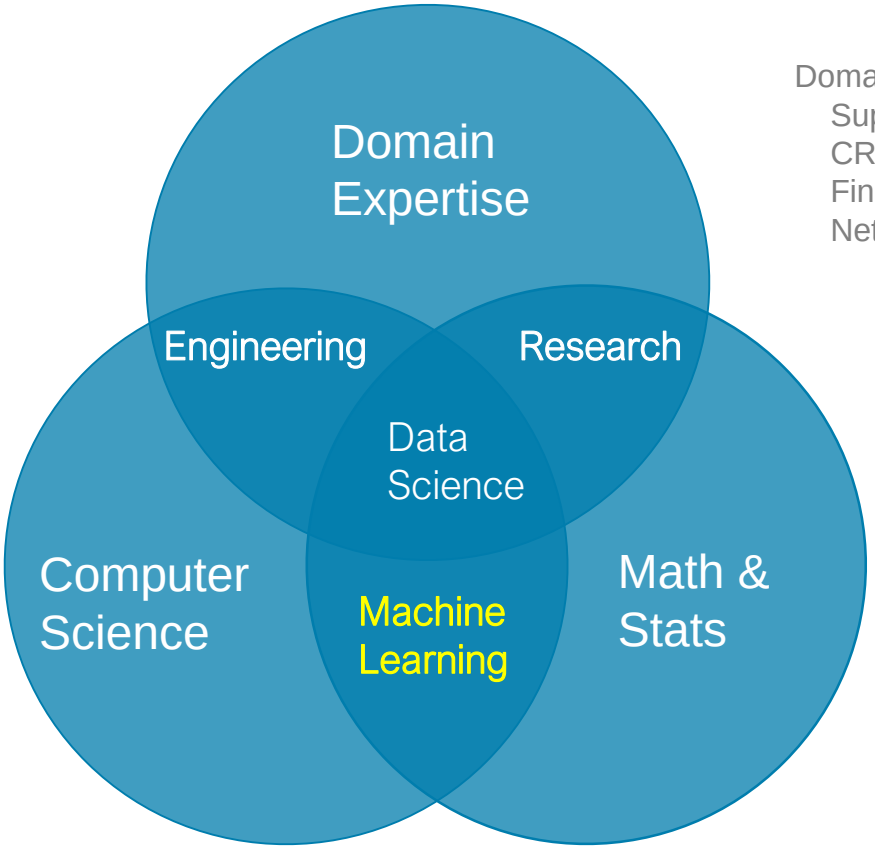
Presentation Outline

- **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….



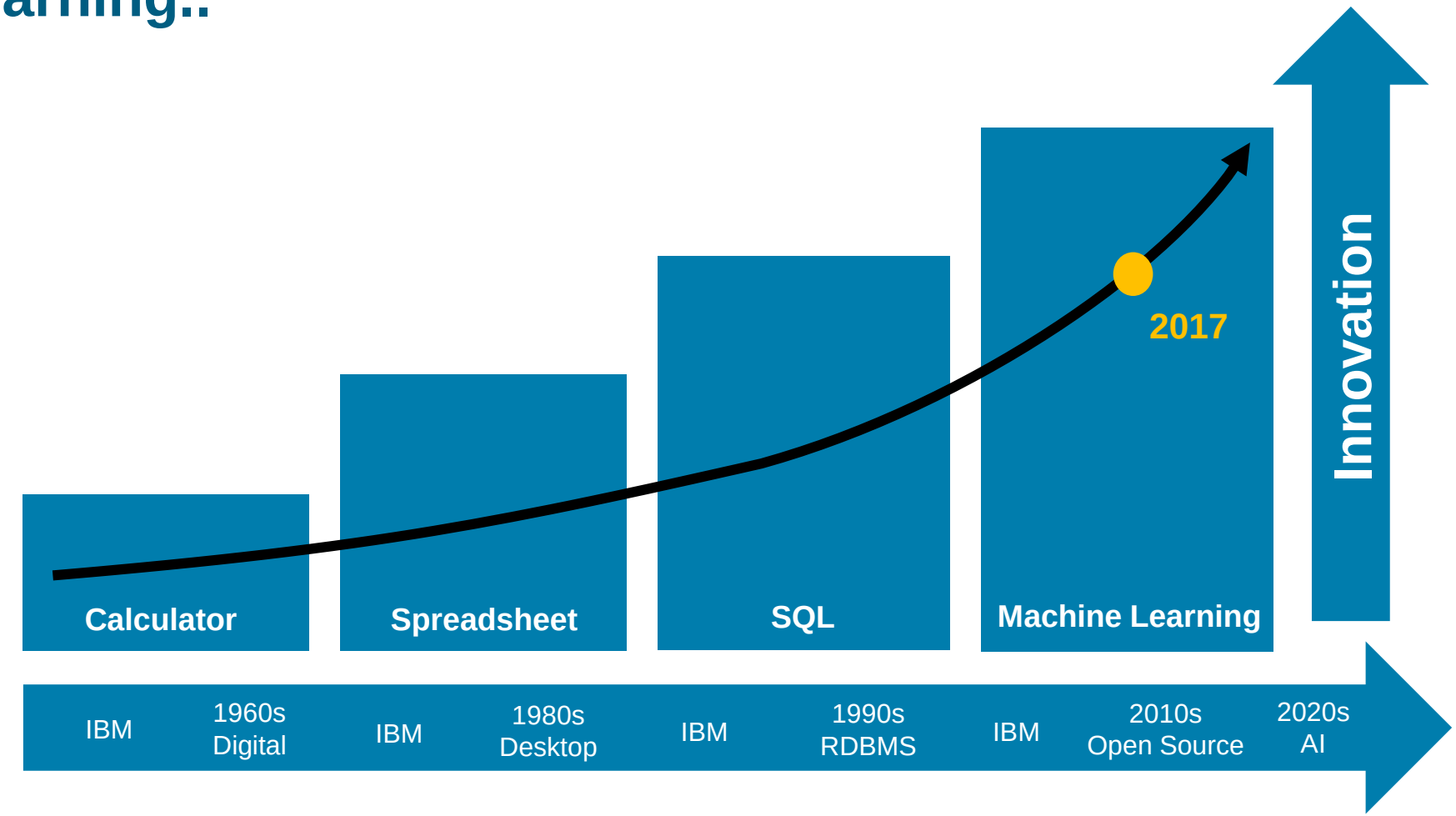
Domain Knowledge
- Supply Chain
- CRM
- Financials
- Networking

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

Mathematics
Computational

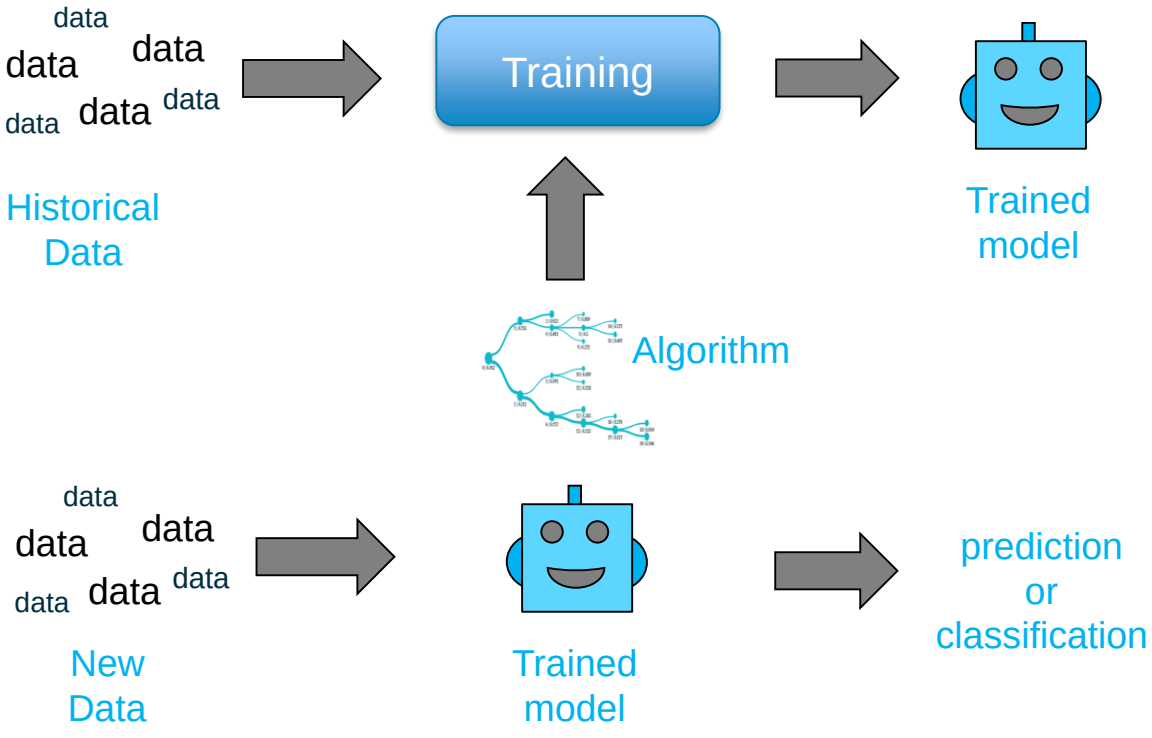
Data Science Projects Require Multiple Skills

Future of Data Science is Democratizing Machine Learning..



But what is Machine Learning?

"Computers that learn without being explicitly programmed"

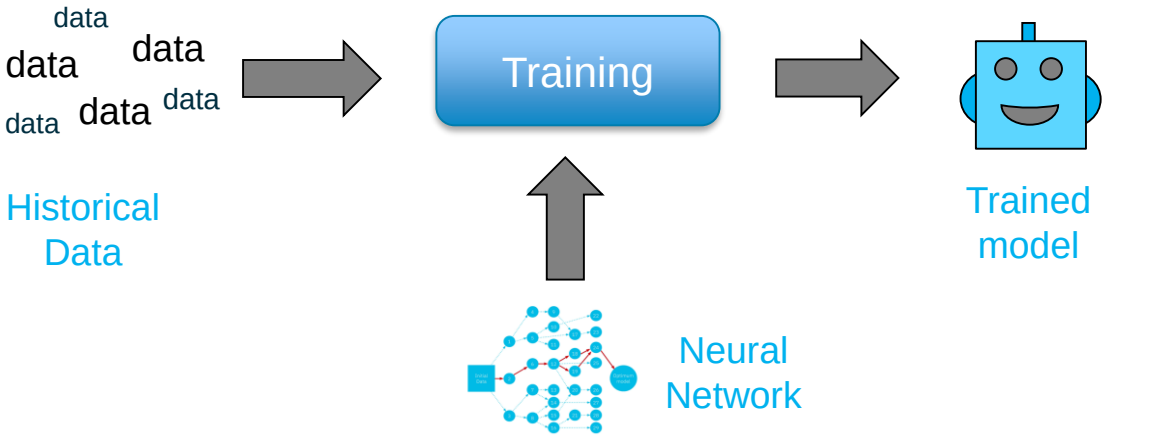


1. A machine learning model is trained to recognize patterns in historical data

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"

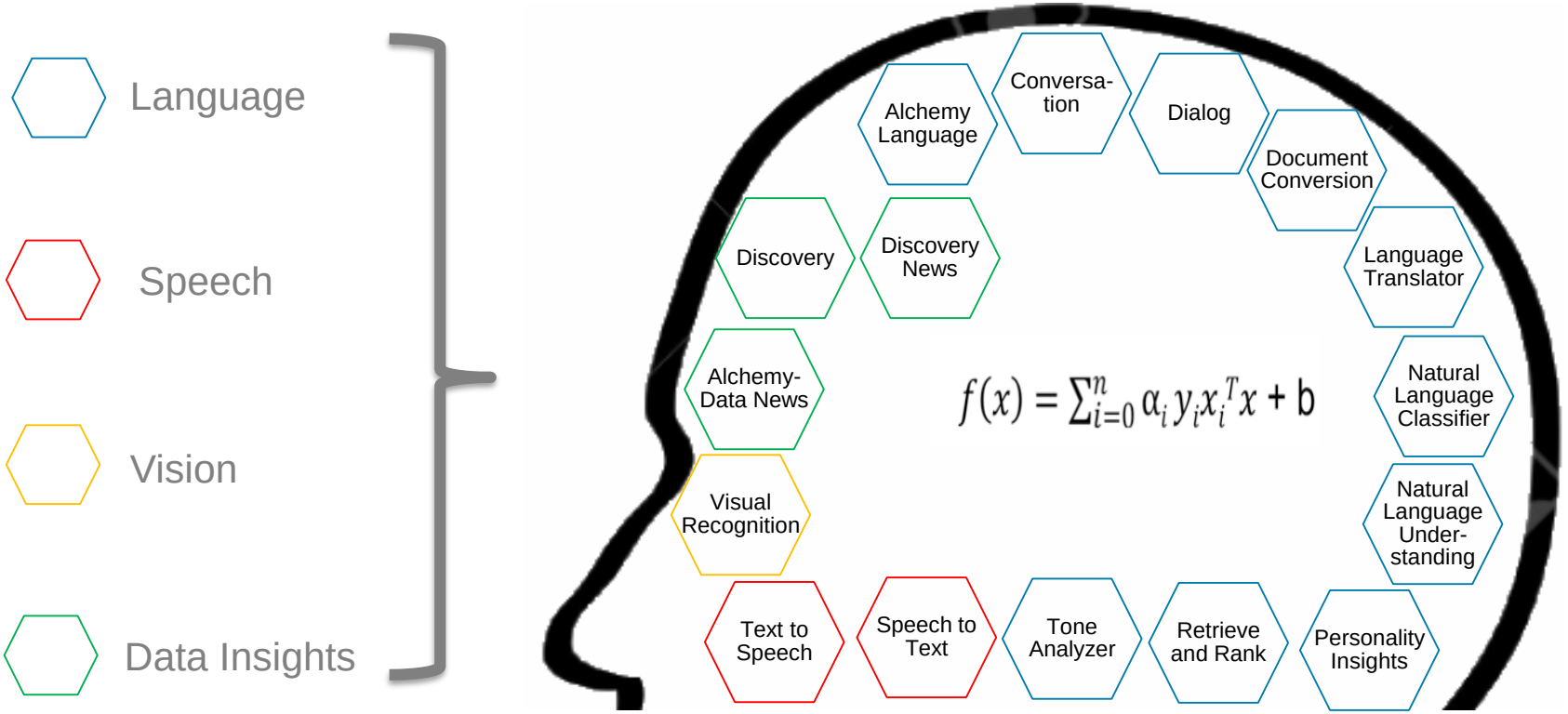


But what is Artificial Intelligence?

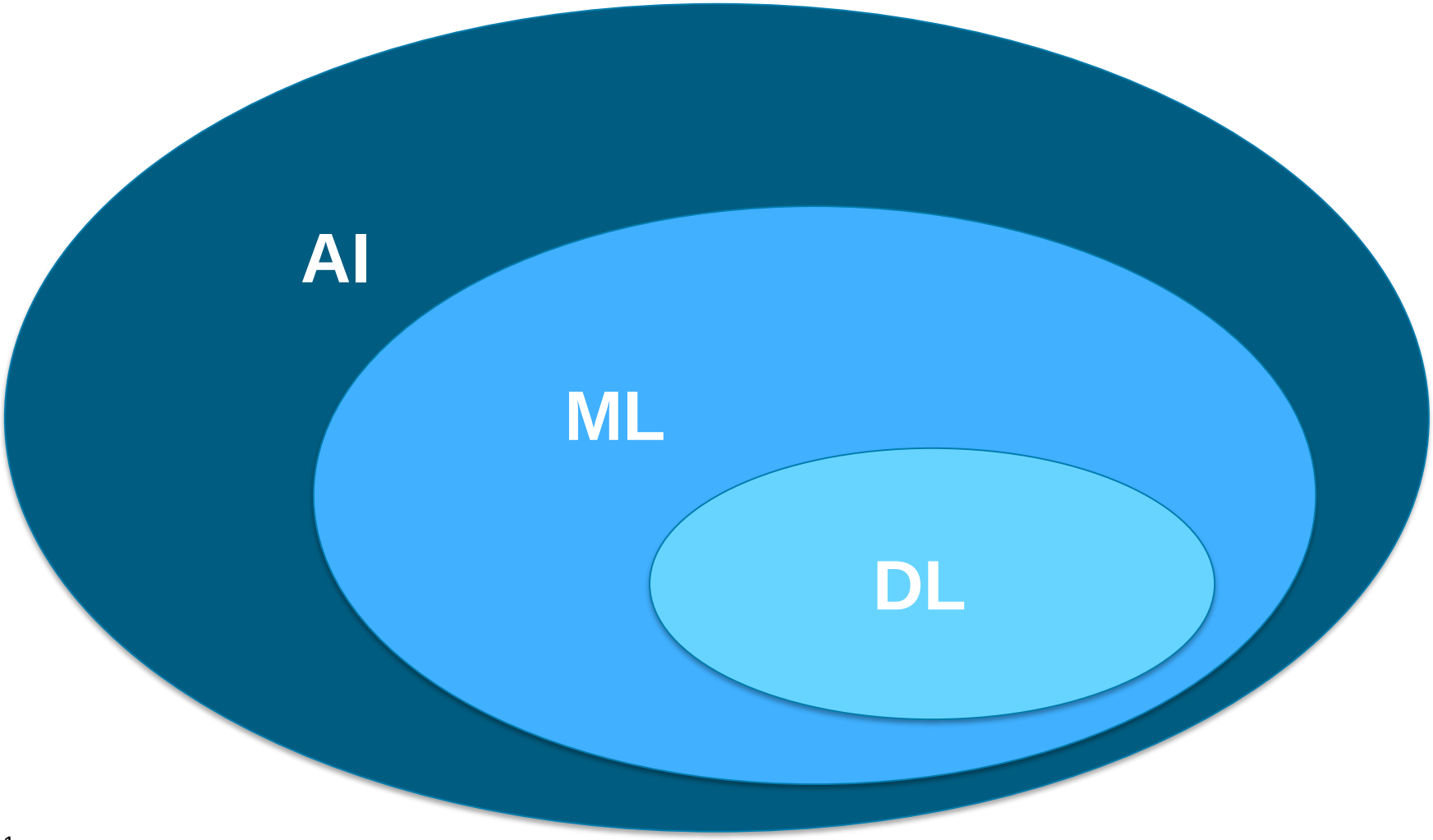
A theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages..

Machine Learning = Artificial Intelligence???

