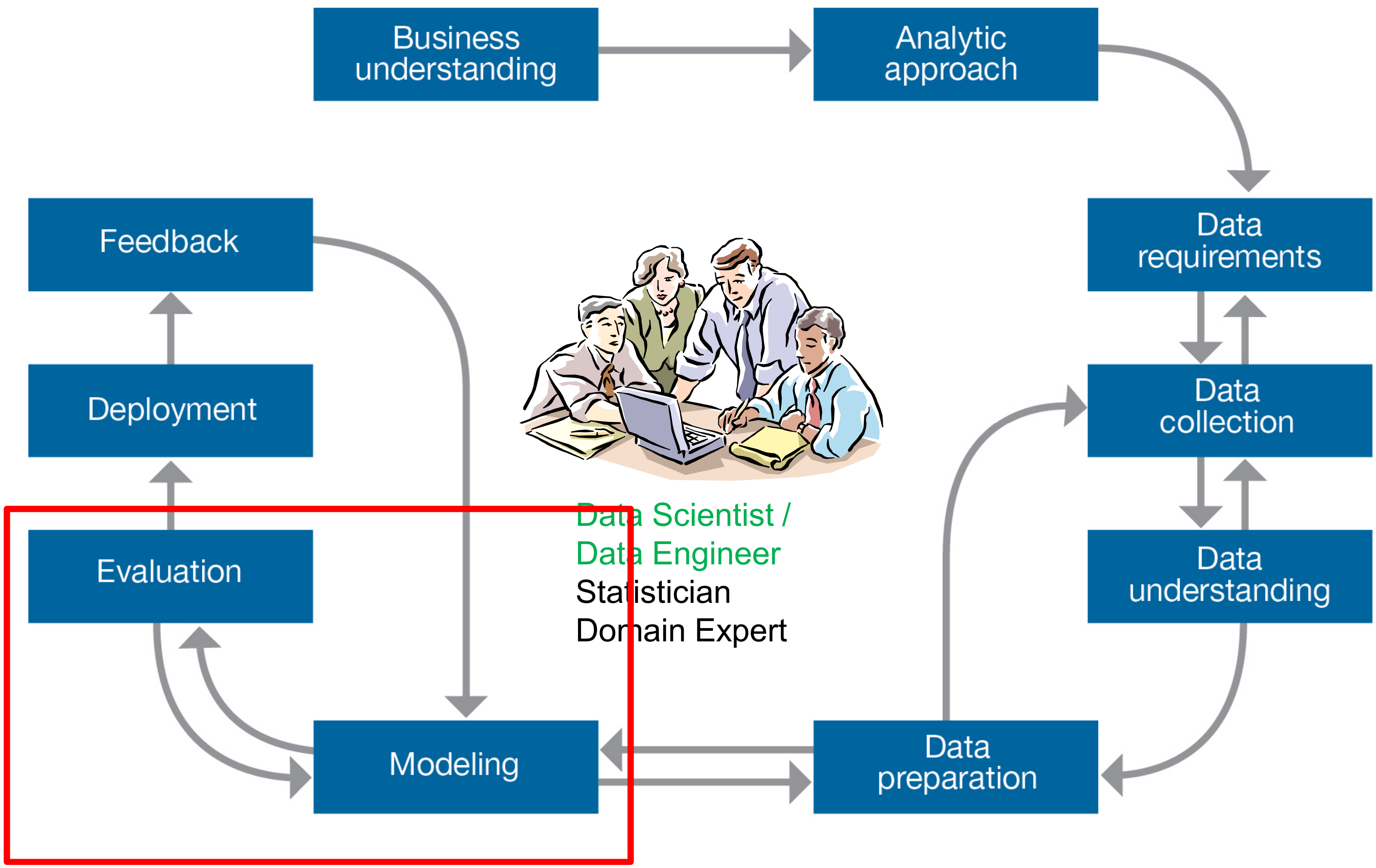
Lab 3 – Machine Learning

Spark Capabilities

Spark Core	Spark Streaming	**Stream Processing** Near real-time data processing & analytics	• **Micro-batch event processing** for near real-time analytics • Process live streams of data (IoT, Twitter, Kafka) • No multi-threading or parallel processing required
	MLlib (machine learning)	**Machine Learning** Incredibly fast, easy to deploy algorithms	• **Predictive and prescriptive analytics**, and smart application design, from statistical and algorithmic models • Algorithms are pre-built
	Spark SQL	**Unified Data Access** Fast, familiar query language for all data	• **Query your structured data sets** with SQL or other dataframe APIs • Data mining, BI, and insight discovery • Get results faster due to performance
	GraphX (graph)	**Graph Analytics** Fast and integrated graph computation	• **Represent data in a graph** • Represent/analyze systems represented by nodes and interconnections between them • Transportation, person to person relationships, etc.

