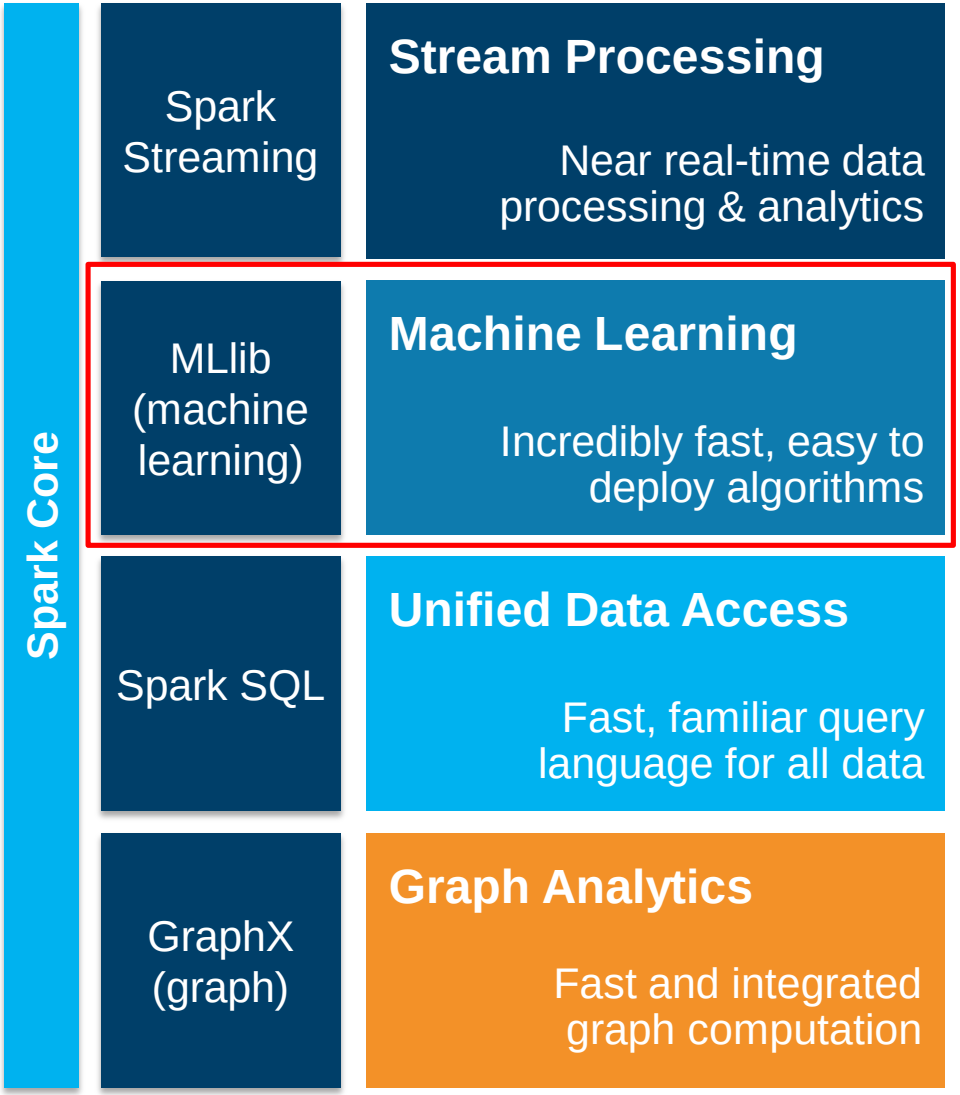
# Spark Machine Learning Binary Classification with the Titanic Dataset


**Joel Patterson**
**Bernie Beekman**
**Davin Shearer**

© 2016 IBM Corporation

# Spark Capabilities




- **Micro-batch event processing** for near real-time analytics
- Process live streams of data (IoT, Twitter, Kafka)
- No multi-threading or parallel processing required