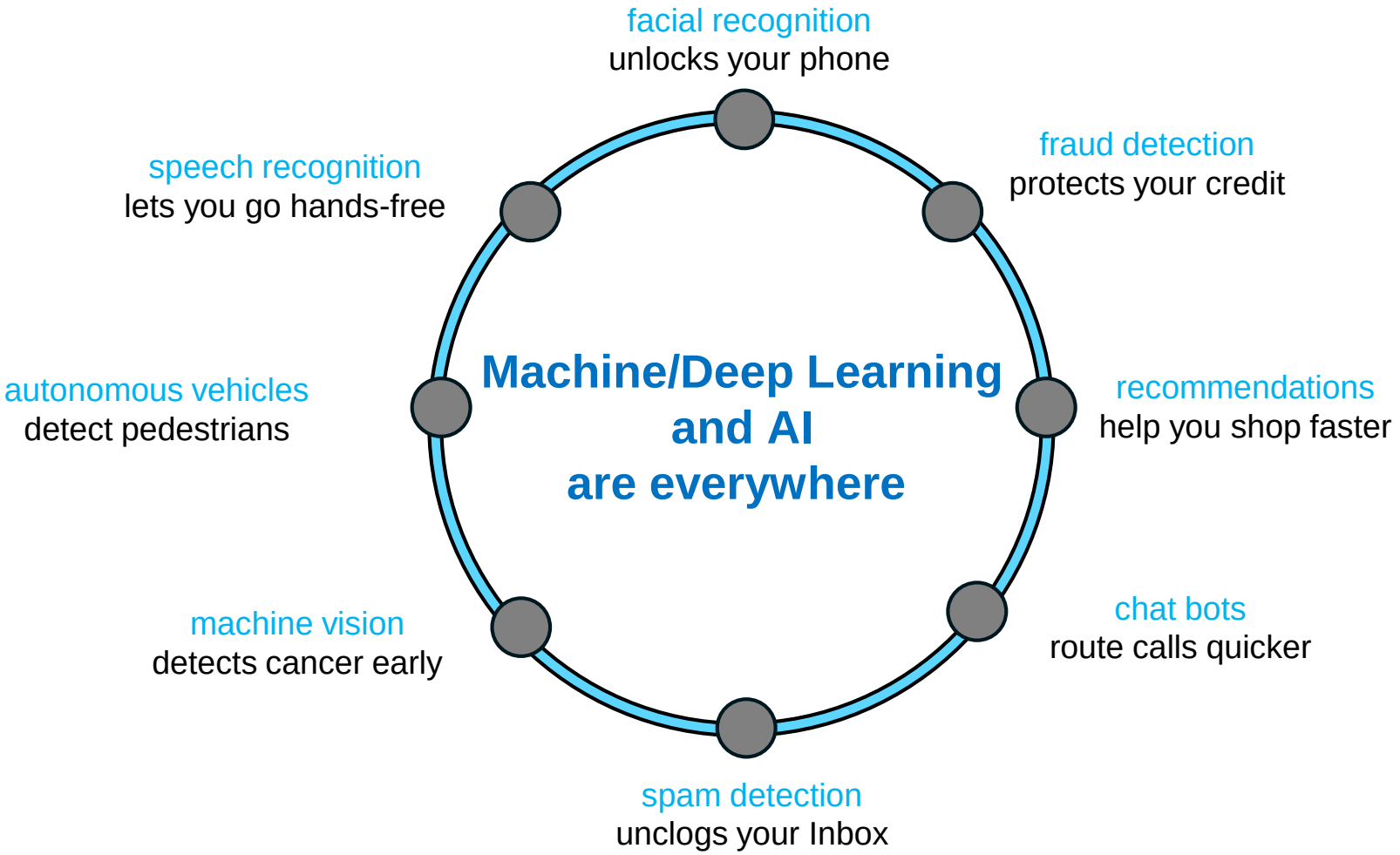
Data + Algorithms = Scored AI Models



Understanding AI, ML & DL Relationship…



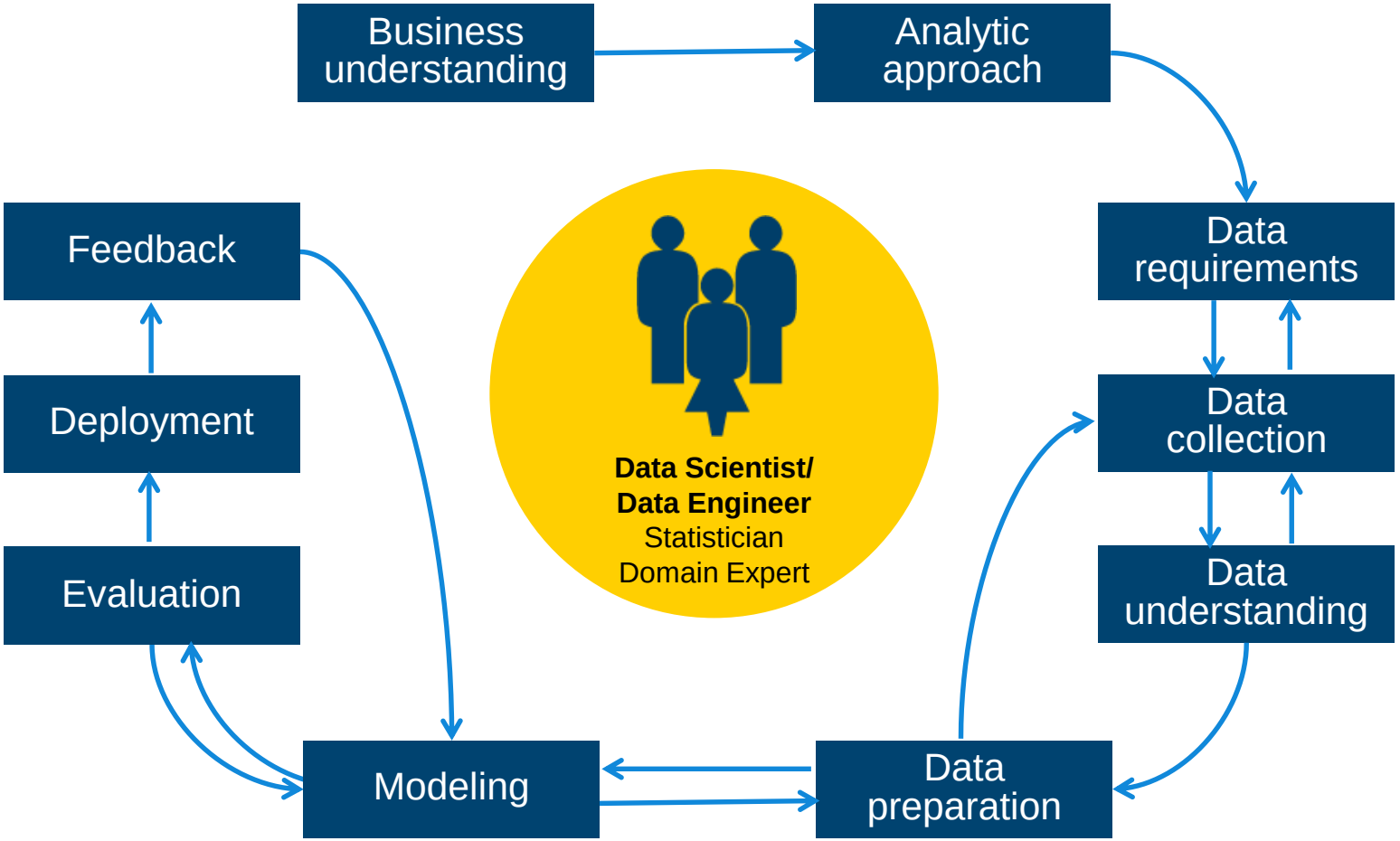
The future is now



Introduction to Machine Learning

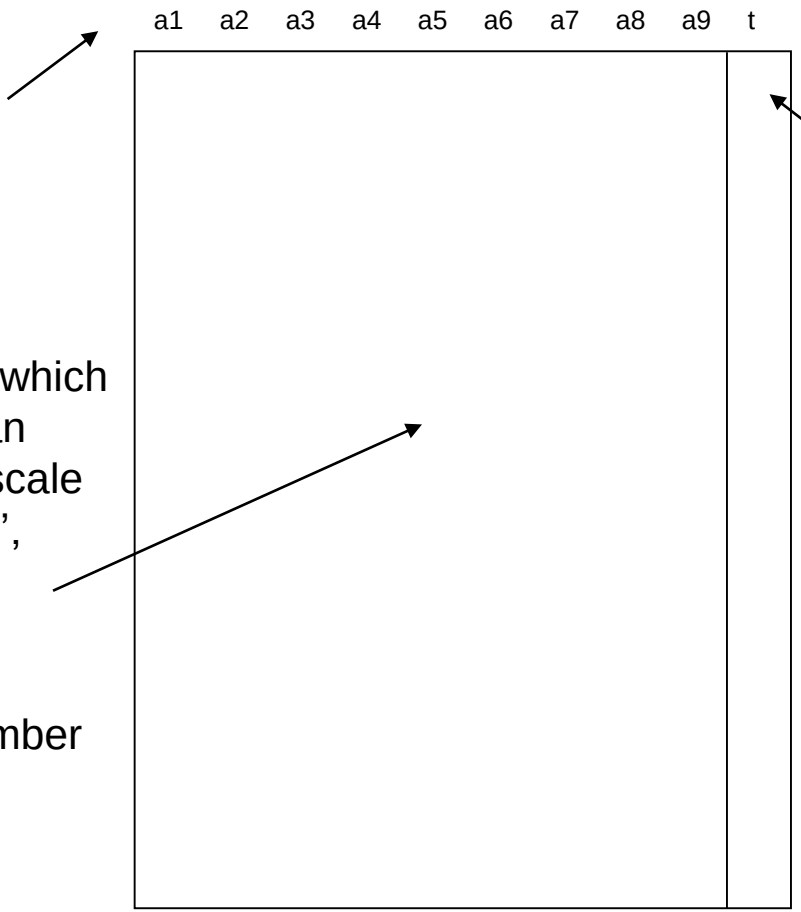
- Overview
- Data Science Methodology 
- Data Understanding
- Data Preparation
- Categories of Machine Learning
- Learning Challenges
- Machine Learning Algorithms
- Model Evaluation

Data Science Methodology



Matrix for Machine Learning

Known as:
- Attributes
- Features
- Predictor variables
- Explanatory variables



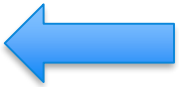
Scale variables:
- Continuous variables, which can be measured on an interval scale or ratio scale
- ‘Weight’, ‘Temperature’, ‘Salary’, etc…

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

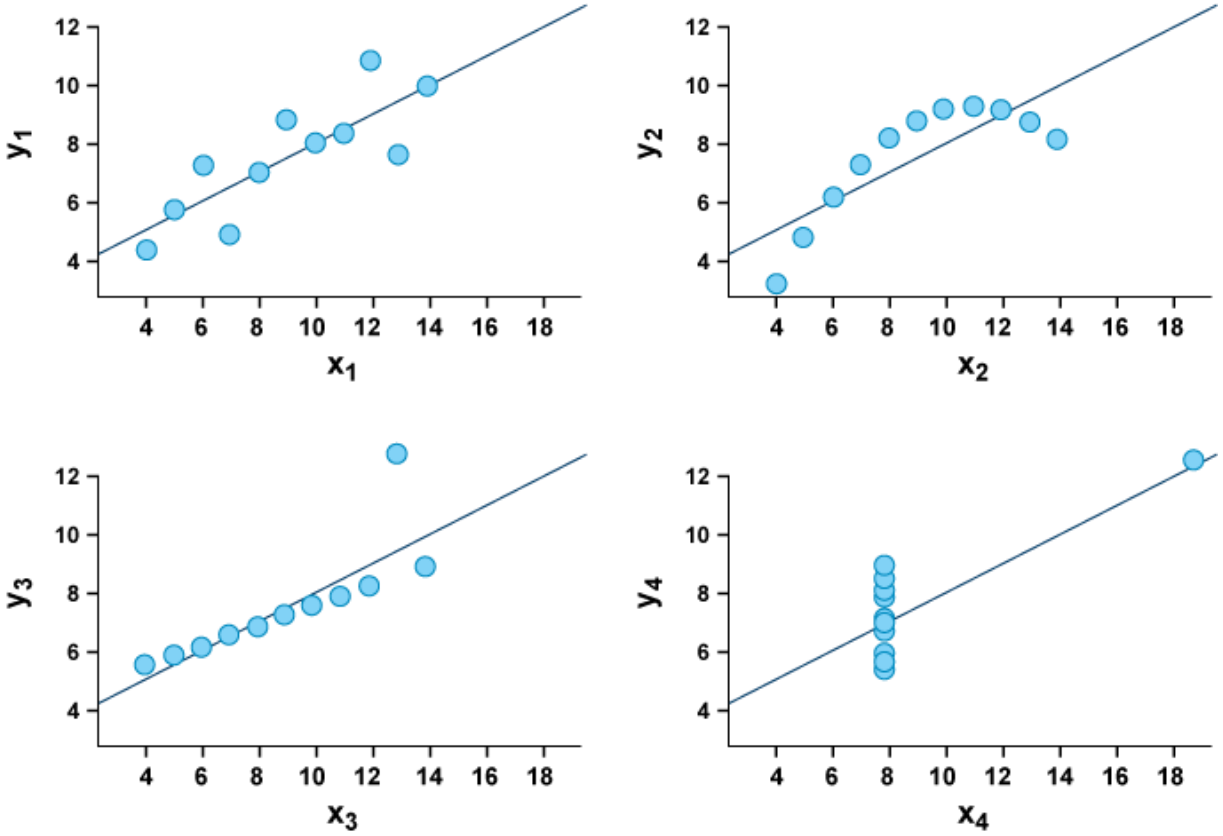
Introduction to Machine Learning

- 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



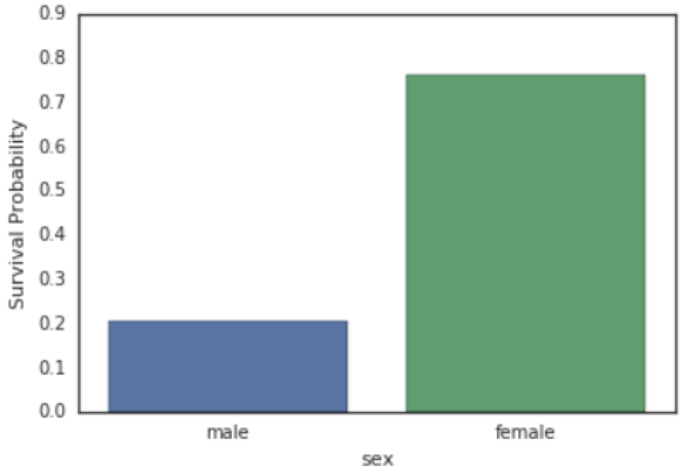
The four data sets have similar statistical properties:
- The mean of x is 9
- The variance of x is 11
- The mean of y is approx. 7.50
- The variance of y is approx. 4.12
- The correlation is 0.816

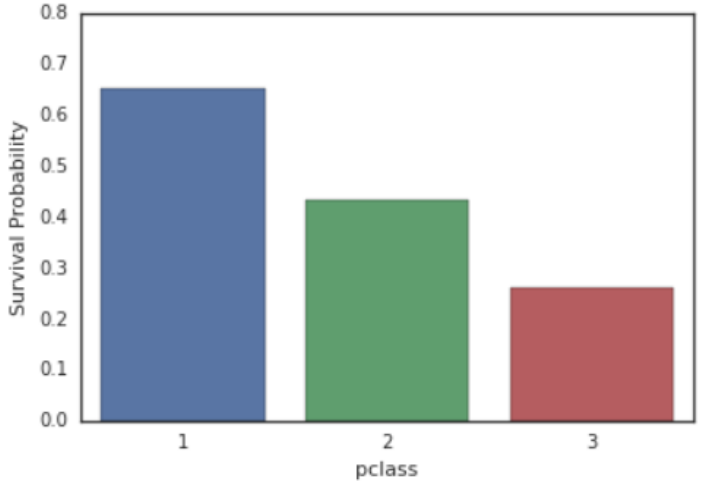
As shown the linear regression lines are approx. y=3.00+0.500x.

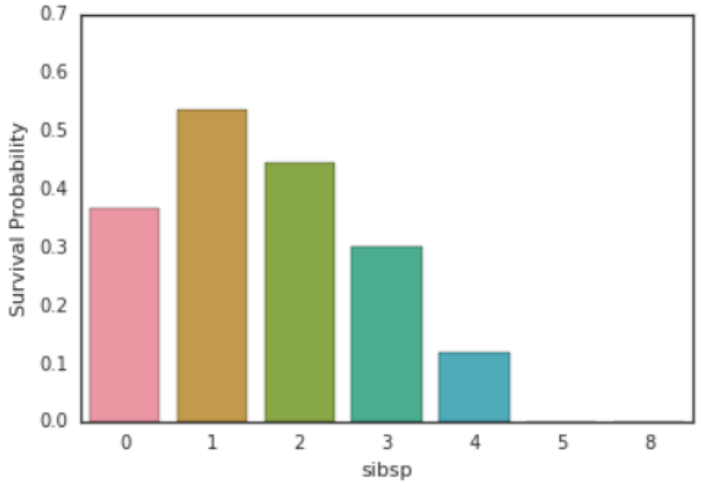
- **Anscombe's quartet**
  - The four datasets have nearly identical statistical properties (mean, variance, correlation), yet the differences are striking when looking at the simple visualization

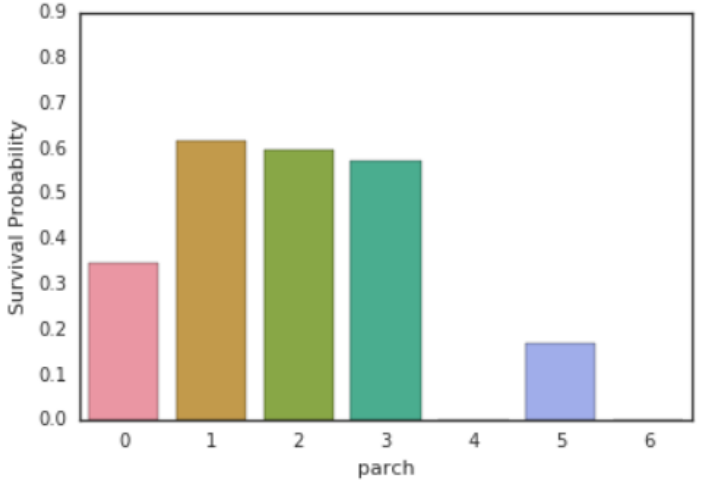
Data Understanding: Visualizations

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



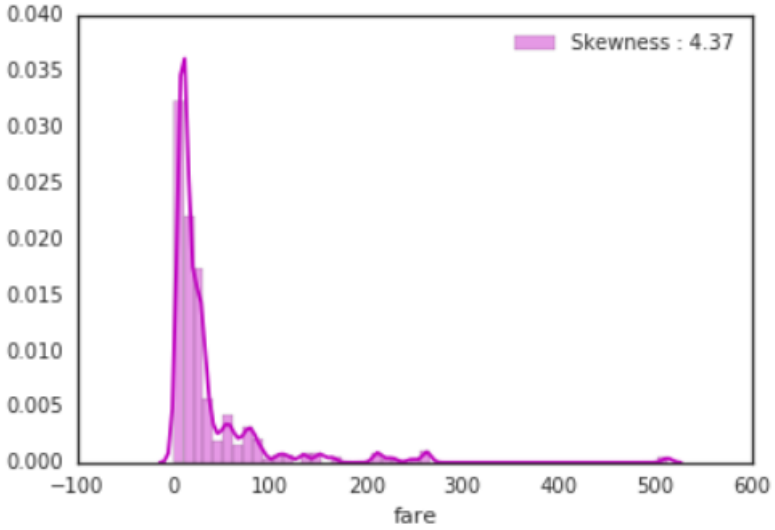




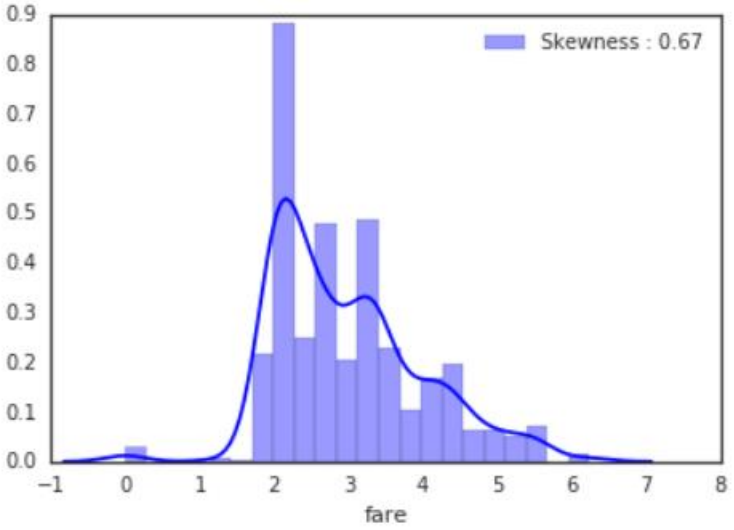


Data Understanding: Visualizations

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

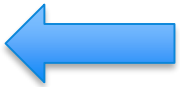


Original Data



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

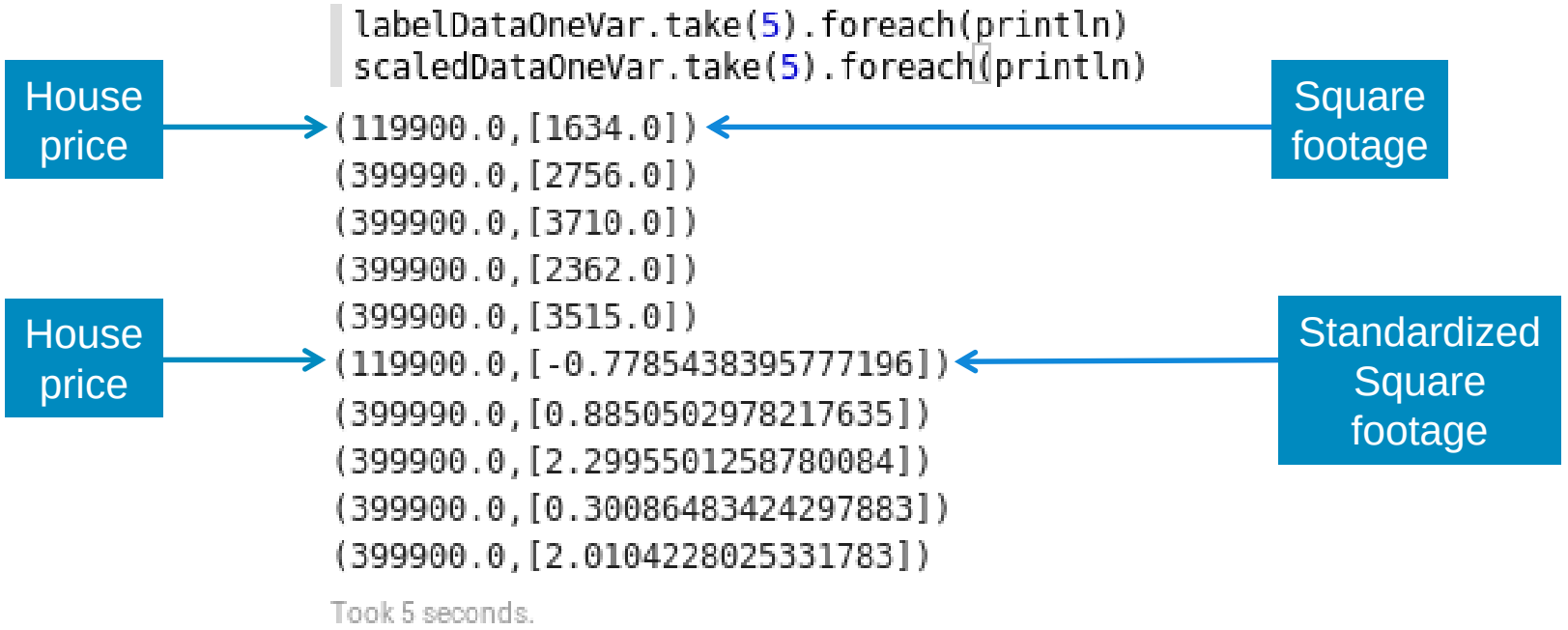
- **Data preparation can be very time consuming depending on:**
  - The state of the original data
    - Data is typically collected in a “human” friendly format
  - The desired final state of the data (as required by the machine learning models and algorithms)
    - The desired final state is typically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

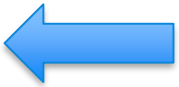
- **Data dimensionality may need to be reduced:**

- **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: fishing, fished, 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