Data Science Methodology

(John B. Rollins – rollins@us.ibm.com)

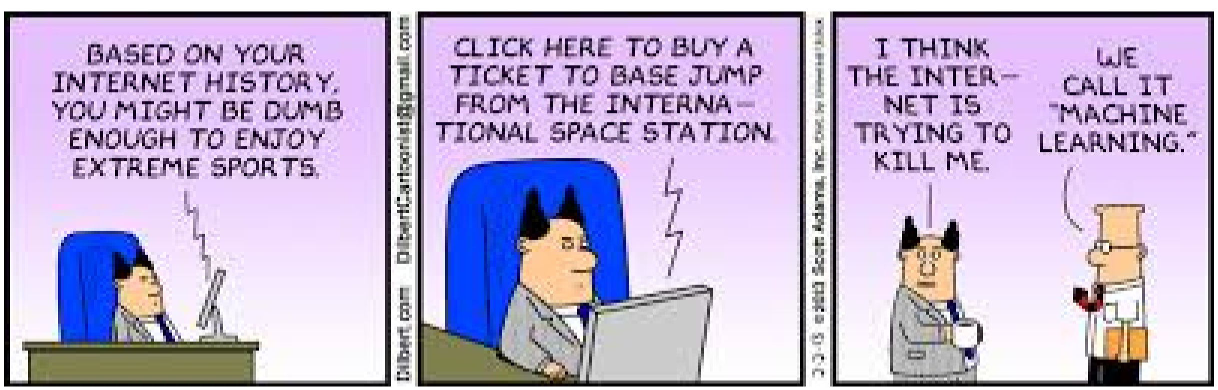


Machine Learning

- In 1959, Arthur Samuel defined machine learning as a "Field of study that gives computers the ability to learn without being explicitly programmed"

- Machine learning automates the development of analytical models that can learn and make predictions on data

- Machine learning allows computers to find hidden insights without being explicitly programmed where to look



Machine Learning – A more formal definition

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

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

Machine Learning vs Human Learning

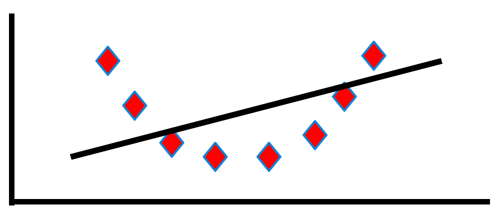
- **In many aspects, ML not fundamentally different from HL:**
  – Repeat the same task over and over again to gain experience.
  – Action of repeating the same task is referred to as “practice”
  – With practice and experience, we get better at learned tasks.

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

Learning challenges

- **Under fitting:**
  - Not knowing enough “basic” concepts, i.e. not being well-equipped enough to tackle learning at hand:
    - You can’t study calculus without knowing some algebra.
    - You can’t learn playing hockey without knowing how to skate.
    - You can’t learn polo without knowing how to ride.

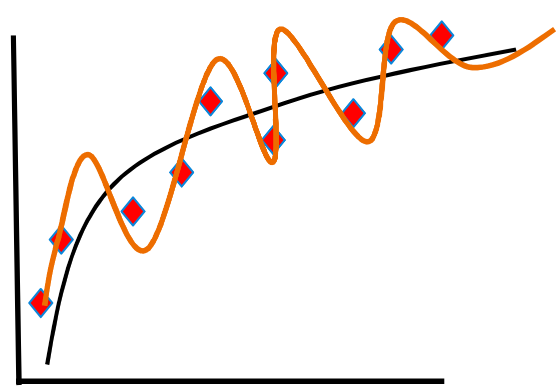
  - This can lead to under fitting in Machine Learning: The chosen model is just not “sophisticated”, “rich”, enough to capture the concept.