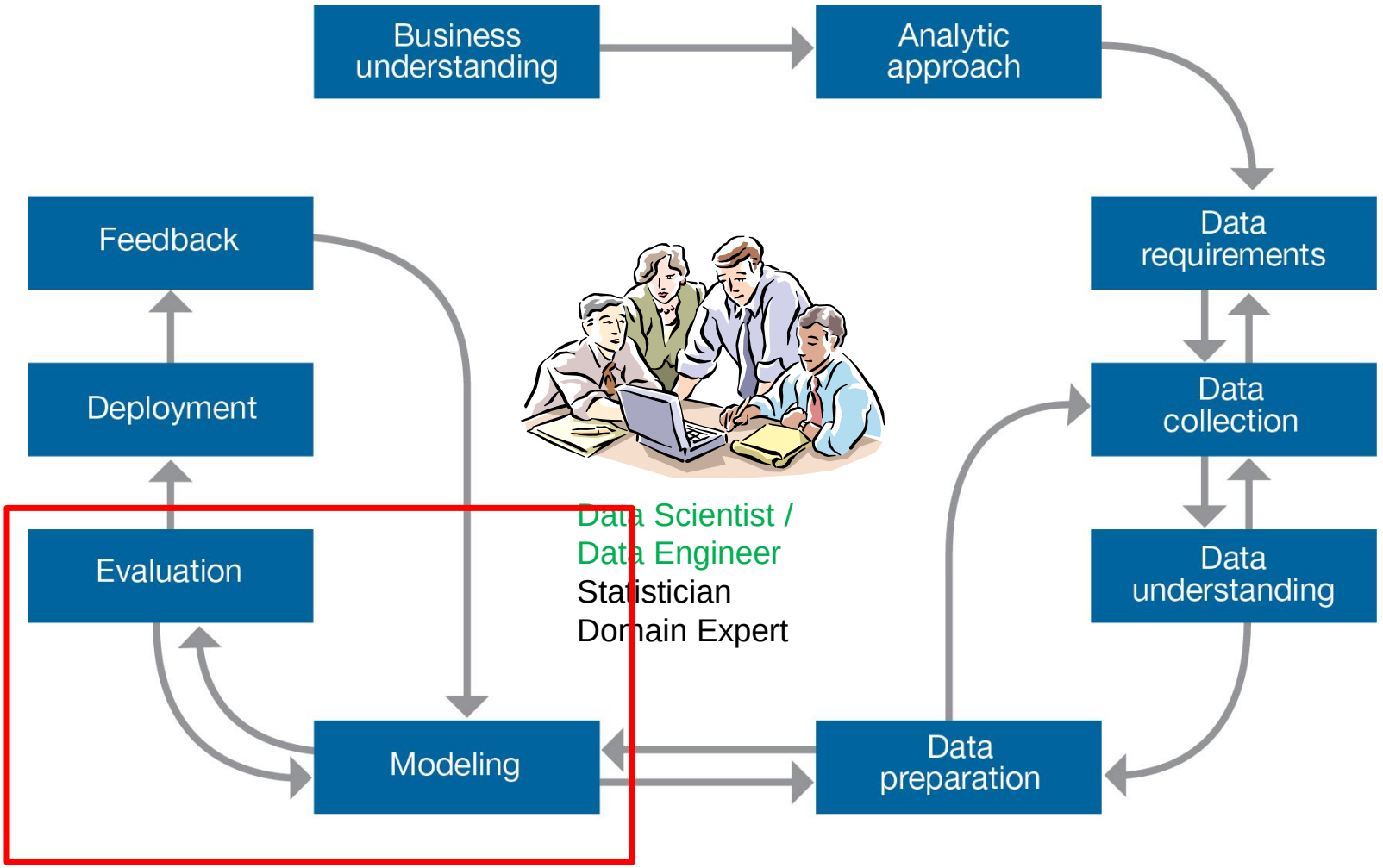
- **Predictive and prescriptive analytics**, and smart application design, from statistical and algorithmic models
- Algorithms are pre-built

- **Query your structured data sets** with SQL or other dataframe APIs
- Data mining, BI, and insight discovery
- Get results faster due to performance

- **Represent data in a graph**
- Represent/analyze systems represented by nodes and interconnections between them
- Transportation, person to person relationships, etc.


© 2016 IBM Corporation

# Data Science Methodology





3
© 2016 IBM Corporation

# Machine Learning – A more formal definition

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

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


© 2016 IBM Corporation

# 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.

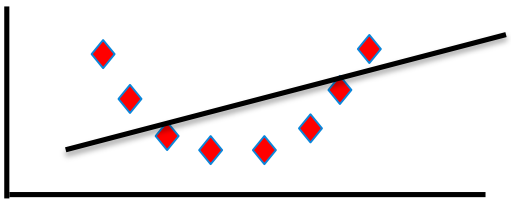
- **Examples:**
  - Learning how to play a music instrument
  - Learning how to play a sport (golf, tennis, etc…)
  - Practicing for a math exams doing exercises
  - A teacher or coach will measure performance to evaluate progress
  - Practice makes perfect