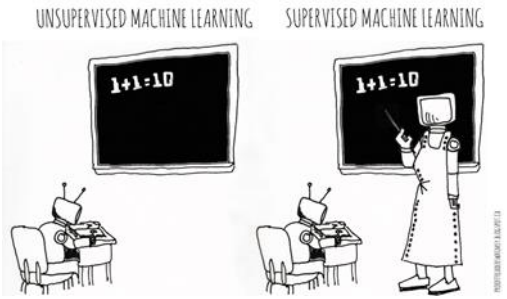
- Is this cancer ? (Medical diagnosis)
- Is this legitimate or fraud (spam) ?
- What is the market value of this house ?
- Which of these people are good friends with each other ?
- Will this engine fail (when) ?
- Will this person like this movie ?
- Who is this ?
- What did you say ? (Speech recognition)

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

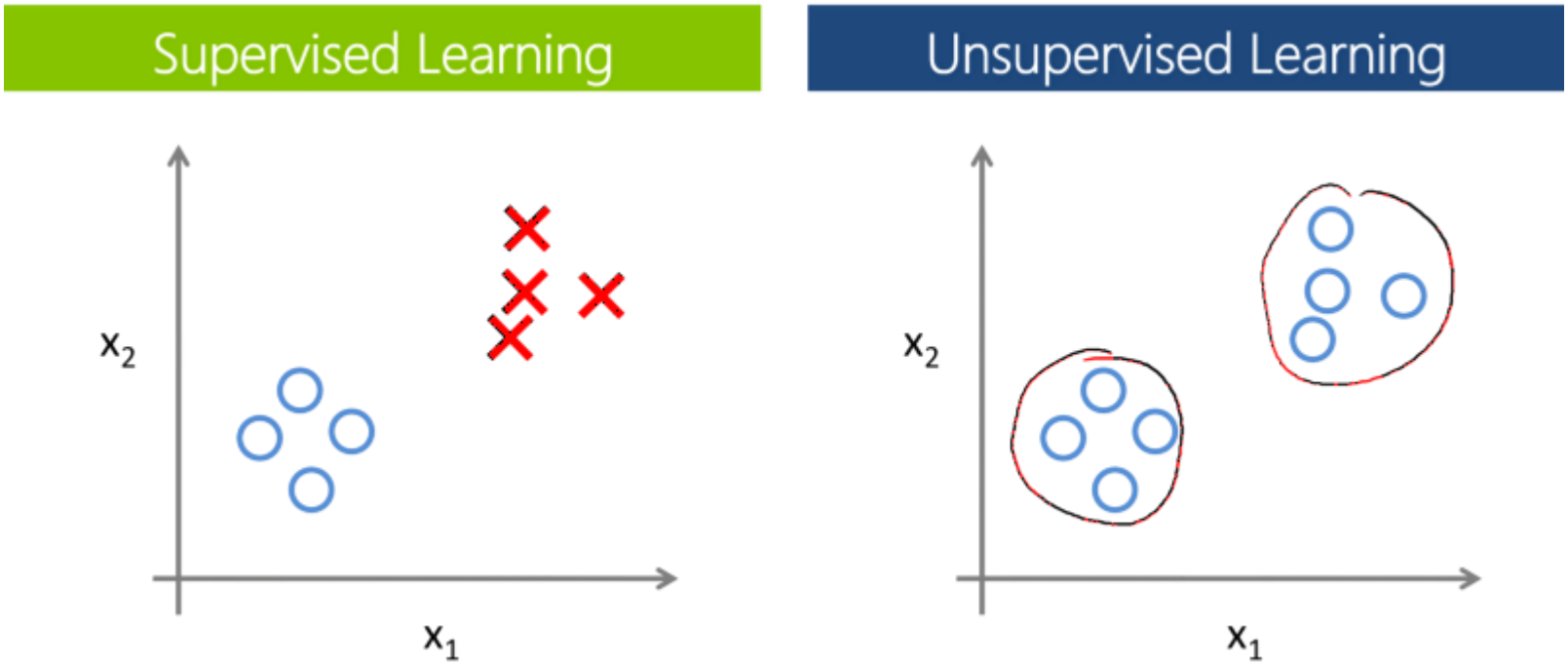
Categories of Machine Learning

- **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

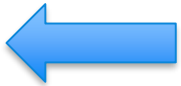


Categories of Machine Learning

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

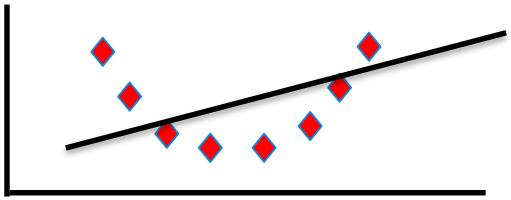
Introduction to Machine Learning

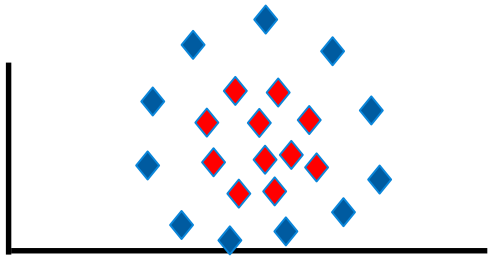
- Overview
- Data Science Methodology
- Data Understanding
- Data Preparation
- Categories of Machine Learning
- Learning Challenges 
- Machine Learning Algorithms
- Model Evaluation

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.

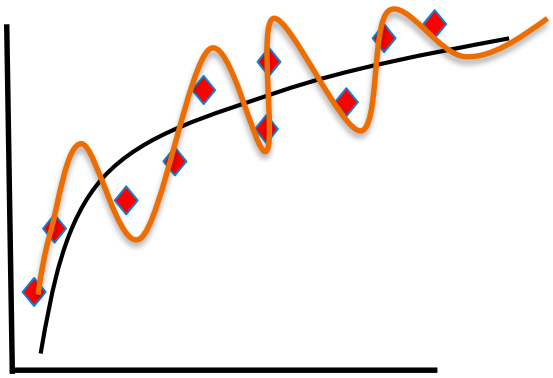
aX + BY + C





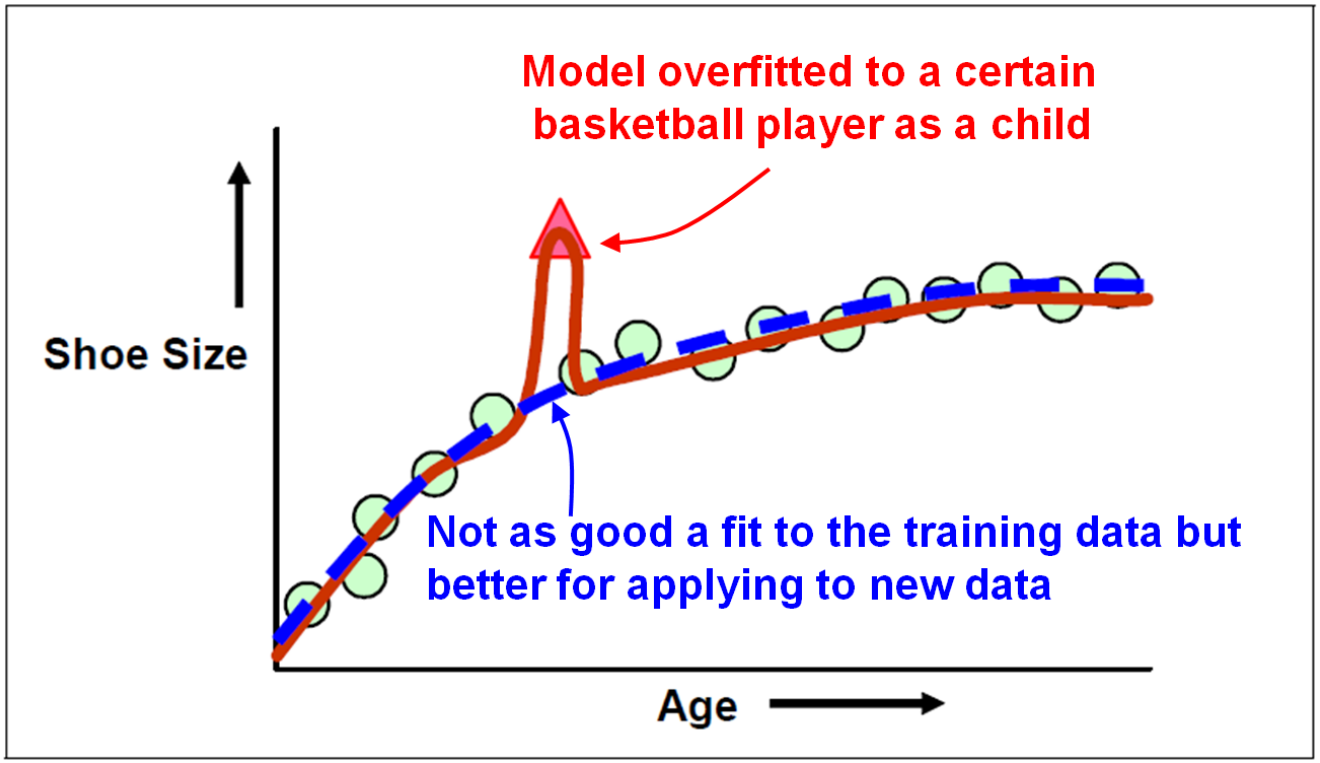
Learning challenges

- **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.



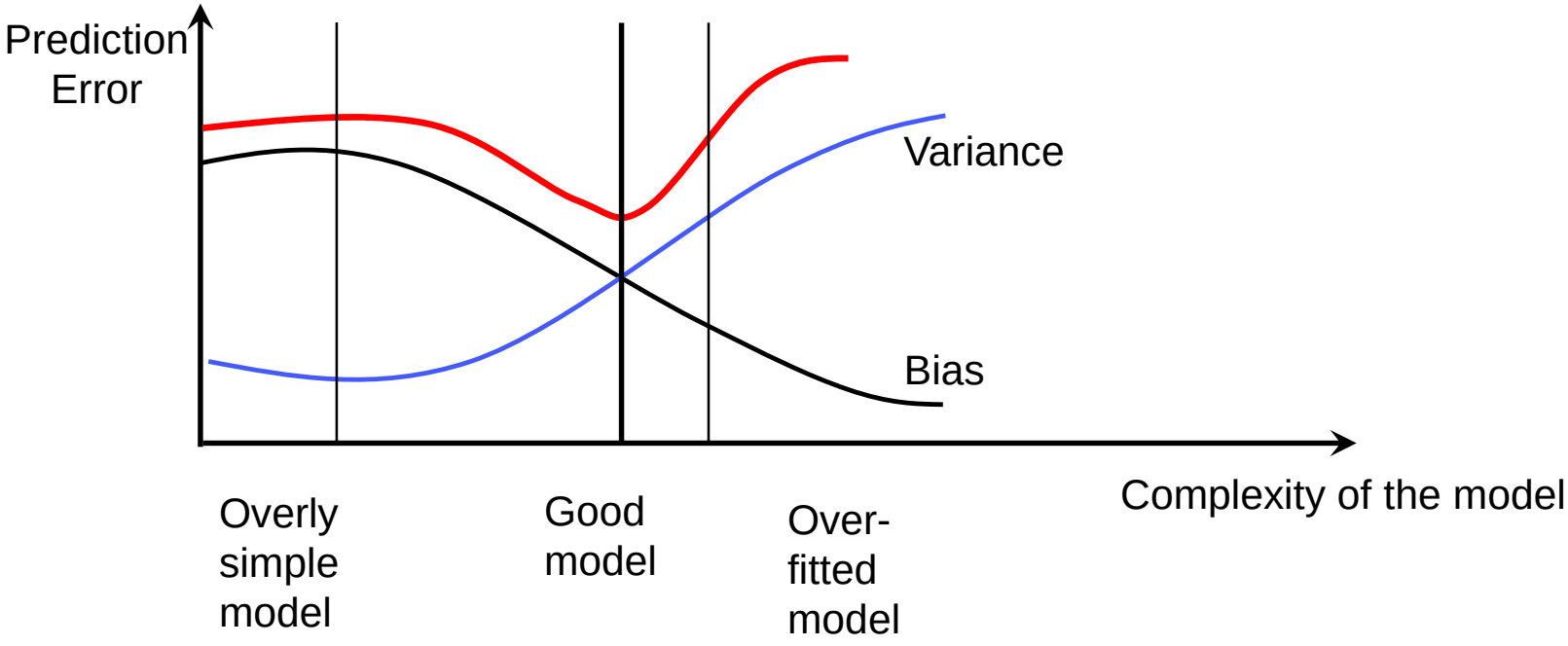
Model overfitting

- 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

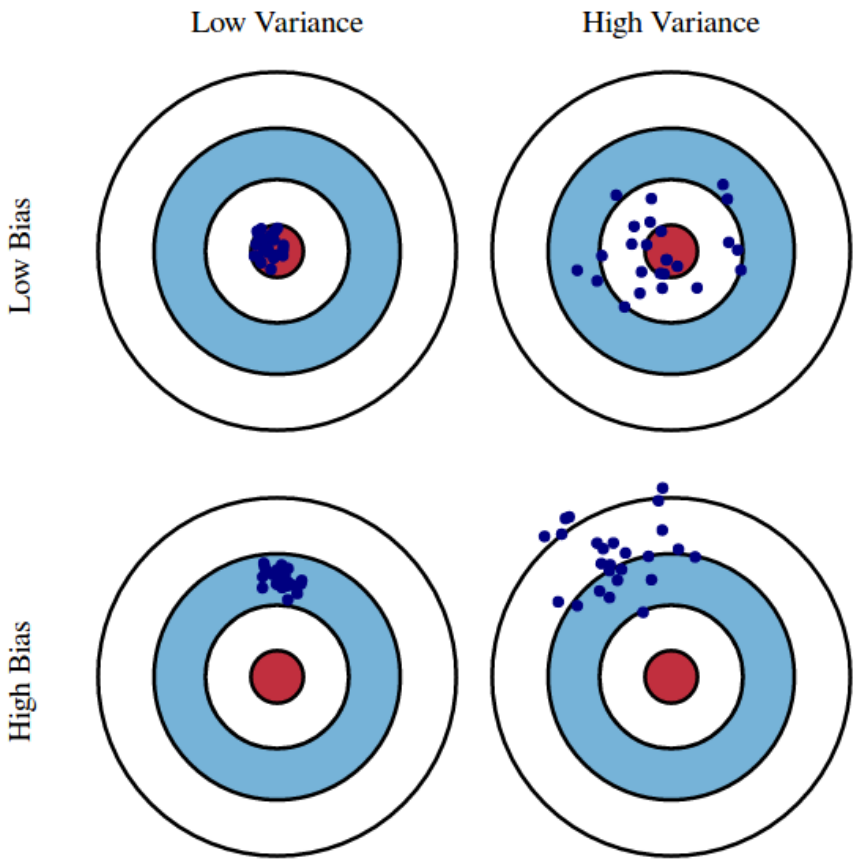


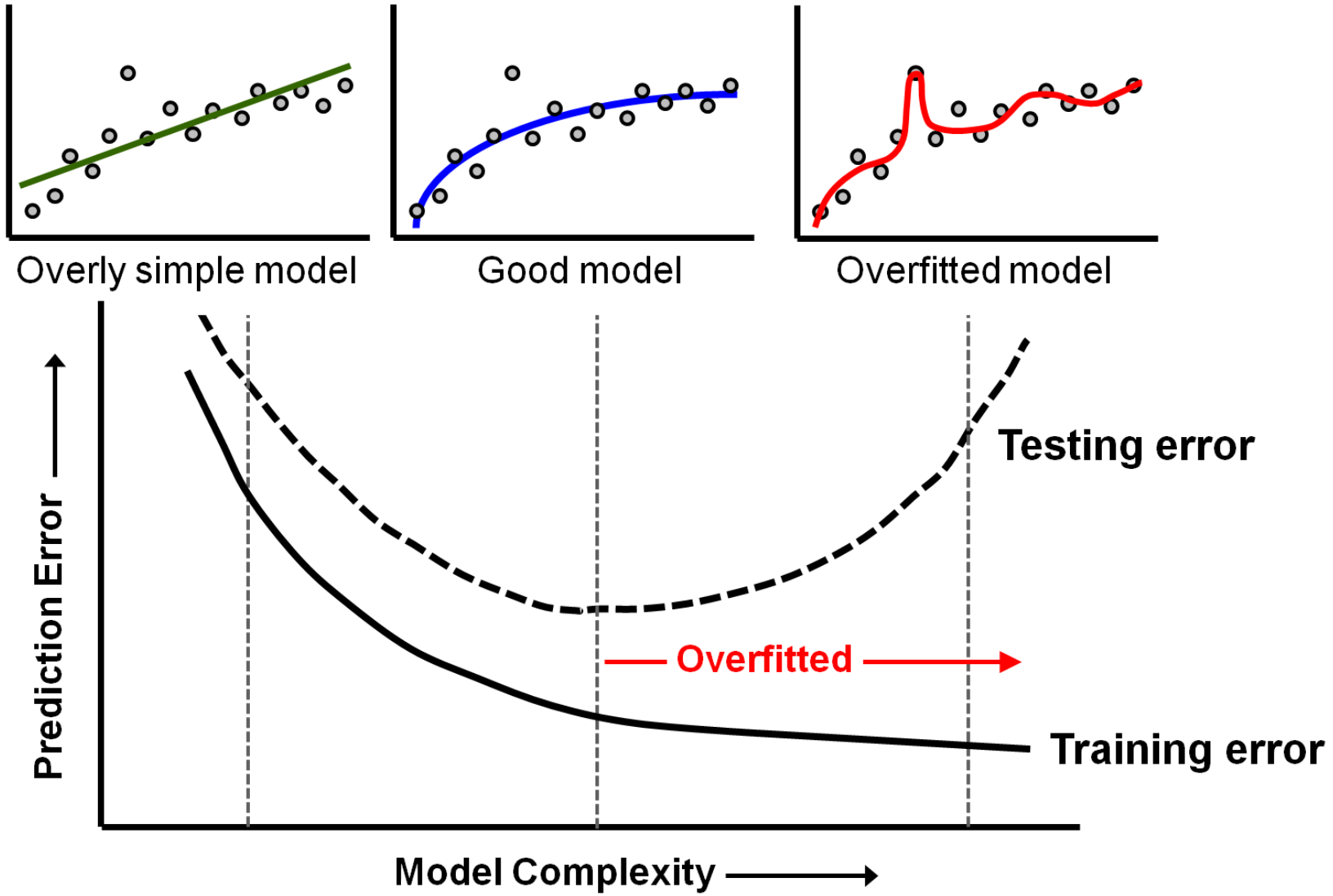
Fig. 1 Graphical illustration of bias and variance.

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

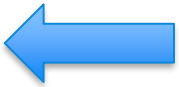
Learning challenges

- **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



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	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

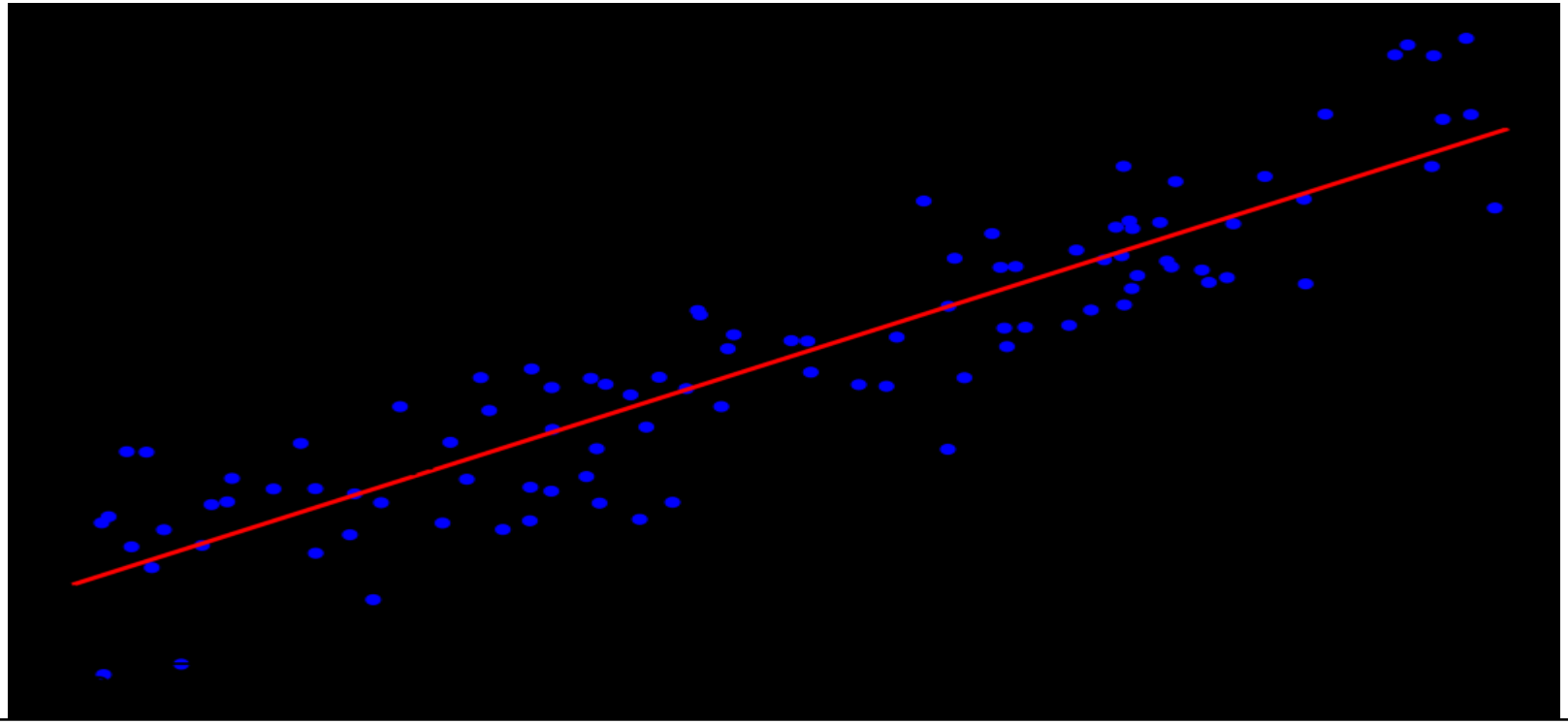
- L(yes) = 2/9 * 3/9 * 3/9 * 3/9 = 0.0082
- L(no) = 3/5 * 1/5 * 4/5 * 3/5 = 0.0577

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

- The decision would be: NO.