Learning challenges

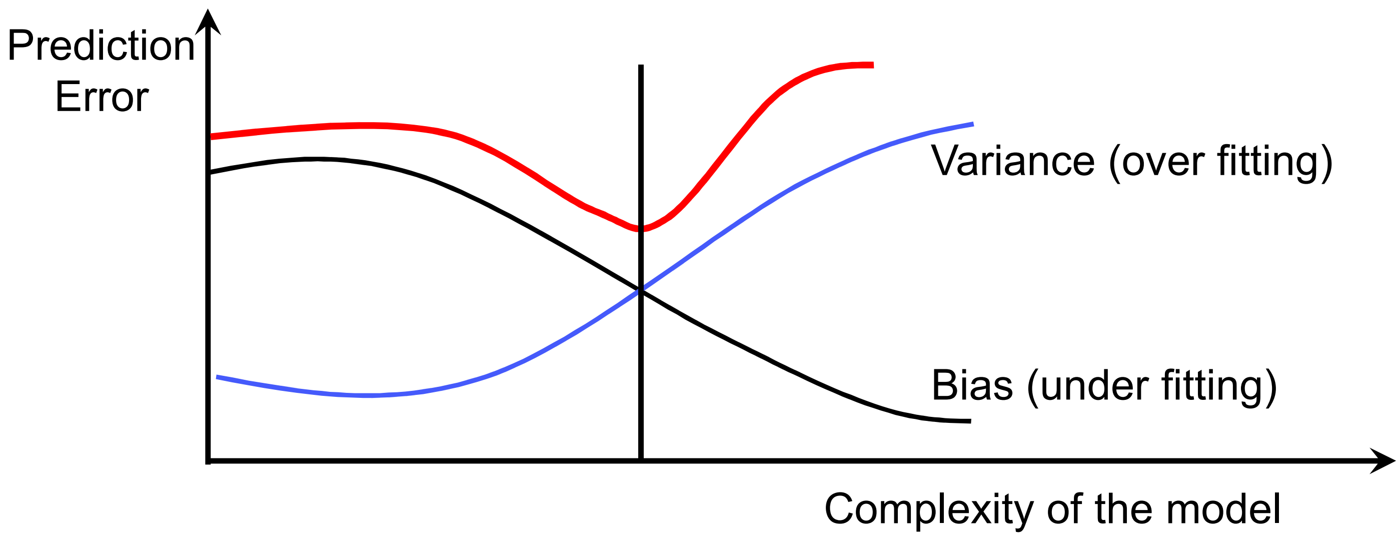
- **Over fitting:**
  – Hyper-sensitivity to minor fluctuations, ending up in modeling a lot of the unwanted noise in the data:

  – This can lead to over fitting in Machine Learning.



Learning challenges

- **Compromise between bias and variance:**



Learning challenges

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

Machine Learning Examples

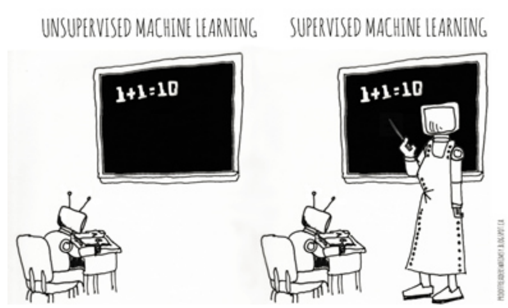
- Is this cancer ? (Medical diagnosis)
- Is this legitimate or fraud (spam) ?
- What is the market value of this house ?
- Which of these people are good friends with each other ?
- Will this engine fail (when) ?
- Will this person like this movie ?
- Who is this ?
- What did you say ? (Speech recognition)

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

Categories of Machine Learning

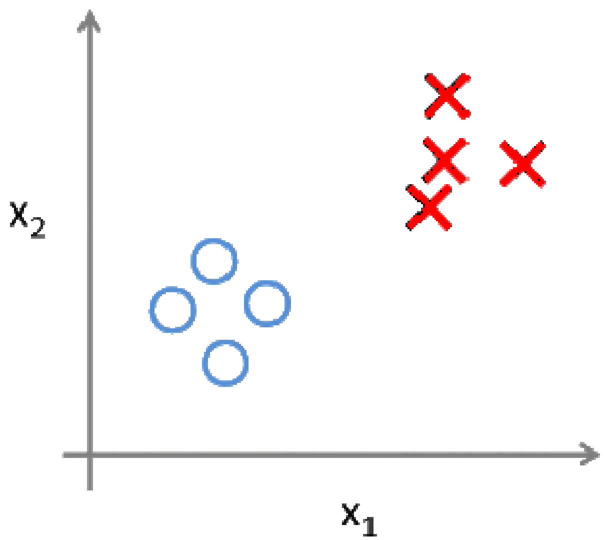
- **Supervised learning**
  – The program is “trained” on a pre-defined set of “training examples”, which then facilitate its ability to reach an accurate conclusion when given new data
  – The algorithm is presented with example inputs and their desired outputs (correct results)
  – The goal is to learn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