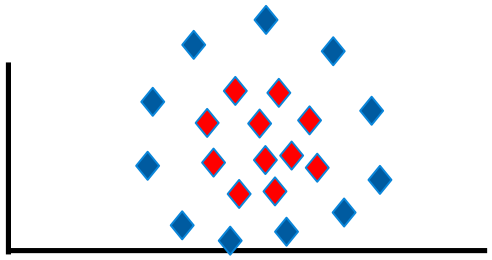

© 2016 IBM Corporation

# 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

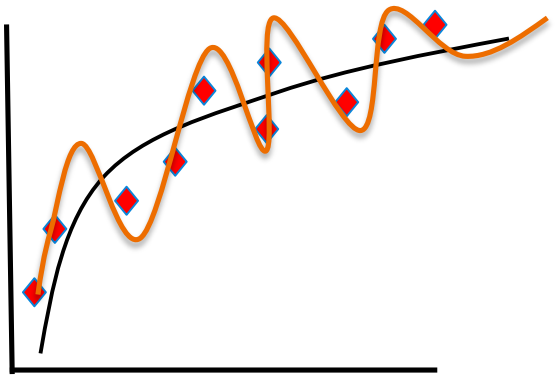





© 2016 IBM Corporation

# Learning challenges

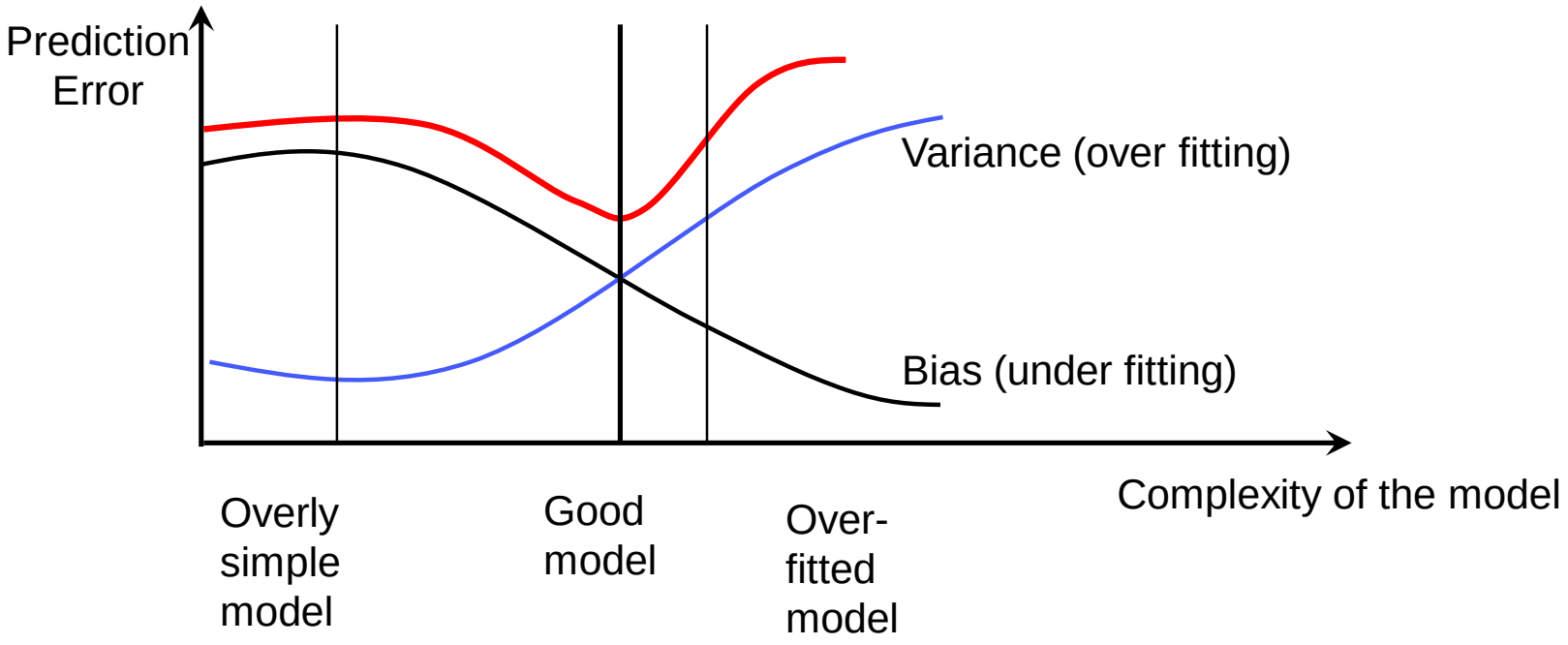
- **Over fitting:**
  - Hyper-sensitivity to minor fluctuations, ending up in modeling a lot of the unwanted noise in the data:
  - This can lead to over fitting in Machine Learning.