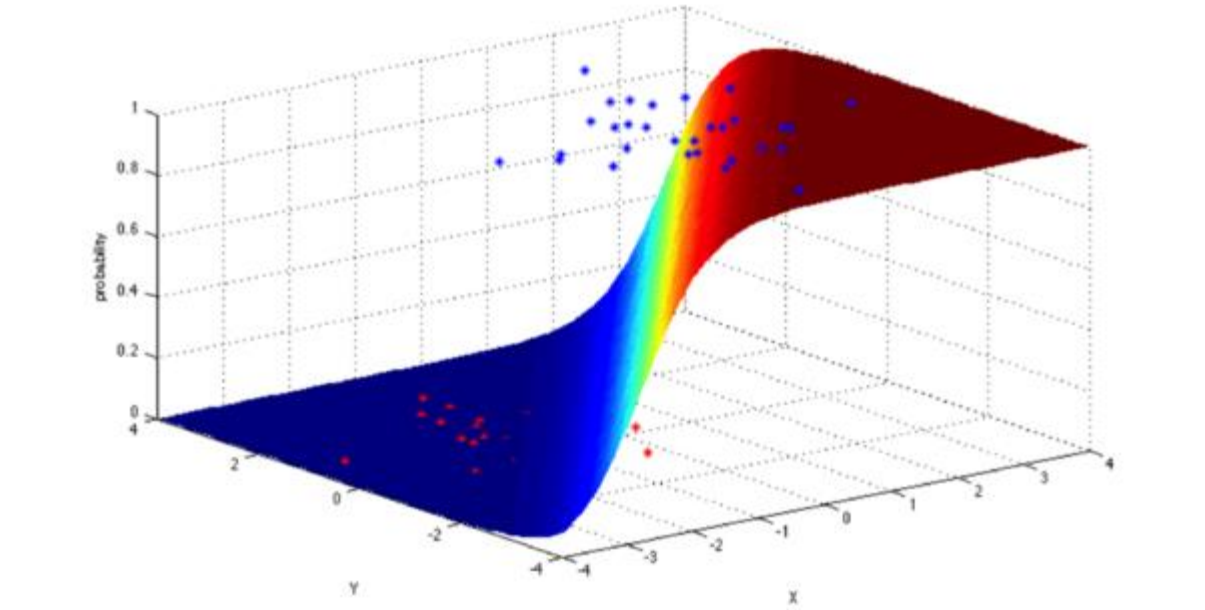
Linear Regression (supervised)

- **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)

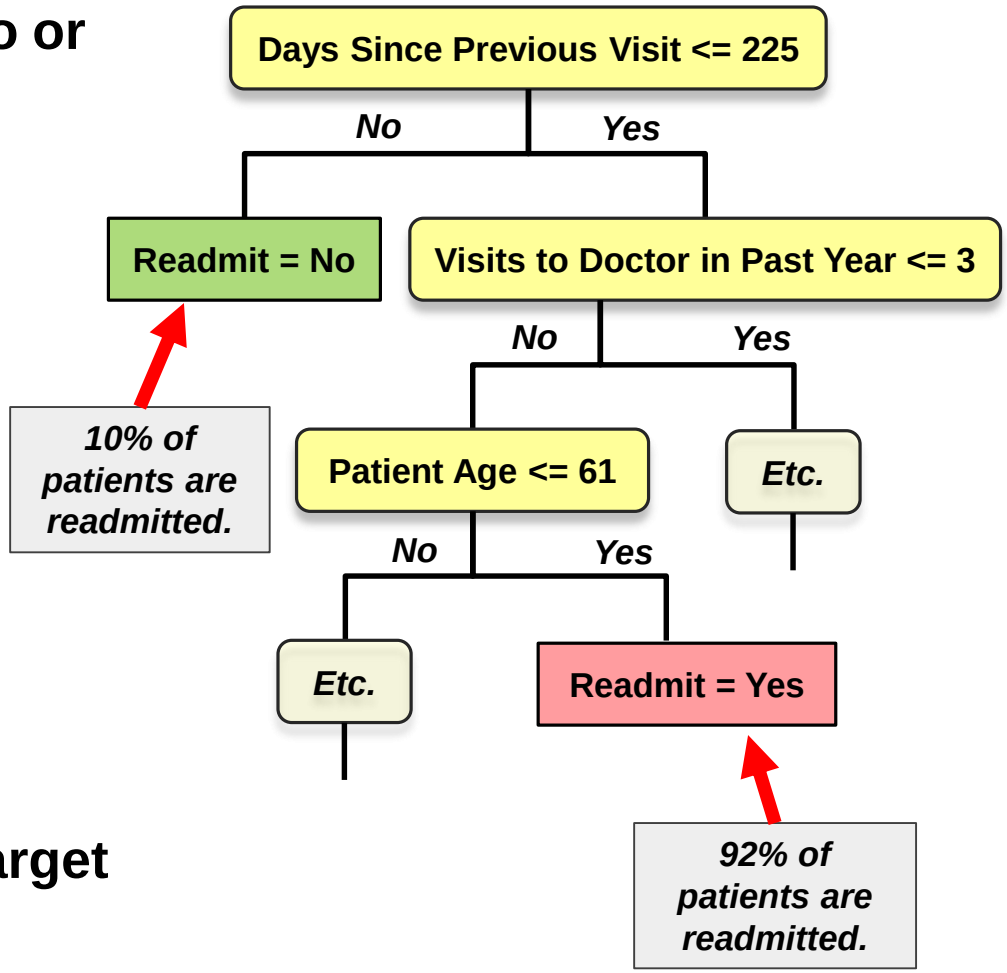
- **Logistic regression is a powerful statistical way of modeling a binomial outcome with one or more explanatory variables. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the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



Classification – Decision tree (supervised)