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

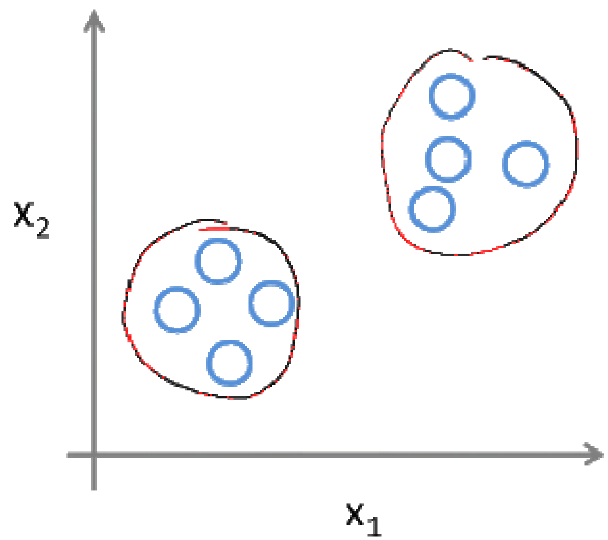


Supervised vs. Unsupervised Learning

Supervised Learning

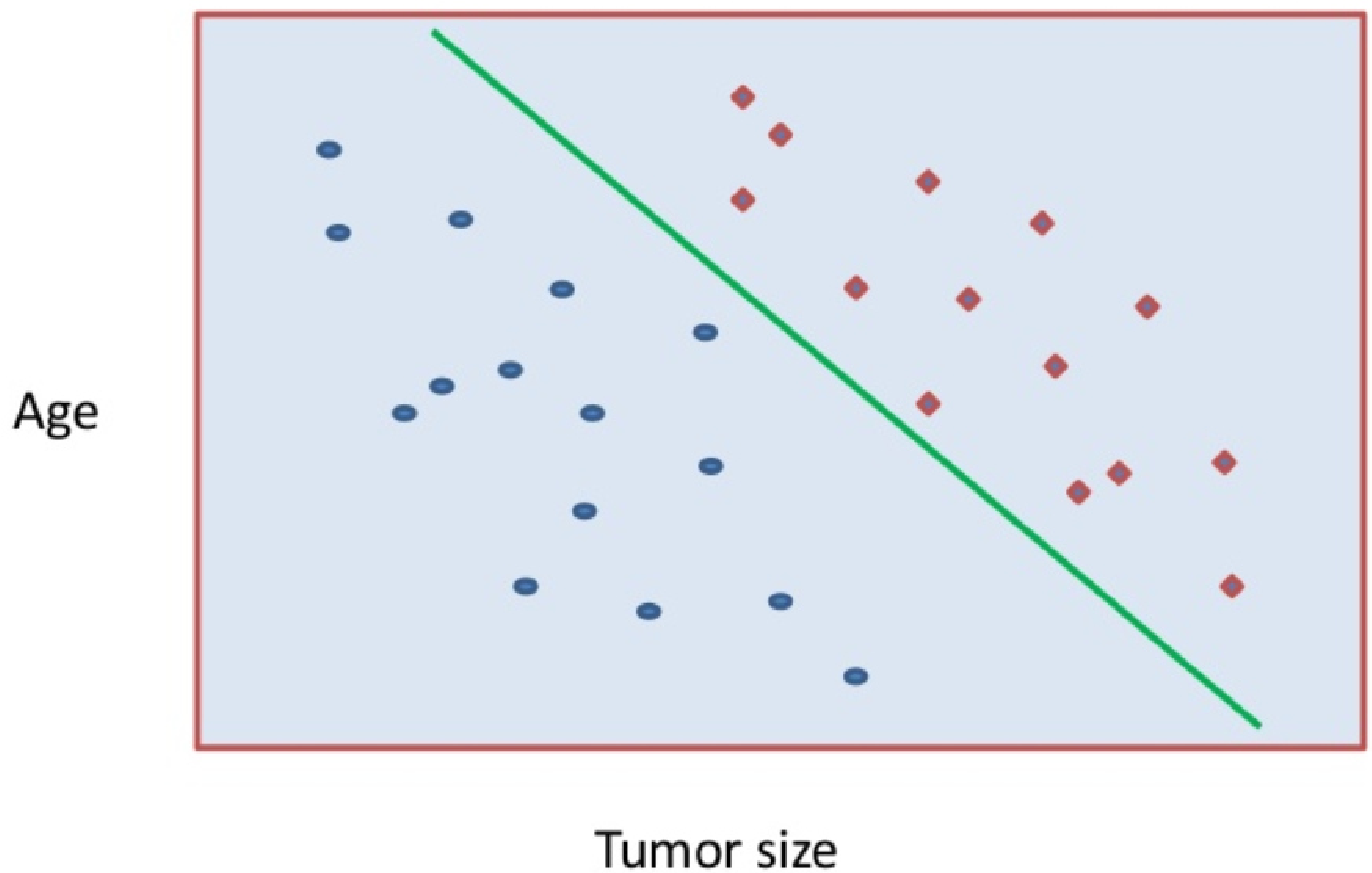


Unsupervised Learning