© 2016 IBM Corporation

# Learning challenges

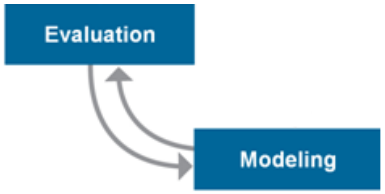
- **Compromise between bias and variance:**

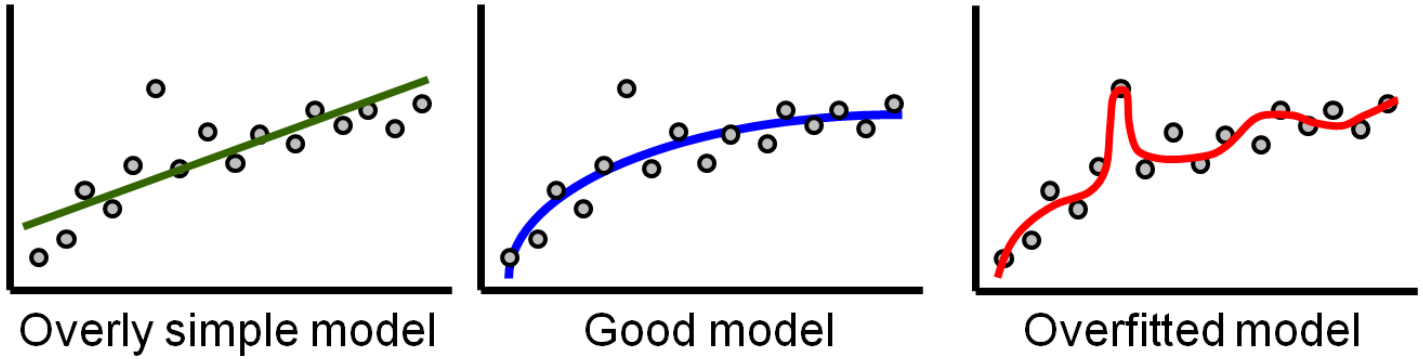


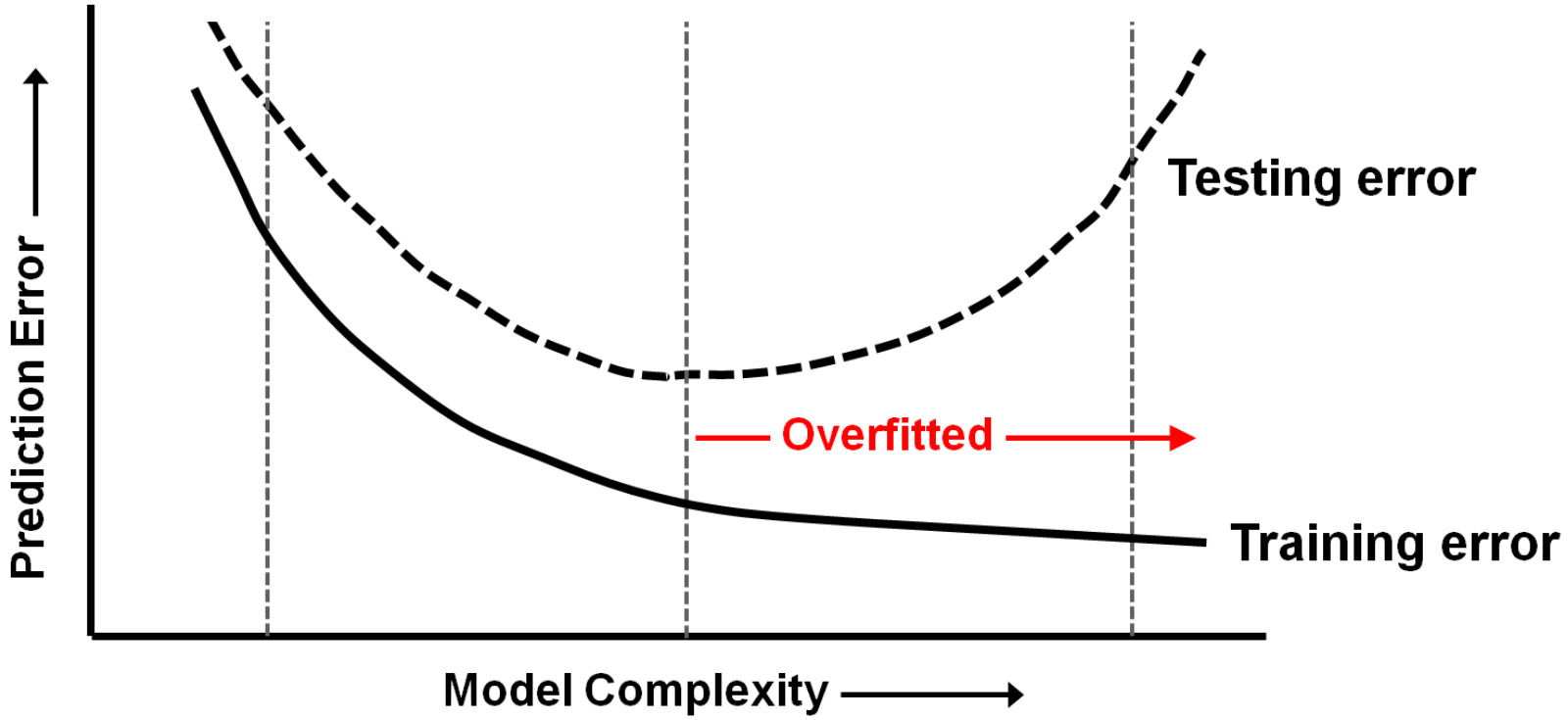


© 2016 IBM Corporation

# When to stop training a model





© 2016 IBM Corporation

# Learning challenges

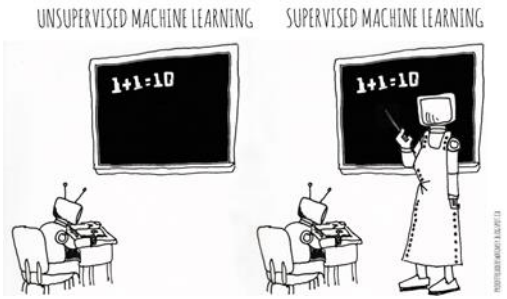
- **Diminishing returns:**
  - People can:
    - Have more or less talent
    - get bored or enthusiastic
  - Machines will not, however:
  - Making progress initially is usually more easy, but improving gets harder as we move along. We may need to try different learning methods, styles to keep going: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


© 2016 IBM Corporation

# 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

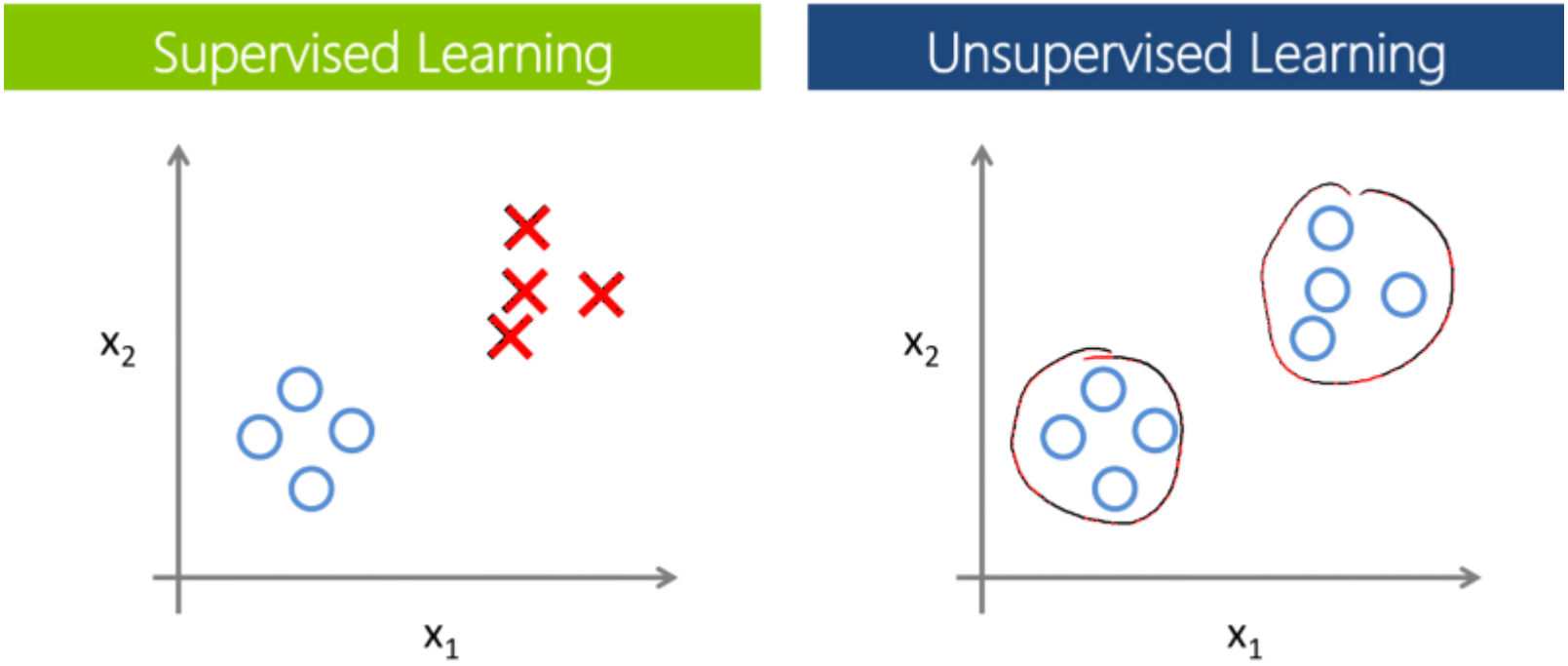
- **Unsupervised learning**
  - No labels are given to the learning algorithm, leaving it on its own to find structure (patterns and relationships) in its input
  - Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning)