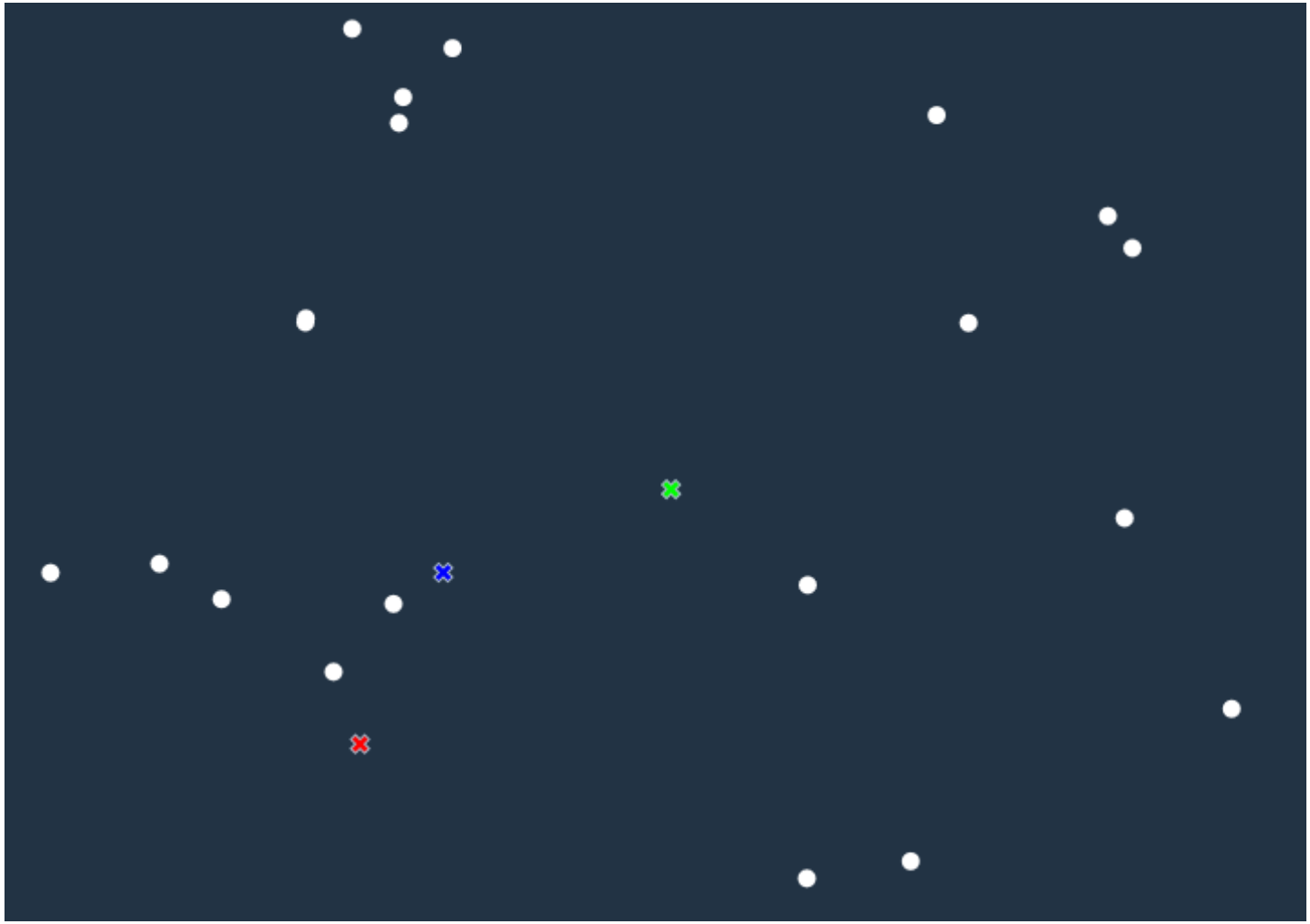
Modeling

- **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").**



Clustering – K-means method

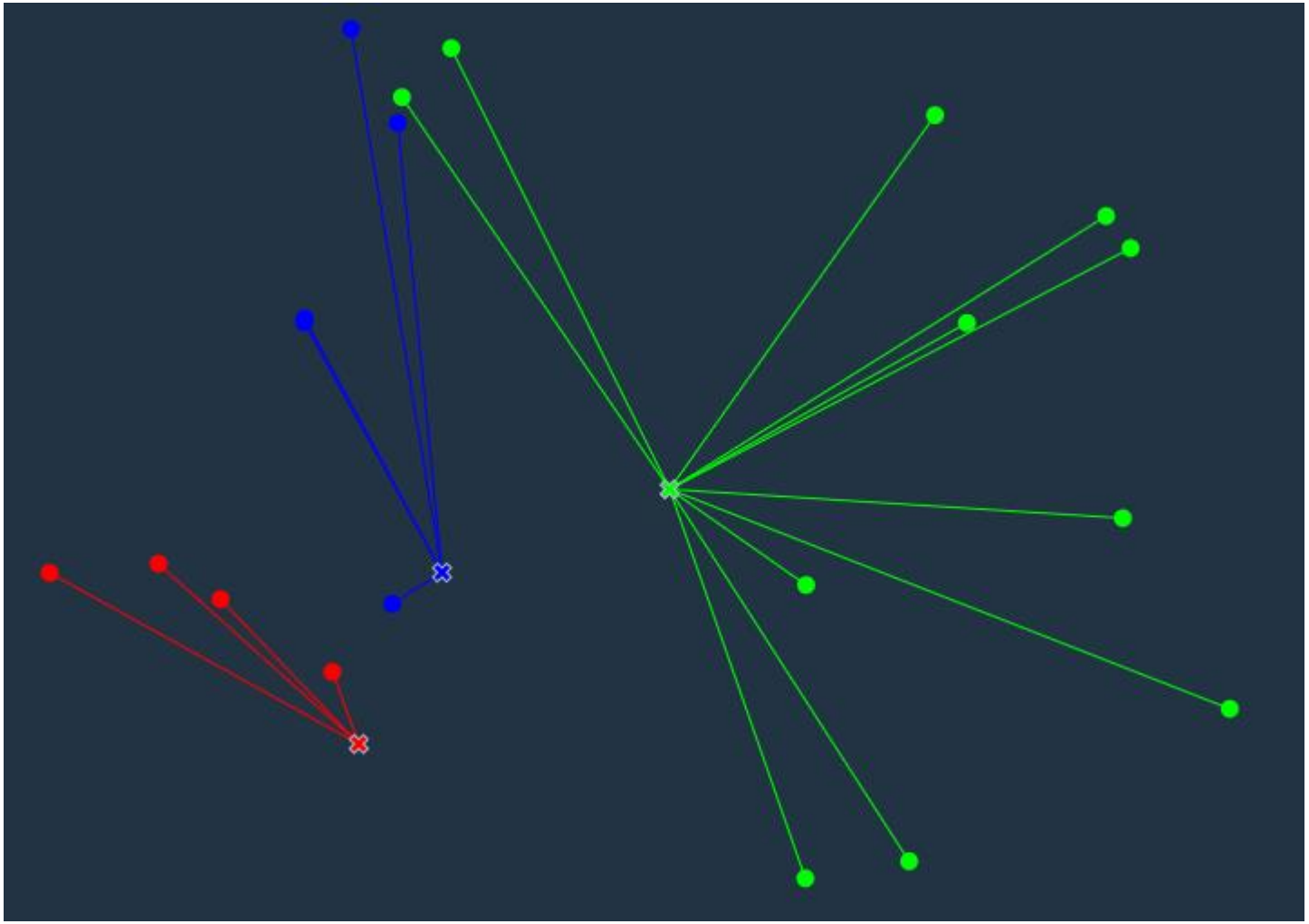
Modeling



Start with 20 data points and 3 clusters

Clustering – K-means method

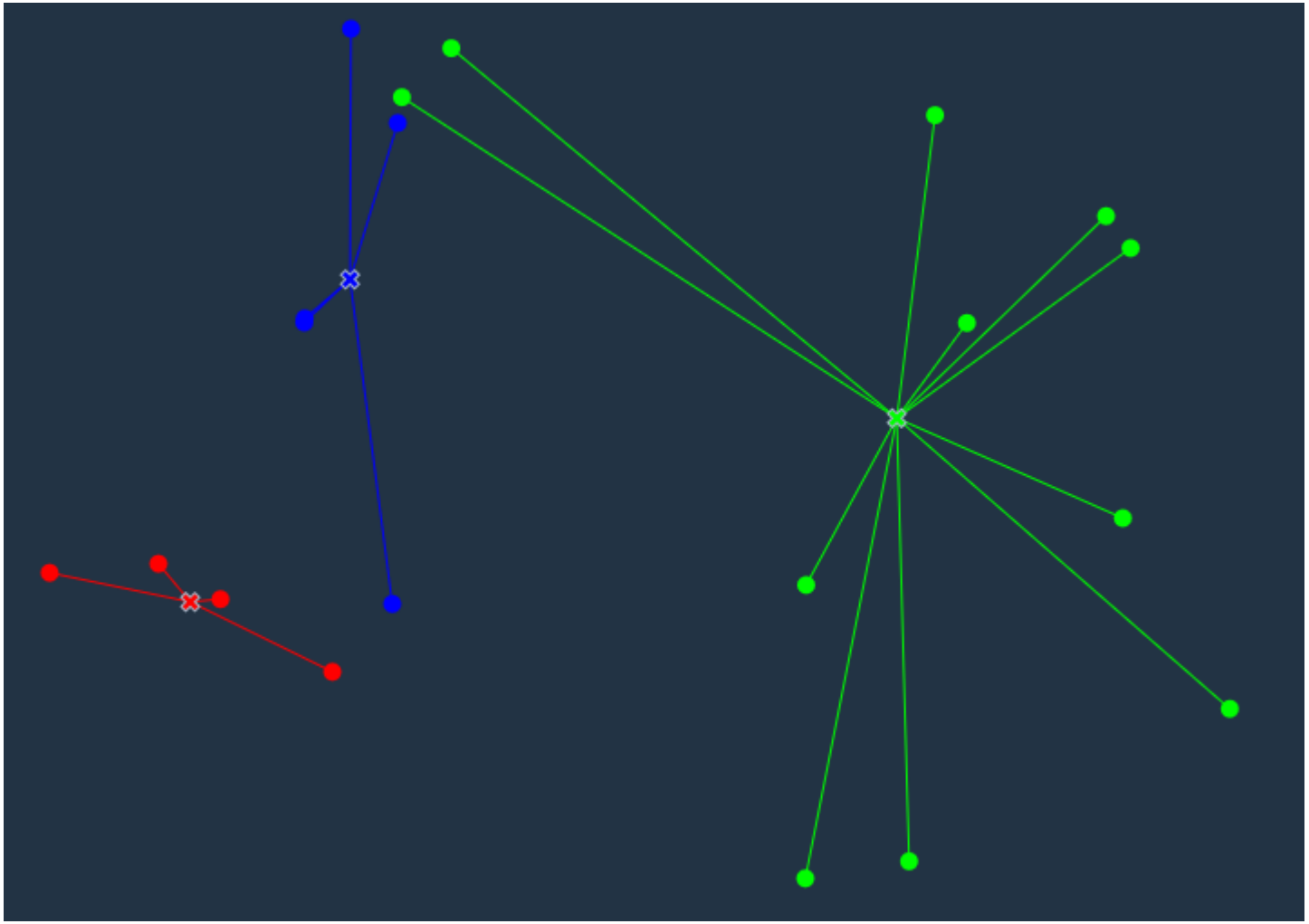
Modeling



Assign each data point to the nearest cluster

Clustering – K-means method

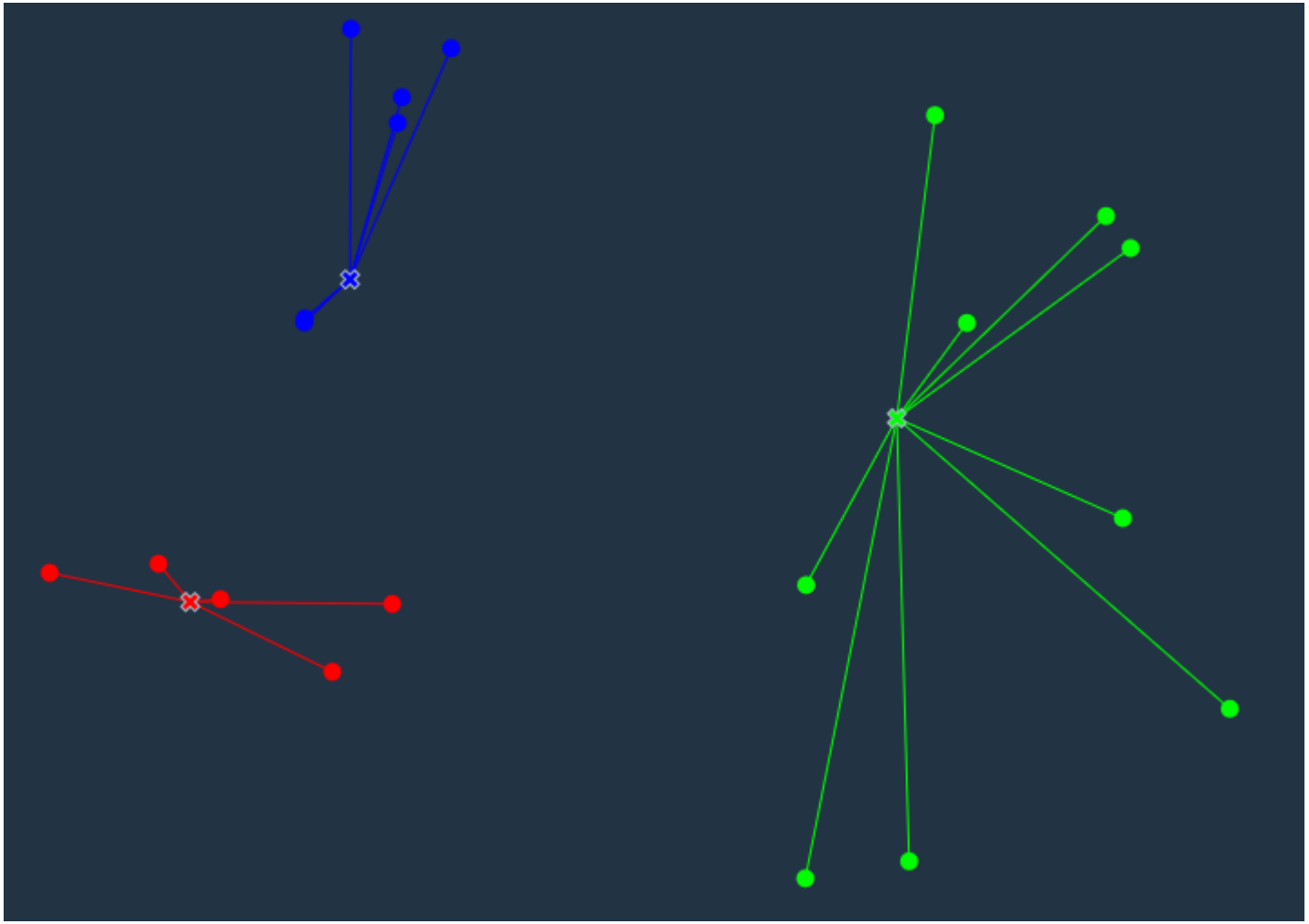
Modeling



Calculate centroids of new clusters

Clustering – K-means method

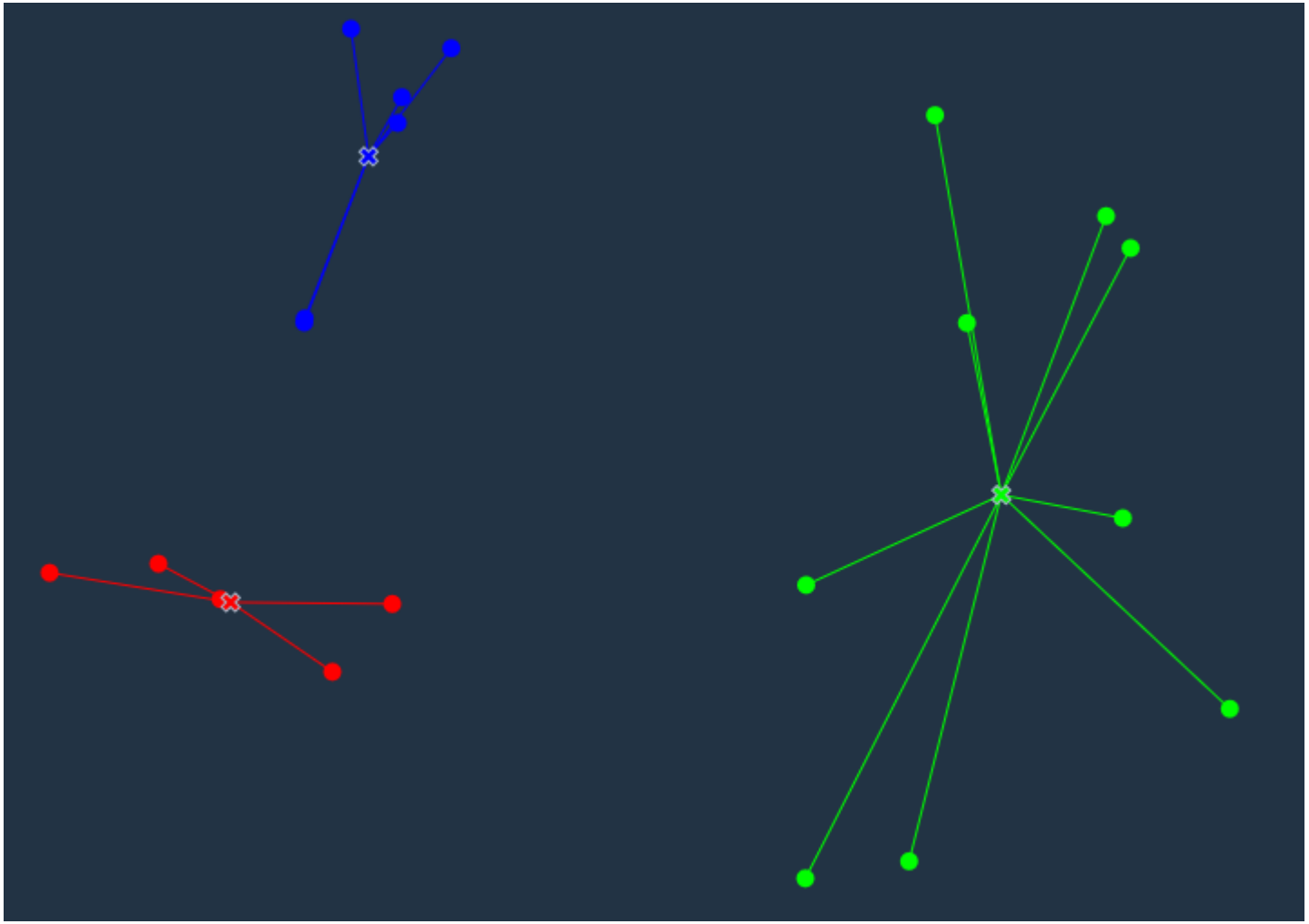
Modeling



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**

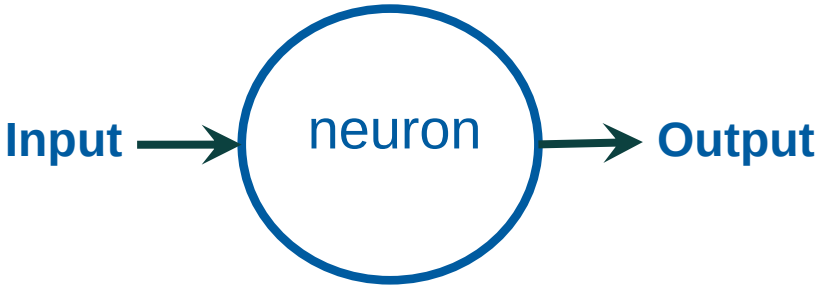
- **Bagging**
  - New training datasets are generated based on random sampling with replacement of the original data set
  - Models are constructed for each sample and the results are combined
  - Random Forest is bagging applied to Decision Trees
- **Boosting**
  - Successive models are built to predict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

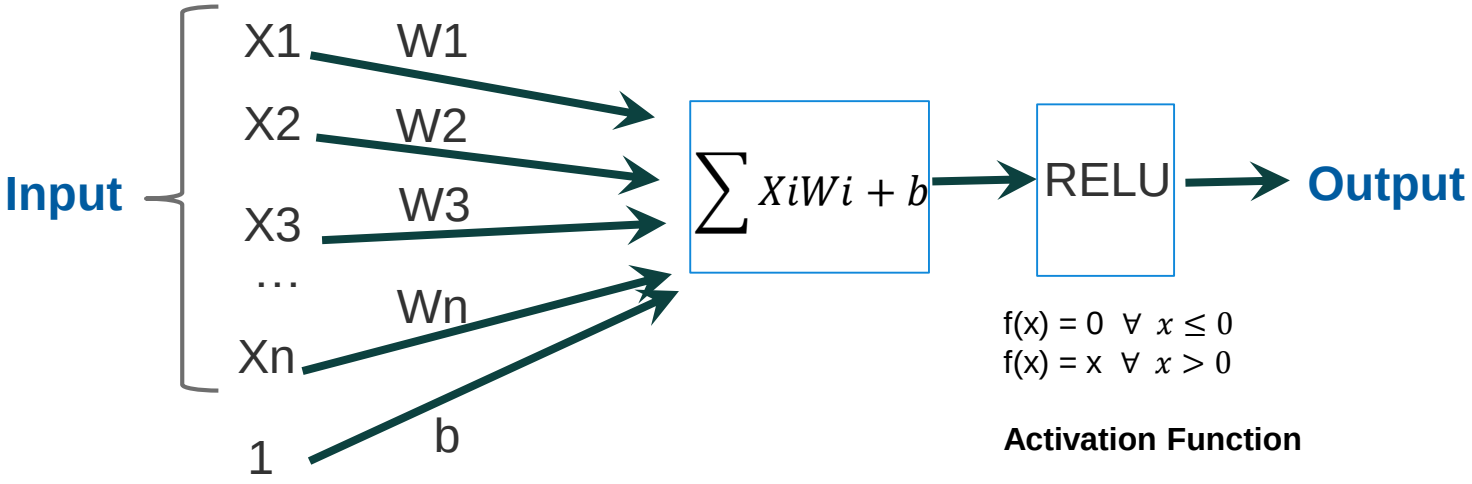
Neural Network

Modeling

- **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

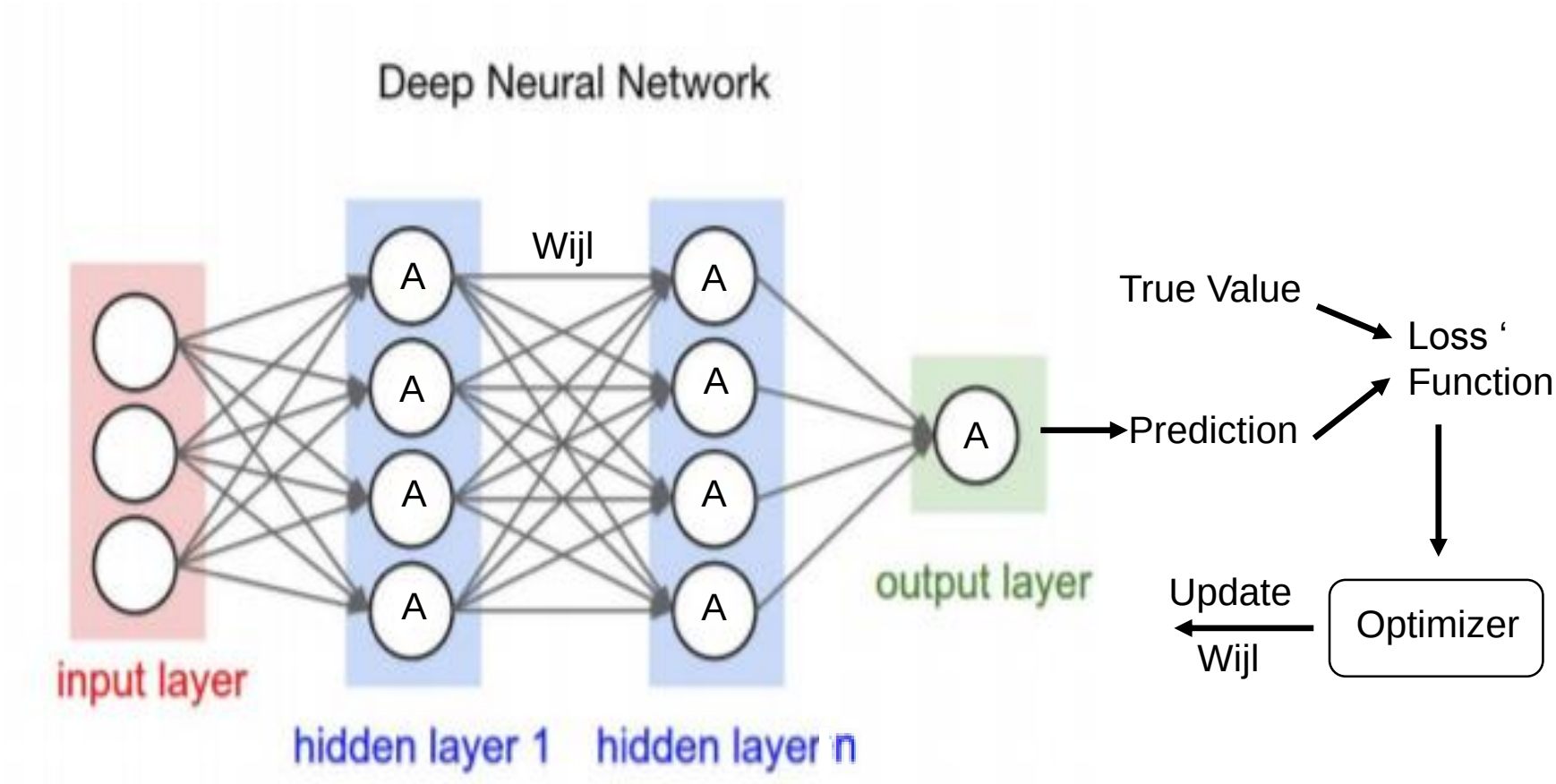




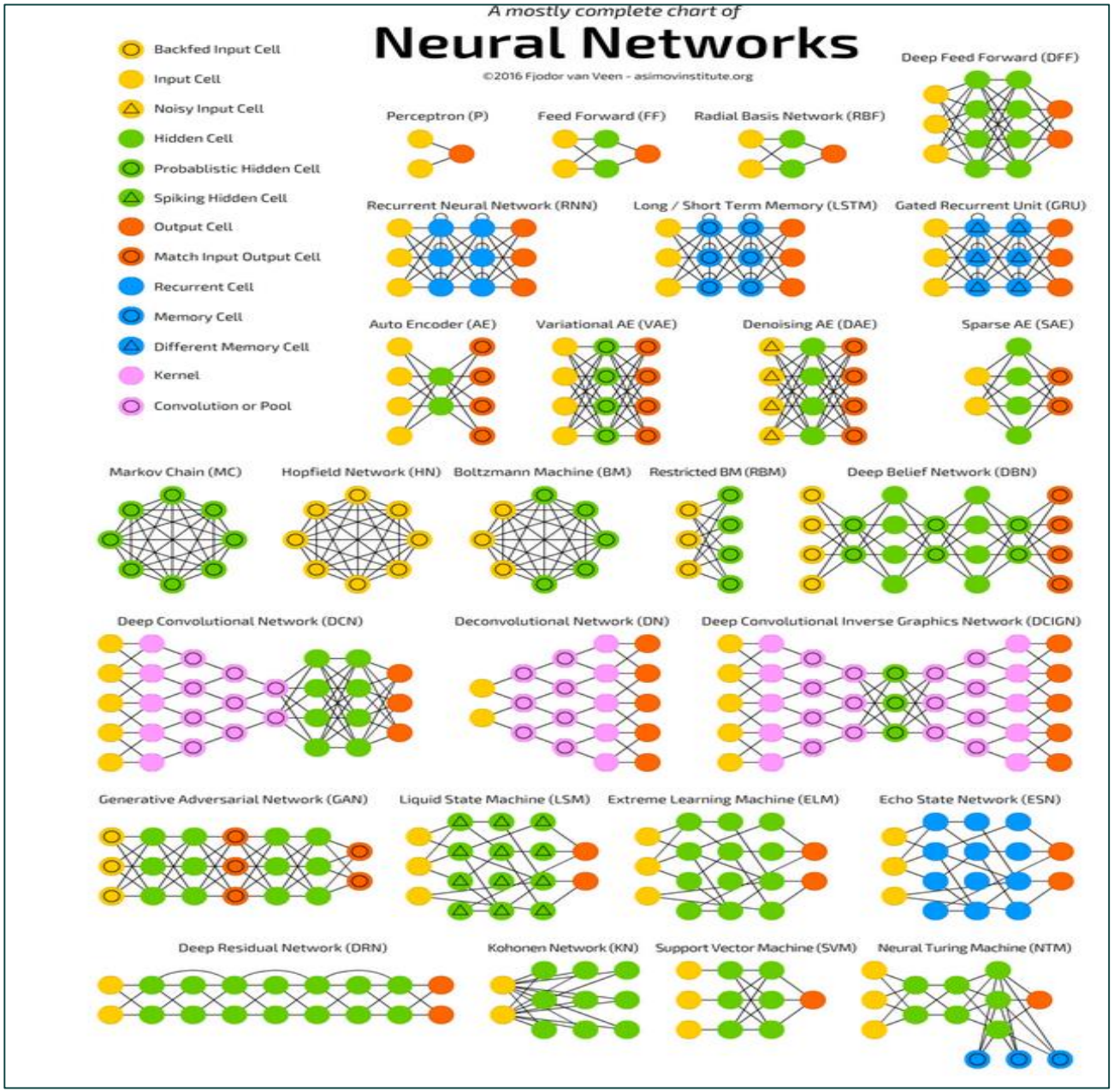
Neural Network

Modeling

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



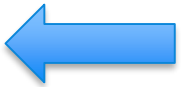
Wijl – weight from neuron (i) in level (l-1) to neuron (j) in level (l)



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

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

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

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

Model Evaluation: 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

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

Precision = tp/(tp + fp)

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

FPR = fp/(fp + tn) 1 – specificity

ROC = plot of TPR/FPR at different thresholds

Model Evaluation: 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

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

Precision = tp/(tp + fp)

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

FPR = fp/(fp + tn) 1 – specificity

ROC = plot of TPR/FPR at different thresholds

Model Evaluation: 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

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

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

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

Model Evaluation

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

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

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

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.

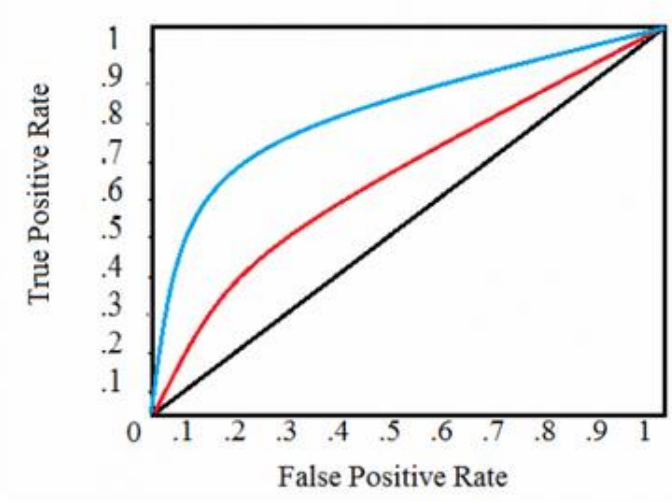
Model Evaluation - Metrics

- $Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} = \frac{Tp+Tn}{Tp+Tn+Fp+Fn}$

- $F1 = 2 * \frac{Recall*Precision}{Recall+Precision}$

- **Area under Receiver Operating Characteristic (ROC)**

Source



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

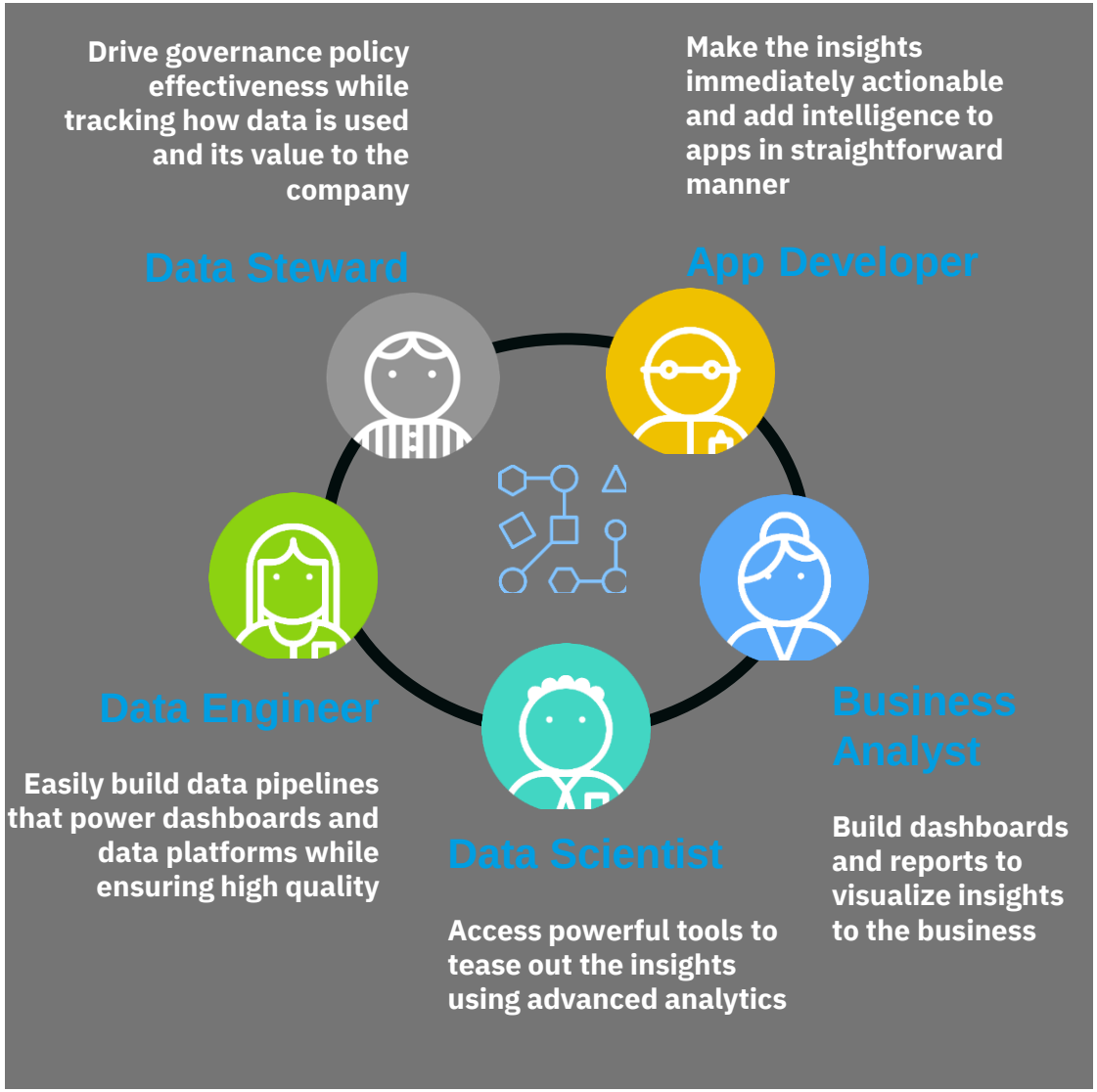
- Data Science skills are in low supply and high demand

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 meta-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 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.