© 2016 IBM Corporation

# Supervised vs. Unsupervised Learning





© 2016 IBM Corporation

# Categories of Machine Learning

| | Discrete Output | Continuous Output |
|---|---|---|
| Supervised Learning (require Ground-Truth) | • **Classification** (outcome is discrete)<br>  • Binary Classification<br>  • Detecting Fraud<br>  • Predicting defaults on loans<br>  • Discovering spam<br>  • Predicting users who might churn<br>  • Multi class Classification<br>  • Classifying images, sounds<br>  • Assigning categories to news articles, webpages, etc.… | • **Regression**<br>  - Predicting the price of a house<br>  - Predicting loss amounts for loans |
| Unsupervised Learning (no Ground-Truth data required) | • Clustering<br>  - Grouping discrete elements<br>• Frequent Patterns and associations<br>  - People who buy chips also buy beer | • Clustering<br>  - Grouping continuous variables<br>• Dimensionality Reduction<br>  - PCA<br>  - SVD |


© 2016 IBM Corporation

# Categories of Machine Learning

| | Discrete Output | Continuous Output |
|---|---|---|
| Supervised Learning (require Ground-Truth) | • **Classification** (outcome is discrete)<br>• Binary Classification<br>• Linear Models (Logistic Regression)<br>• Decision Trees<br>• Naïve Bayes<br>• Multi class Classification<br>• Decision Trees<br>• Naïve Bayes<br>• K-NN | • **Regression**<br>- Linear<br>- Ridge<br>- Lasso<br>• **Decision Trees**<br>• Random Forest<br>• Gradient Boosted Trees |
| Unsupervised Learning (no Ground-Truth data required) | • Clustering<br>- k-means<br>• FP-Growth | • Clustering<br>- k-means<br>- Gaussian Mixture<br>• Dimensionality Reduction<br>- PCA<br>- SVD |


© 2016 IBM Corporation

# Spark ML Pipeline Terminology

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

- **DataFrame**: Spark ML uses DataFrame from Spark SQL as an ML dataset, which can hold a variety of data types
- **Transformer**: A Transformer is an algorithm which can transform one DataFrame into another DataFrame
- **Estimator**: An Estimator is an algorithm which can be fit on a DataFrame to produce a Transformer
- **Pipeline**: A Pipeline chains multiple Transformers and Estimators together to specify an ML workflow
- **Parameter**: All Transformers and Estimators share a common API for specifying parameters


© 2016 IBM Corporation

# Pipelines – How they work

- **A Pipeline is specified as a sequence of stages where each stage is either a Transformer or an Estimator**

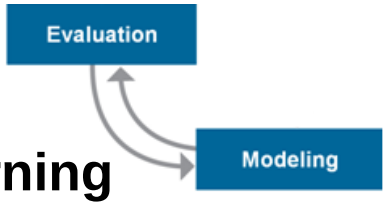
- **These stages are run in order and the input DataFrame is transformed as it passes through each stage**
  - For Transformer stages, the transform() method is called on the DataFrame
  - For Estimator stages, the fit() method is called to produce a Transformer (which becomes part of the fitted Pipeline), and that Transformer's transform() method is called on the DataFrame

- **For example, a simple text document processing workflow might include several stages:**
  - Split each document's text into words
  - Convert each document's words into a numerical feature vector
  - Learn a prediction model using the feature vectors and labels



© 2016 IBM Corporation

# 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**)
  - Testing set ➔ tune the parameters & variables during model building (**iterative**)
    - Assess model quality during training process
    - Avoid overfitting the model to the training set
  - Validation set ➔ estimate accuracy or error rate of model (**once**)
    - Assess model's expected performance when applied to new data