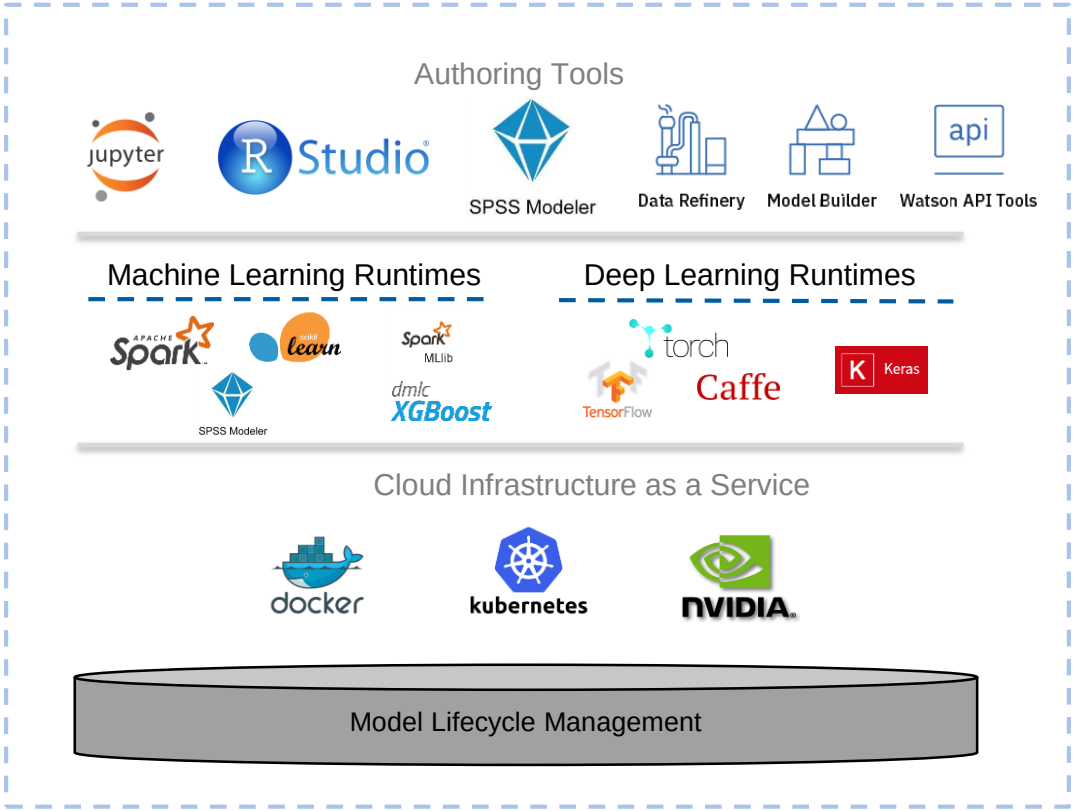
76

Watson Studio Tools

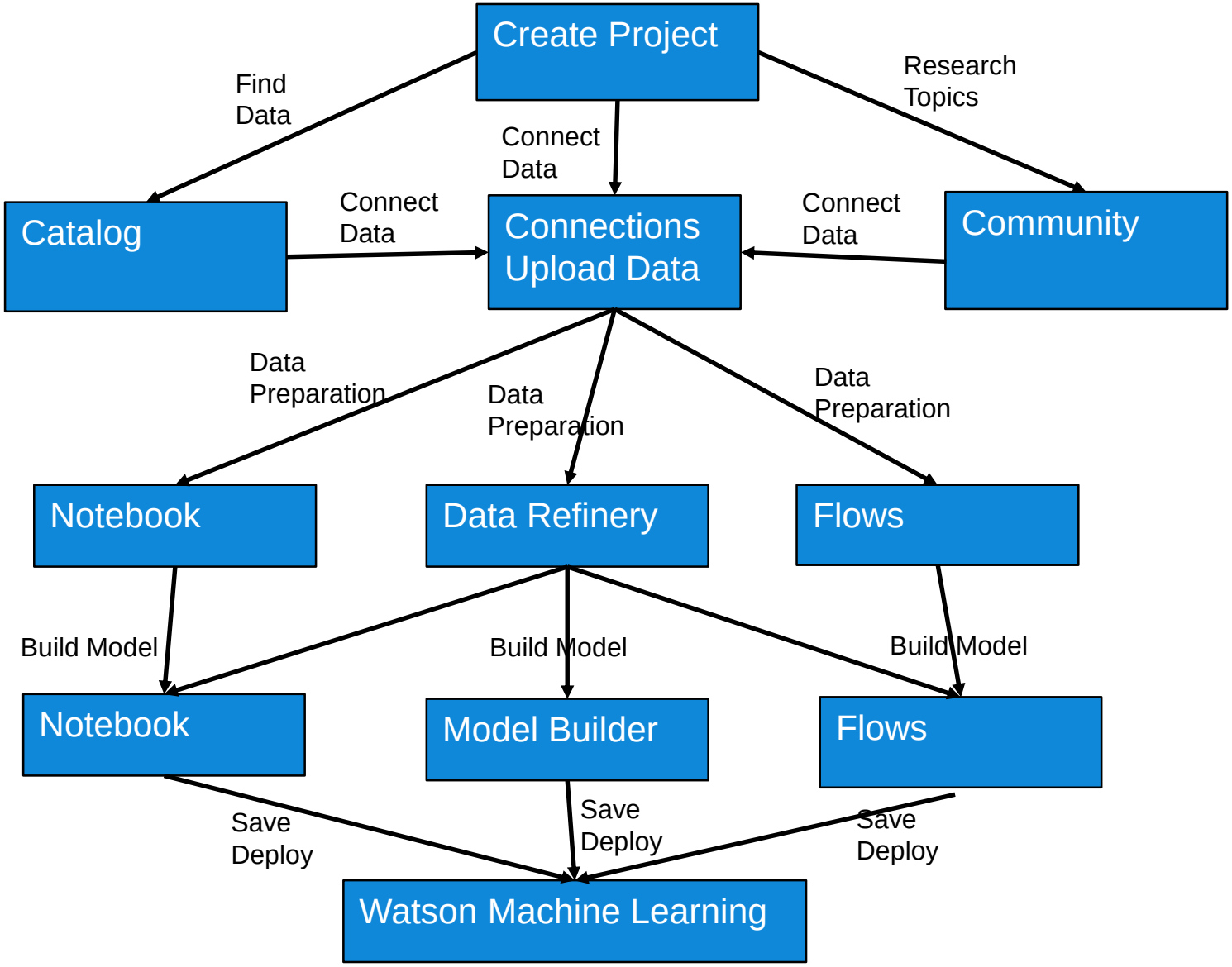
- Create, collaborate, deploy, and monitor
- Best of breed open source & IBM tools
- Code (R, Python or Scala) and no-code/visual modeling tools

- Open Source and IBM libraries/frameworks

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

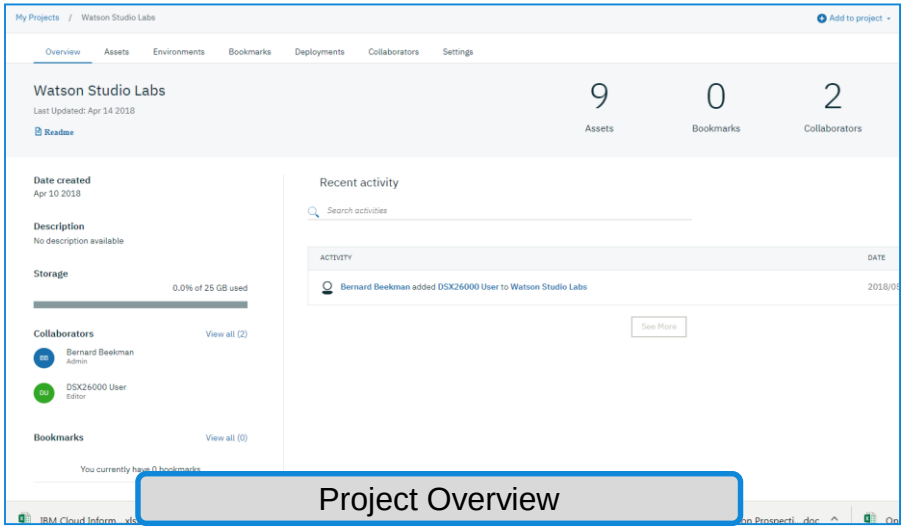


Project Flow

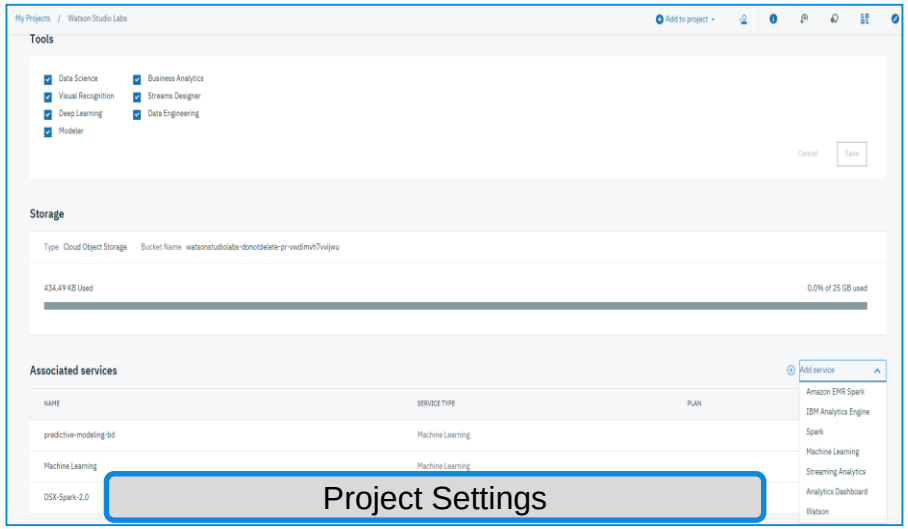


Watson Studio – Projects

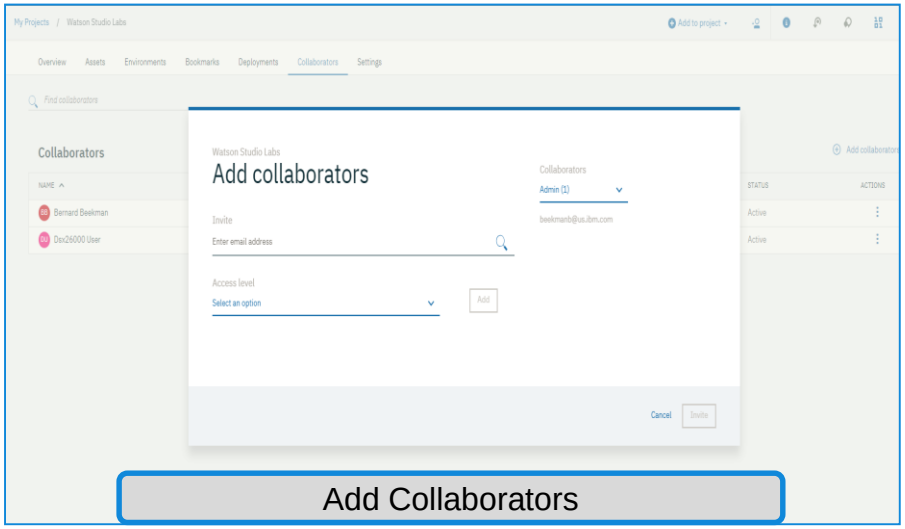
Making Data Science a Team Sport



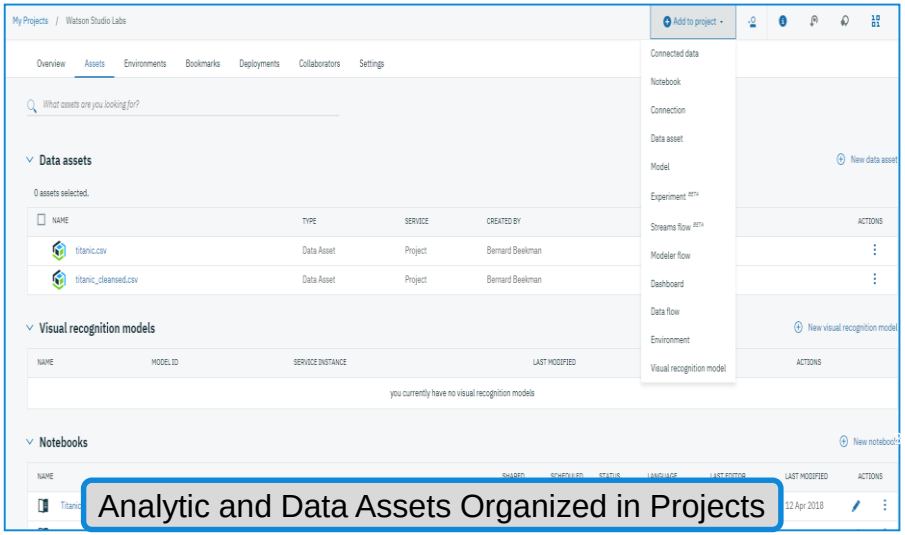
Project Overview



Project Settings



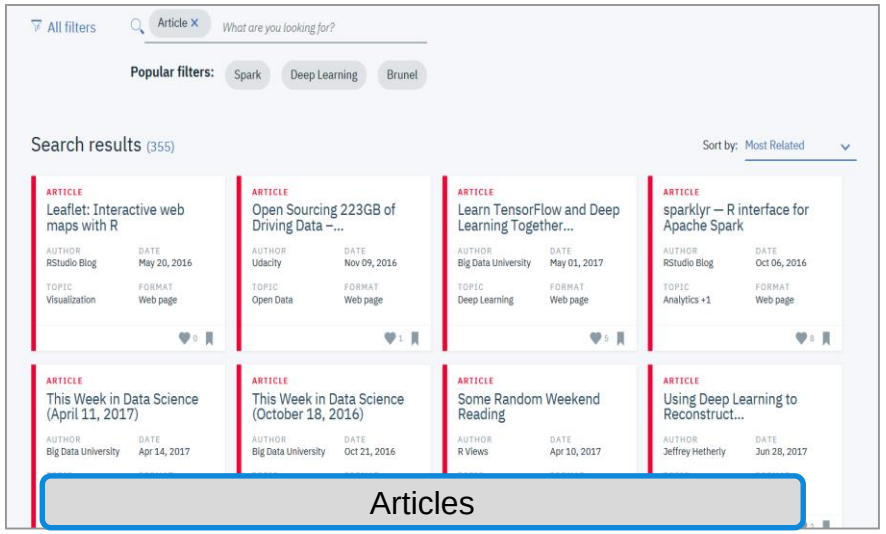
Add Collaborators

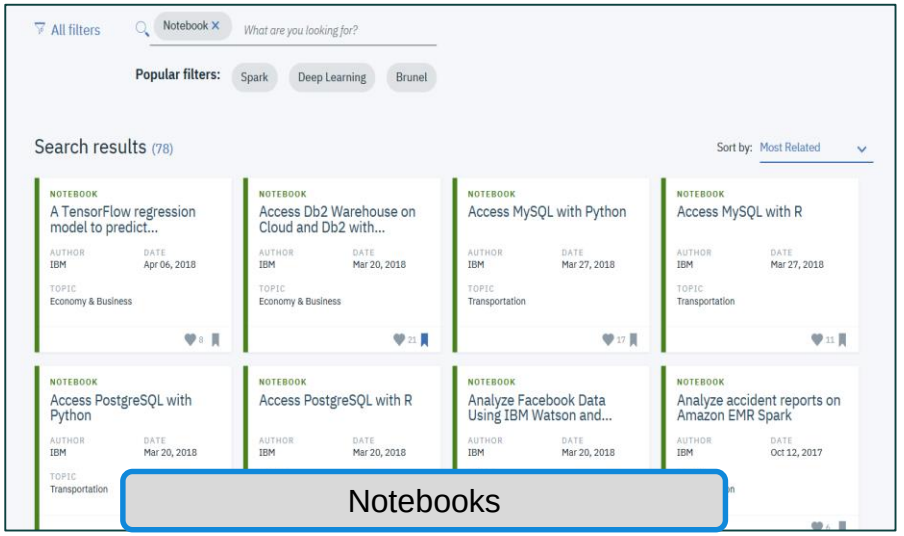


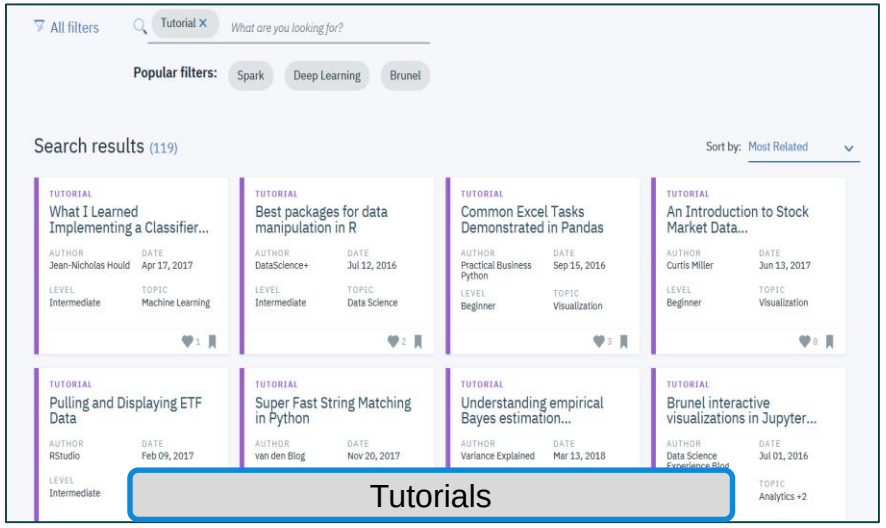
Analytic and Data Assets Organized in Projects

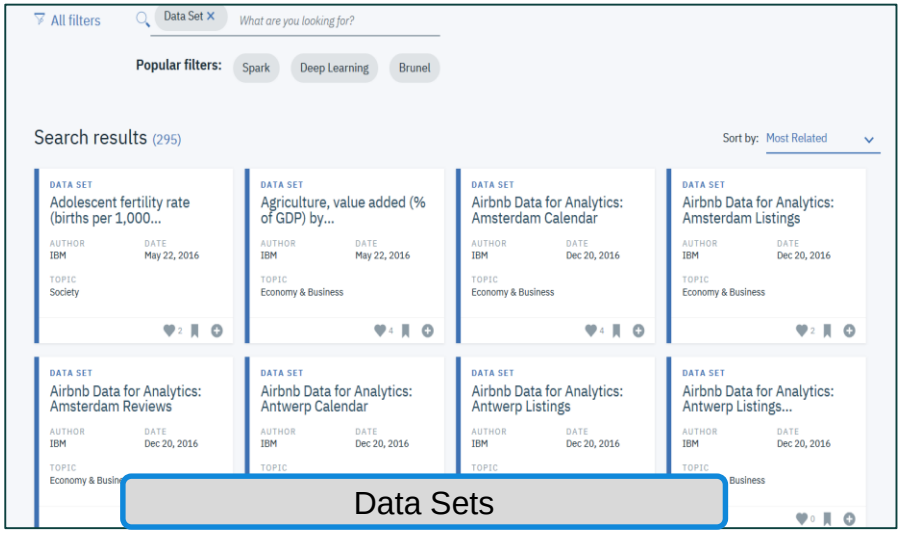
Watson Studio – Community Cards

Built-in learning to get started



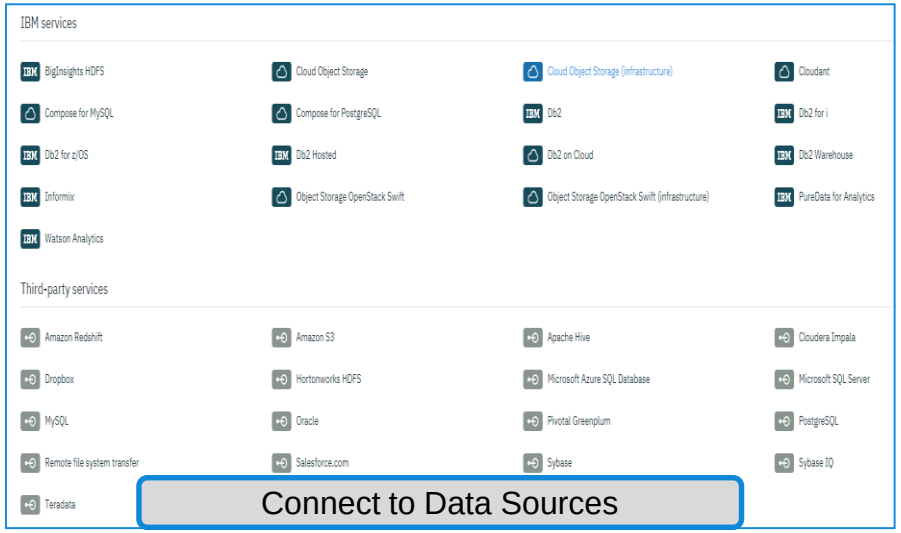




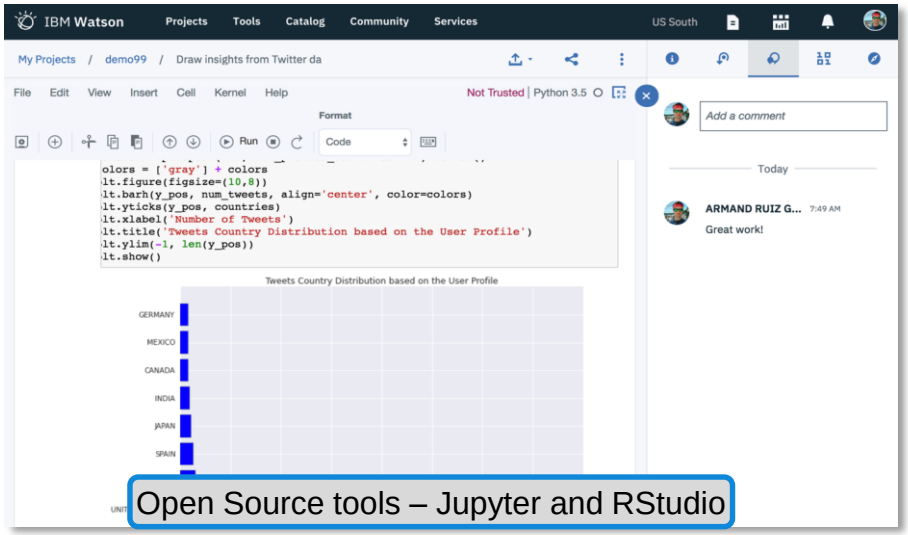


Watson Studio – Create Assets

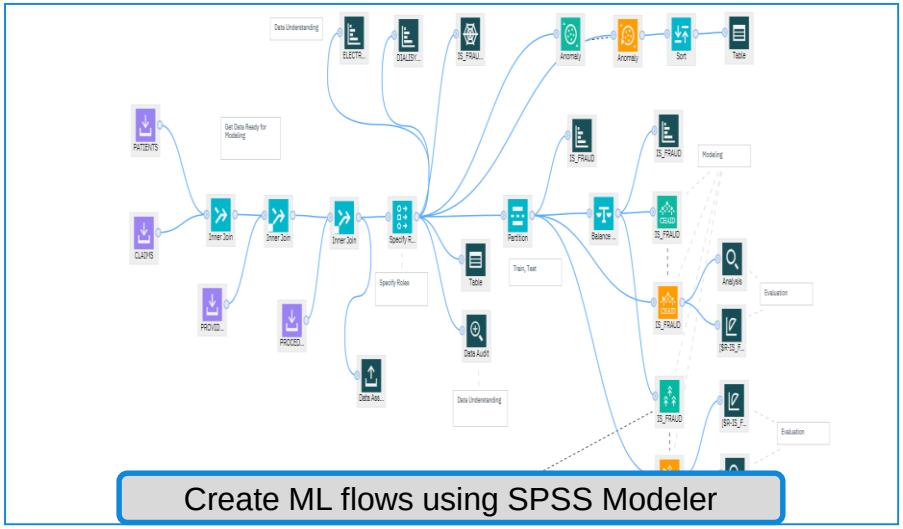
The best of open source and IBM Watson tools to create start-of-the-art data products



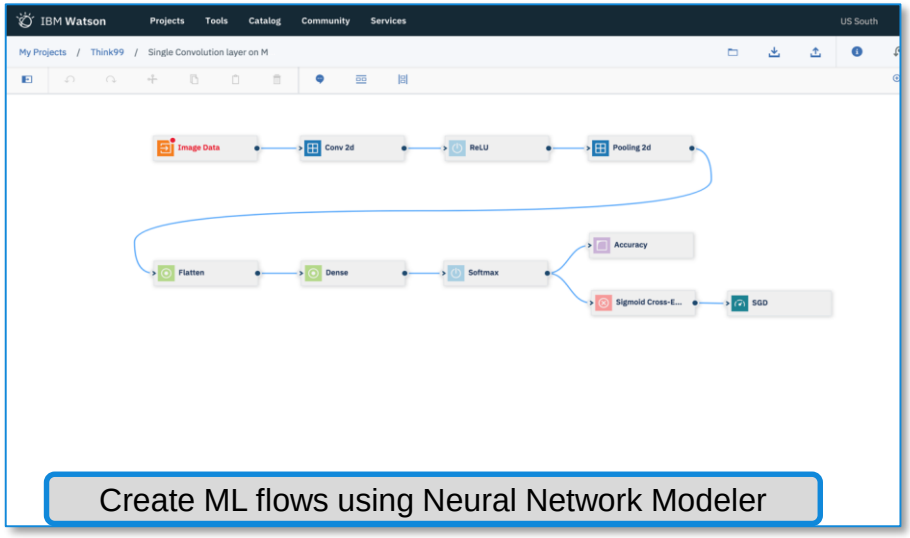
Connect to Data Sources



Open Source tools – Jupyter and RStudio



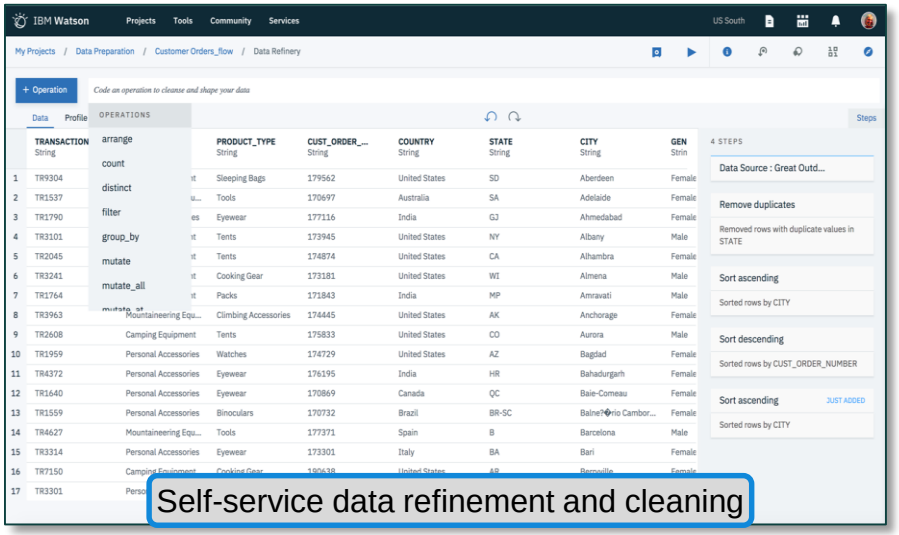
Create ML flows using SPSS Modeler



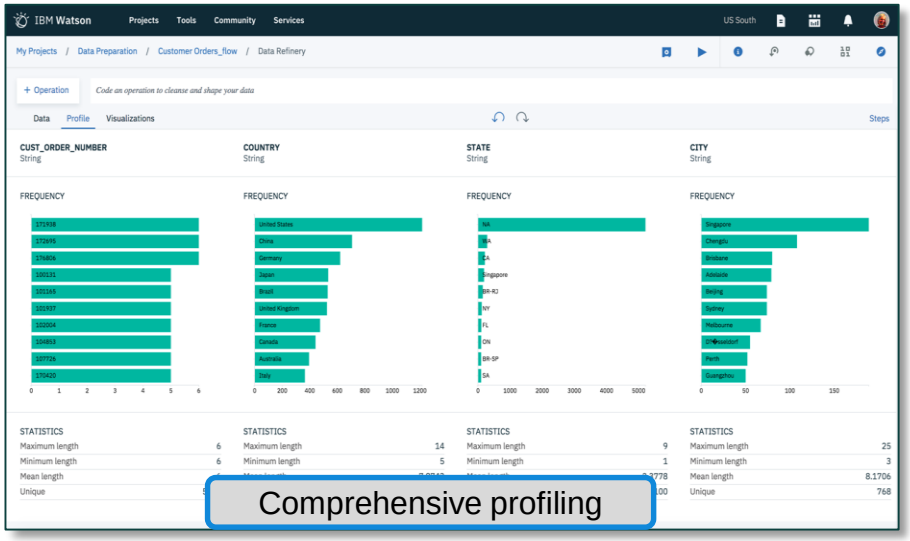
Create ML flows using Neural Network Modeler

Watson Studio - Data Refinery

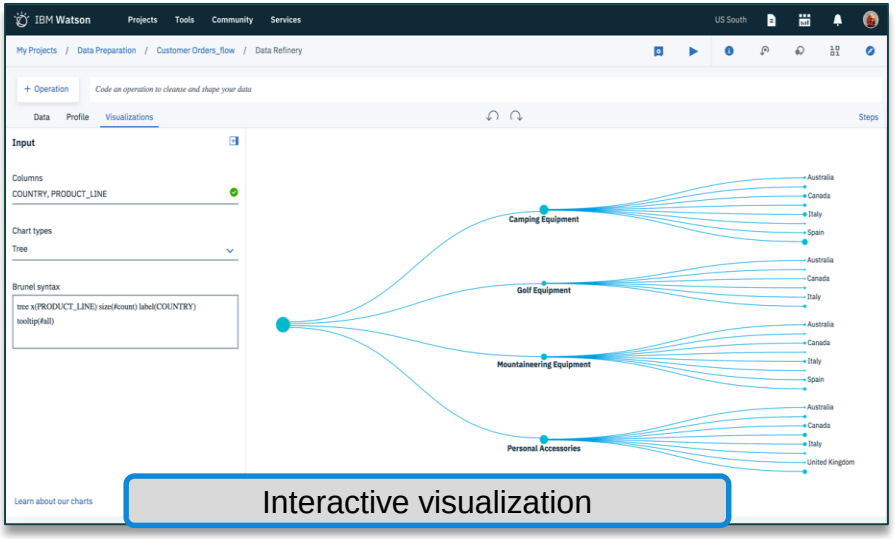
Making Data fit for use



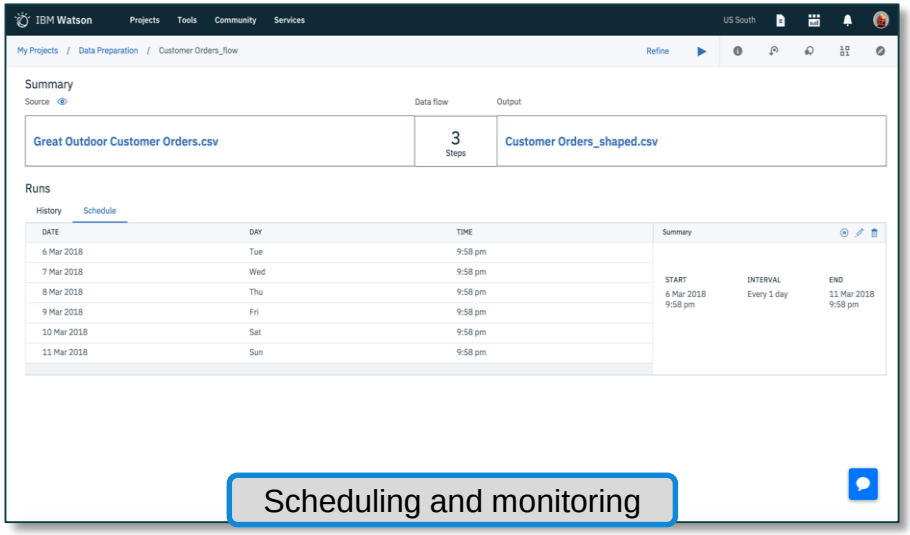
Self-service data refinement and cleaning



Comprehensive profiling



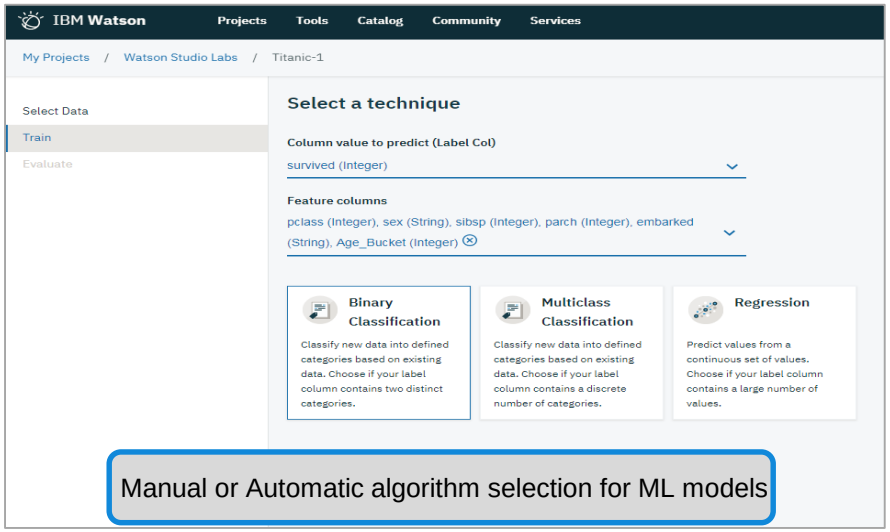
Interactive visualization



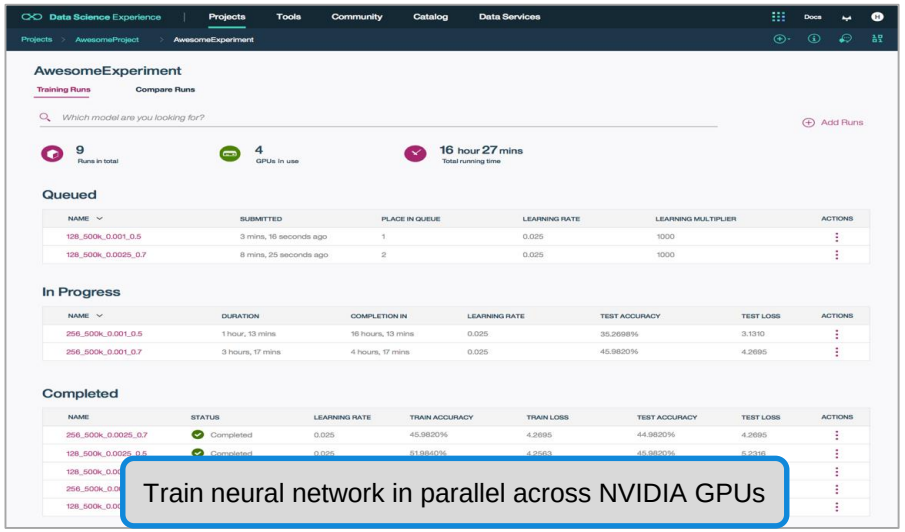
Scheduling and monitoring

Watson Studio – Watson Machine Learning

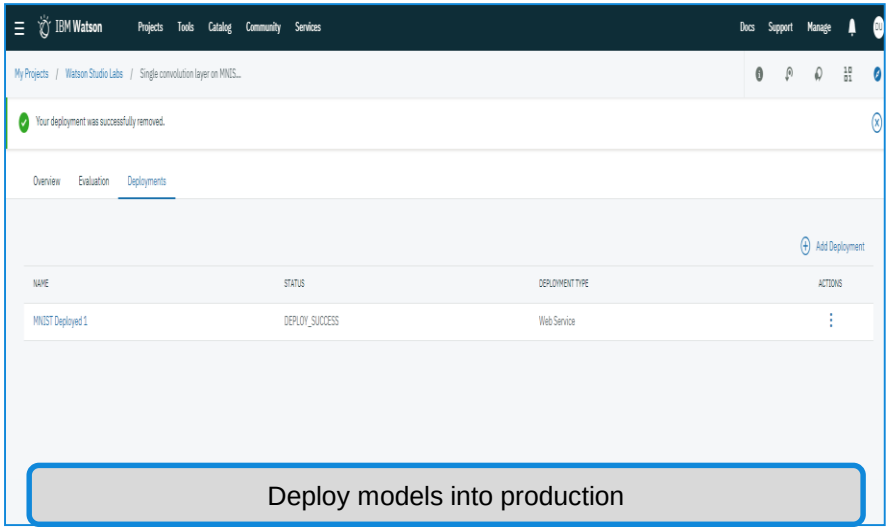
Simplifying deployment and management of ML models in production



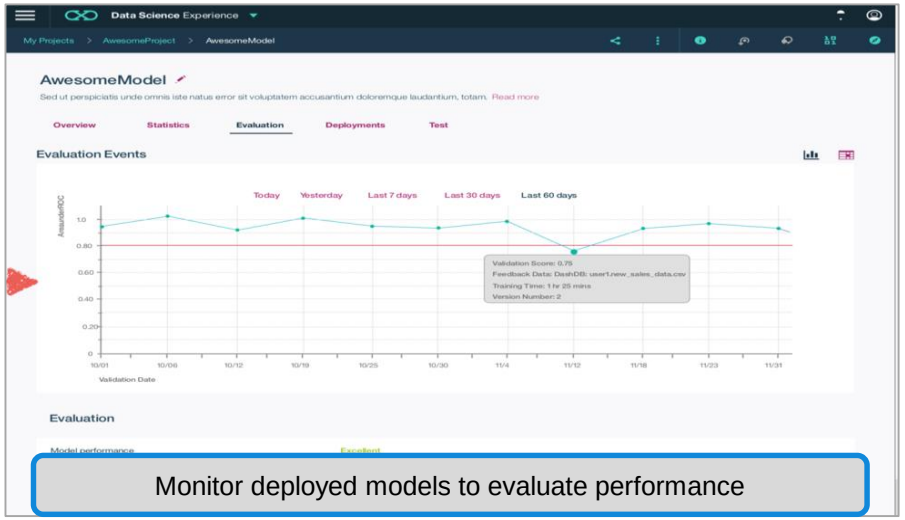
Manual or Automatic algorithm selection for ML models



Train neural network in parallel across NVIDIA GPUs



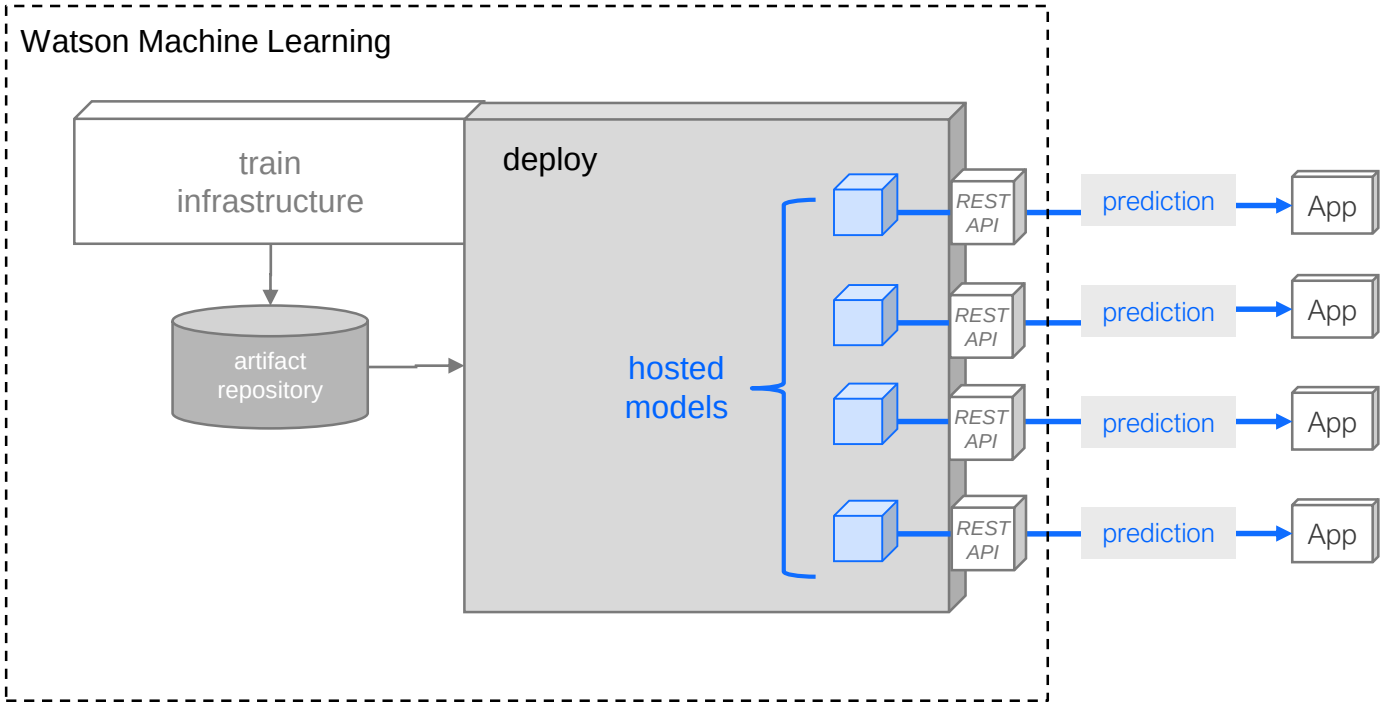
Deploy models into production



Monitor deployed models to evaluate performance

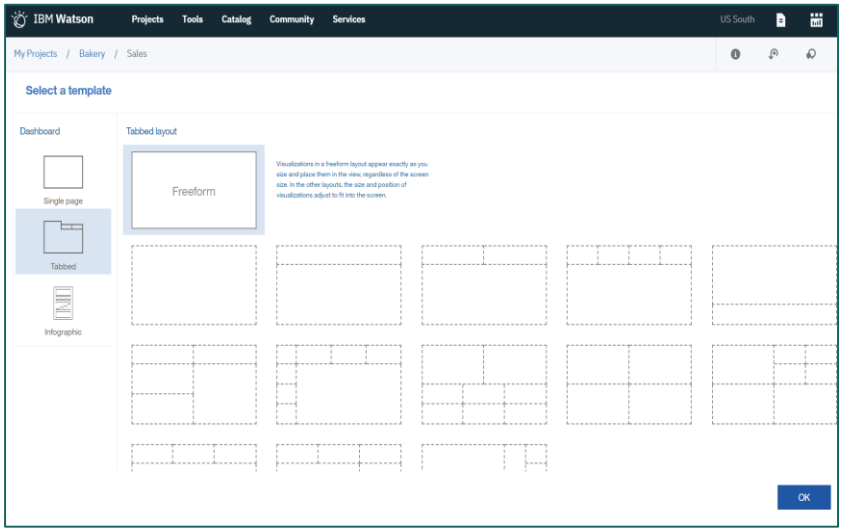
Watson Studio- Deploying Trained Models

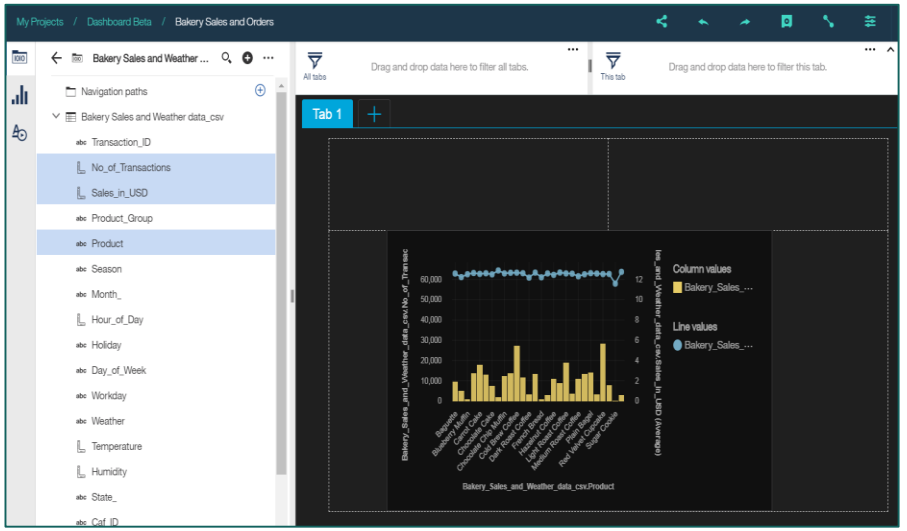
Deploy your models within Watson Machine Learning