© 2016 IBM Corporation

# Spark ML

- Spark ML is Spark's machine learning (ML) library
- Its goal is to make practical machine learning scalable and easy
- Consists of common learning algorithms and utilities, including
  - Classification
  - Regression
  - Clustering
  - Collaborative filtering
  - Dimensionality Reduction
- Lower-level optimization primitives
- Higher-level pipeline APIs


© 2016 IBM Corporation

# Spark ML

- Divides into two packages:
  - spark.mllib contains the original API built on top of RDDs
  - spark.ml provides higher-level API built on top of DataFrames for constructing ML pipelines

- Using spark.ml is recommended because with DataFrames the API is more versatile and flexible
  - spark.mllib will continue to be supported



© 2016 IBM Corporation

# Demo Data - Titanic

- **The binary classification demo will utilize the famous Titanic dataset**
  **https://www.kaggle.com/c/titanic/data**

- **The Titanic data set was chosen as it contains text based and numeric features that are both continuous and categorical, giving us the opportunity to explore and utilize a number of feature transformers available in Spark ML**


© 2016 IBM Corporation

# Demo Data - Titanic

## Variable Descriptions:

```
survival        Survival
                (0 = No; 1 = Yes)
pclass          Passenger Class
                (1 = 1st; 2 = 2nd; 3 = 3rd)
name            Name
sex             Sex
age             Age
sibsp           Number of Siblings/Spouses Aboard
parch           Number of Parents/Children Aboard
ticket          Ticket Number
fare            Passenger Fare
cabin           Cabin
embarked        Port of Embarkation
                (C = Cherbourg; Q = Queenstown; S = Southampton)
```


| PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22 | 1 | 0 | A/5 21171 | 7.25 | | S |
| 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Thayer) | female | 38 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26 | 0 | 0 | STON/O2. 3101282 | 7.925 | | S |
| 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35 | 1 | 0 | 113803 | 53.1 | C123 | S |
| 5 | 0 | 3 | Allen, Mr. William Henry | male | 35 | 0 | 0 | 373450 | 8.05 | | S |
| 6 | 0 | 3 | Moran, Mr. James | male | | 0 | 0 | 330877 | 8.4583 | | Q |
| 7 | 0 | 1 | McCarthy, Mr. Timothy J | male | 54 | 0 | 0 | 17463 | 51.8625 | E46 | S |
| 8 | 0 | 3 | Palsson, Master. Gosta Leonard | male | 2 | 3 | 1 | 349909 | 21.075 | | S |
| 9 | 1 | 3 | Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) | female | 27 | 0 | 2 | 347742 | 11.1333 | | S |
| 10 | 1 | 2 | Nasser, Mrs. Nicholas (Adele Achem) | female | 14 | 1 | 0 | 237736 | 30.0708 | | C |


© 2016 IBM Corporation

# Demo Flow

- **Read in Titanic dataset as a DataFrame**
  - Drop unwanted columns and rows with null or invalid data
  - Label the data (“Survived”)

- **Feature Engineering**
  - StringIndexer (Sex and Embarked variables)
  - Bucketizer (Age and Fare variables)
  - VectorAssembler
  - Normalizer

- **Create a Pipeline**

- **Split Ratings data into Training (90%) and Test (10%) datasets**
  - Cache the resulting DataFrames

- **Fit the Pipeline to the Test data set**
  - Logistic Regression


© 2016 IBM Corporation

# Demo Flow (continued)


- **Evaluate the resulting predictions**
  - Area under the ROC curve
- **Tune the model (hyperparamaters)**
  - Build Parameter Grid
  - Cross-evaluate to find the best model
- **Make improved predictions using the cross-validated model**
- **Make prediction on an imaginary passenger**
- **Show how to easily reuse completed work using a different machine learning algorithm**
  - Random Forest


© 2016 IBM Corporation

# Logistic regression: Defining "odds"



- **Defining the "odds":**
  - **Given an event with probability p. Say p = 0.8**
  - **The probability that the event does NOT take place is: 1 – p = 0.2**
  - **The odds of the event is defined to be: $\frac{p}{1-p} = 4$**
  - **Also noted 4 : 1 (4 to 1)**
- **The odds can be interpreted as:**
  - **Out of 5 occurrences, the event will happen 4 times and will not happen once.**
  - **There are 4 times more chances for the event to happen than not.**


© 2016 IBM Corporation

# Logistic regression



- **Example:**
  - Consider that you are a first time home buyer. Your credit score is **720** (values can range from **320** to **850**). You know that credit scores are taken into account for mortgage approval and you would like to figure out the probability of being approved based on the credit score. (real scenario uses many more parameters).
  - You find online a list of credit scores and whether the applicant was approved or not. The list would look like this:

| Credit Score | Approved |
|---|---|
| 425 | 0 |
| 550 | 1 |
| 550 | 0 |
| 630 | 1 |
| 690 | 0 |
| Etc… | Etc… |


© 2016 IBM Corporation

# Logistic regression



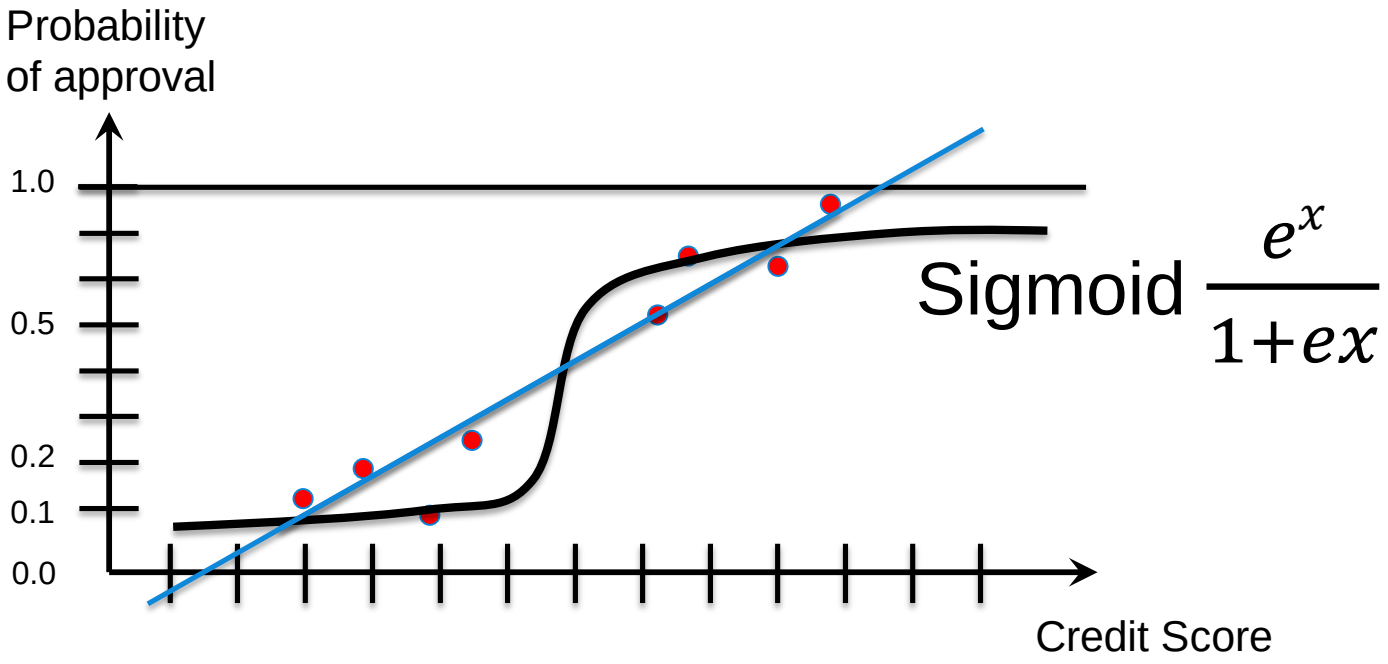
- **For each credit score we can add up all the number of occurrences for approval and rejection and derive the probability of being approved.**
- **Example: Suppose that in our data set, credit score 630 had 12 applications approved and 8 rejected.**
- **We could then calculate for credit score 630:**
  - Probability(approval) = 12 / (12 + 8) = 0.6
  - Probability(rejection) = 8 / (12 + 8) = 0.4
- **The same could be repeated for all credit scores and we could then plot a graph showing the credit scores, versus the probability of being approved. (see next slide).**


© 2016 IBM Corporation

# Logistic regression



- **Graph:**





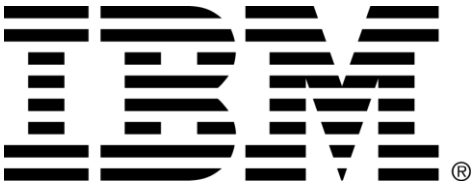

© 2016 IBM Corporation



© 2016 IBM Corporation

# Backup


© 2016 IBM Corporation




© 2016 IBM Corporation