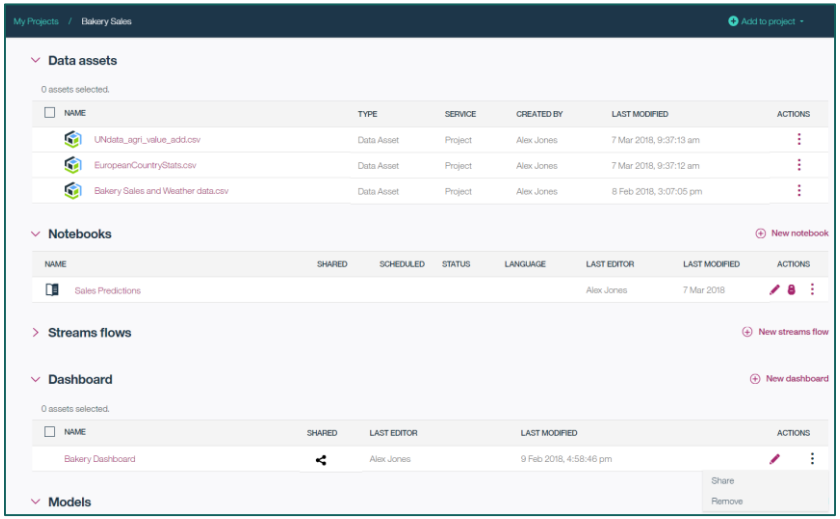


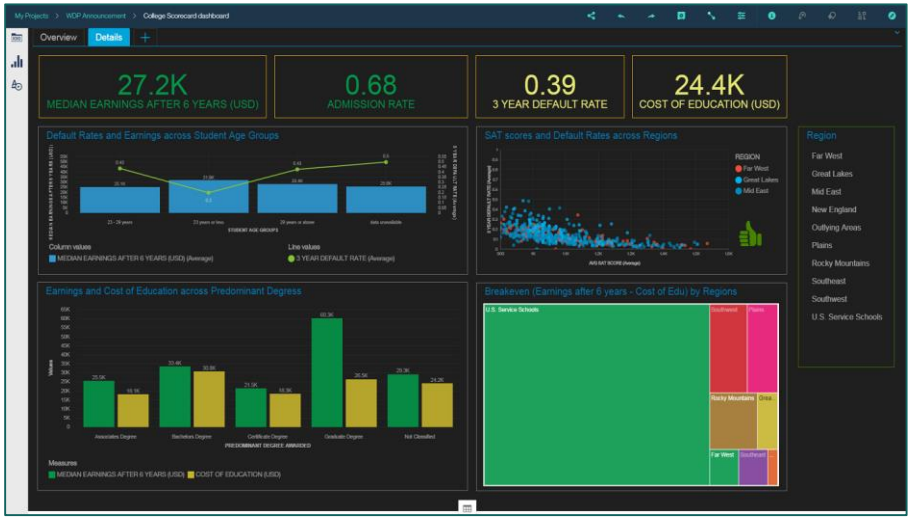
Watson Studio – Dynamic Dashboards

Making insights available to all



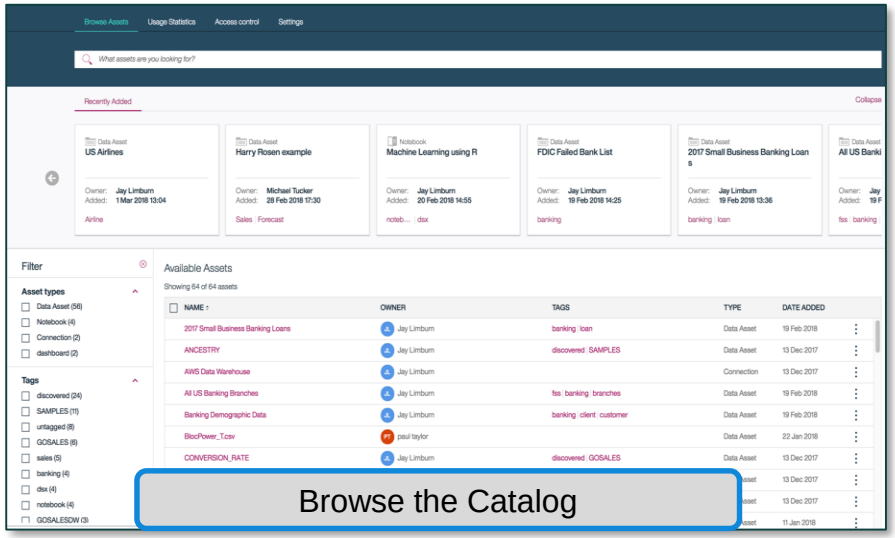




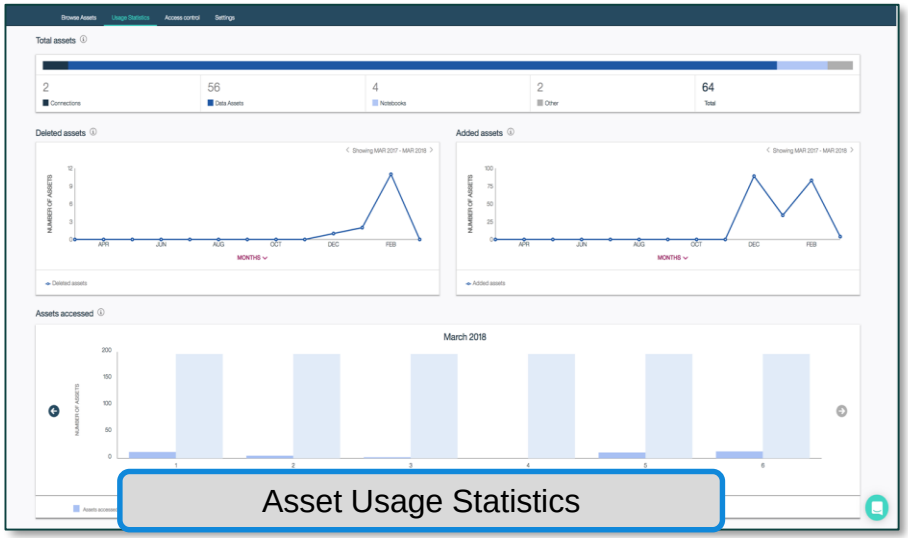


Watson Knowledge Catalog

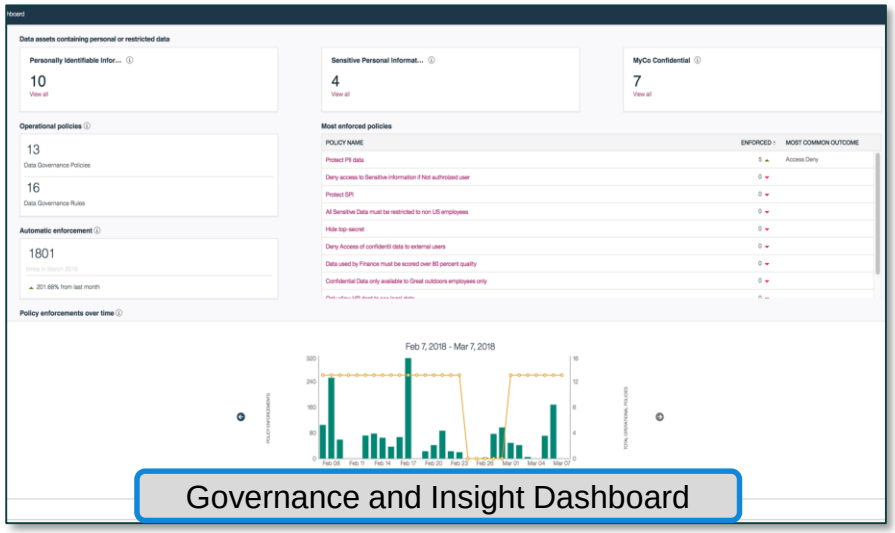
Unlock tribal knowledge and unleash knowledge workers



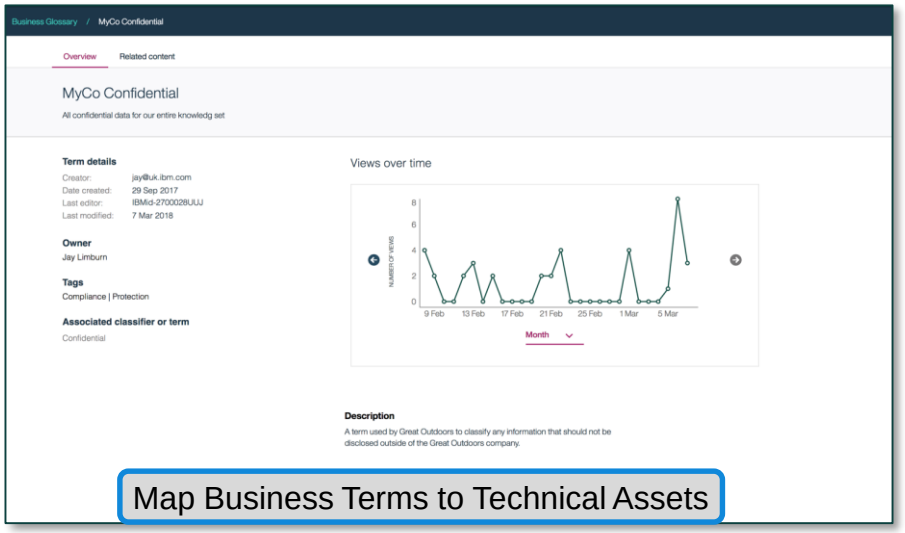
Browse the Catalog



Asset Usage Statistics



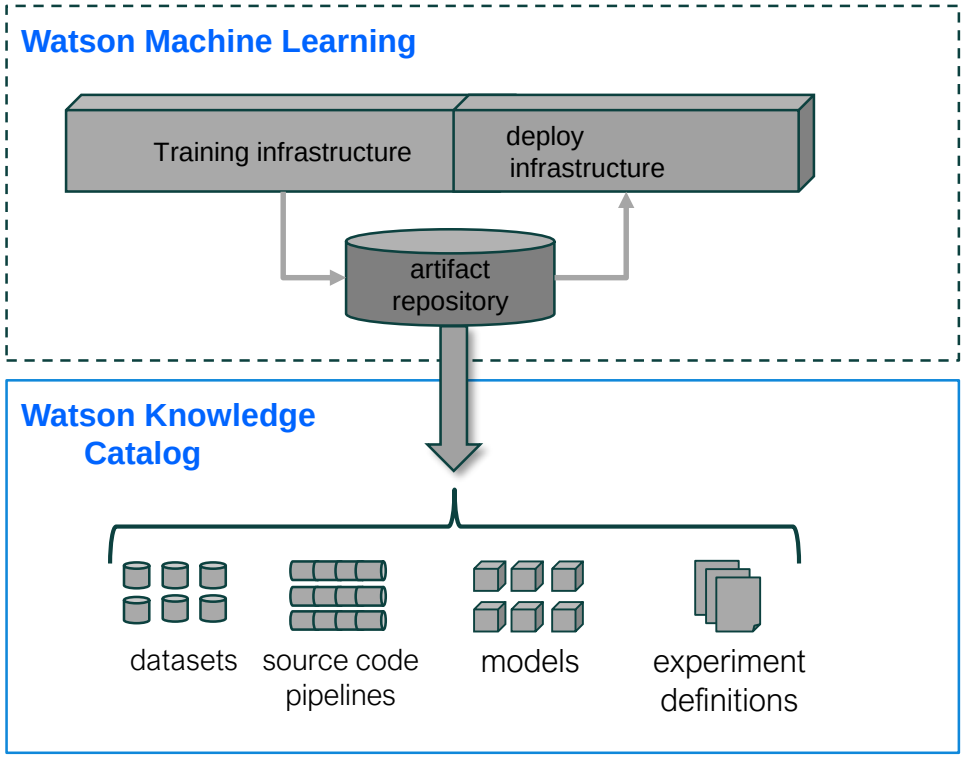
Governance and Insight Dashboard



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



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

88

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 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

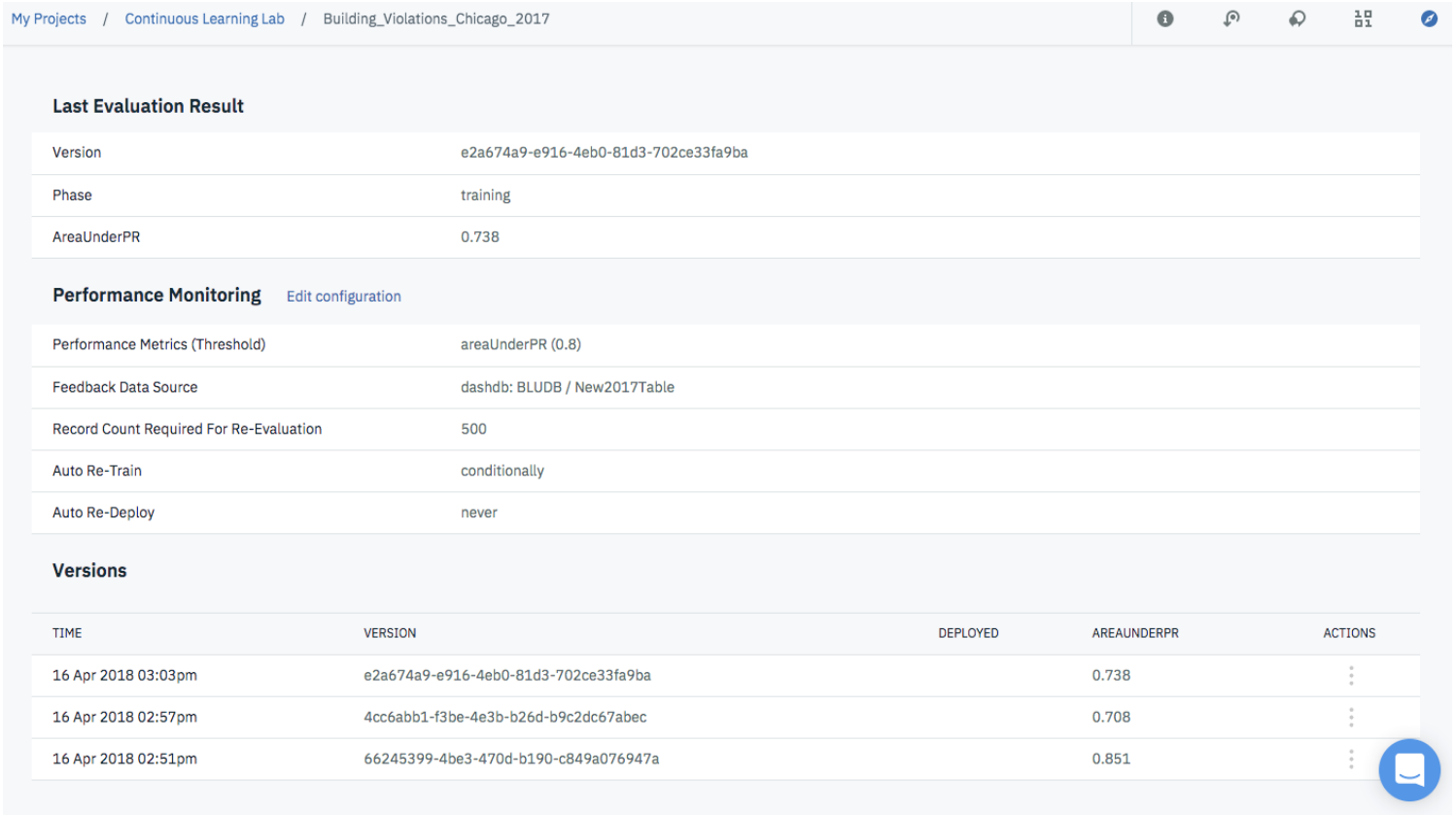
Using 2017 Chicago building data, we will make Chicago a safer place by building a model to predict when buildings are likely to fail inspection. We can then use our model to find which buildings are most dangerous and attend to those first.

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.



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

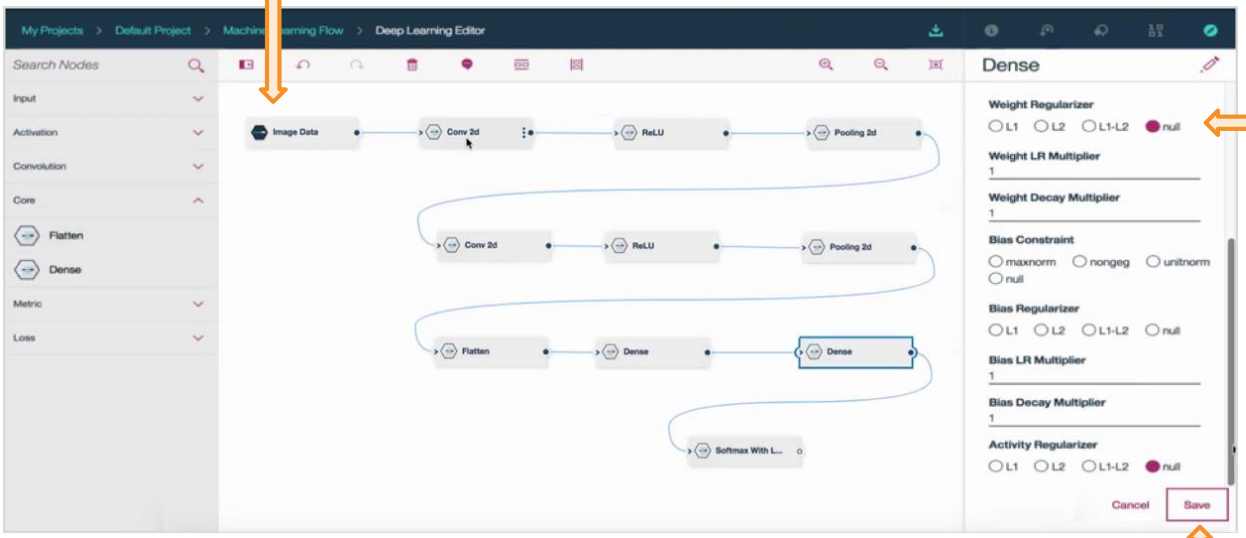
Lab-3 Neural Network Modeler

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.

Supported Frameworks

TensorFlow™ PYTORCH Caffe Keras

Real-time validation of network flow



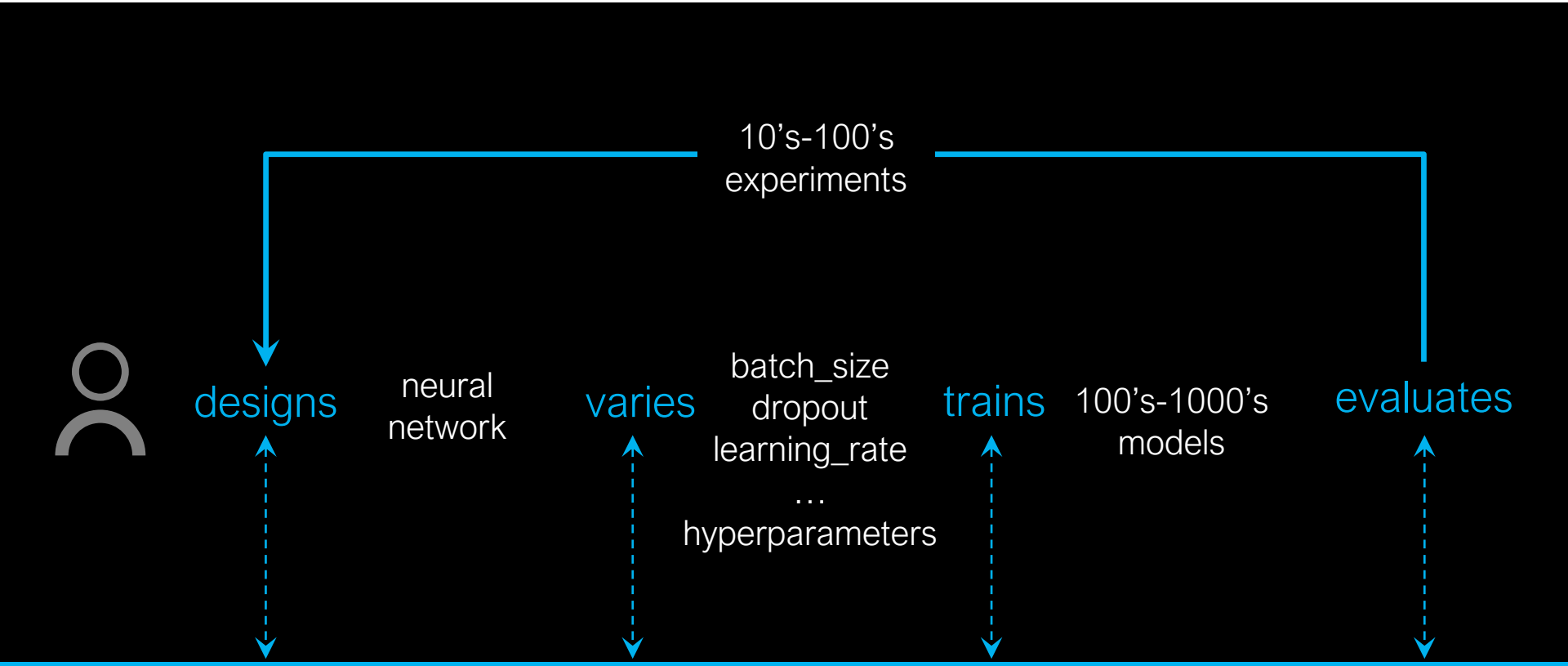
Drag-and-drop network layers

- Define layer configuration
- Choose optimizer params

- 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.