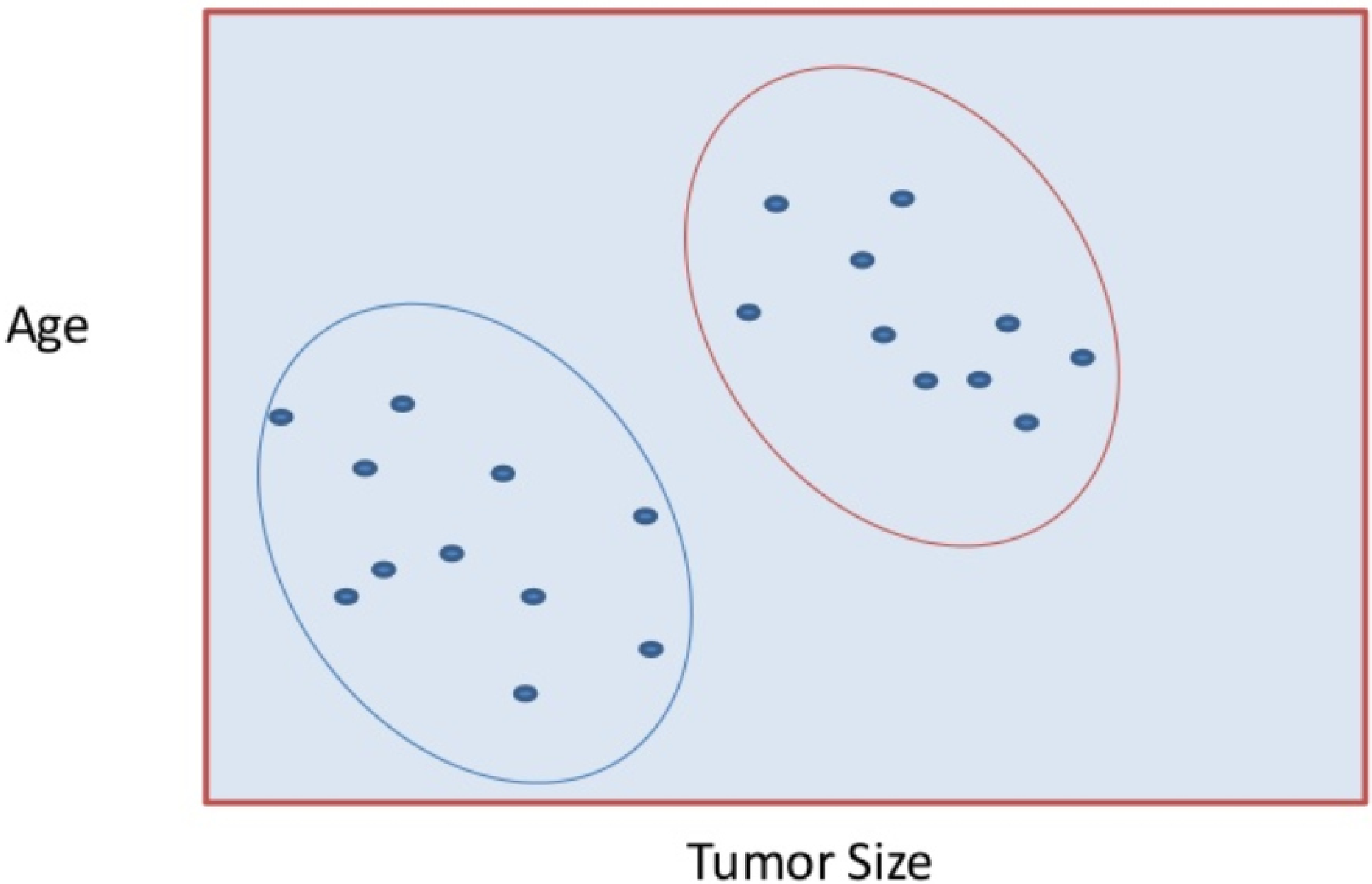
Example of Supervised Learning (Classification)

Goal is to make predictions



Example of Unsupervised Learning (Clustering)

Goal is to understand the structure of the data, not make predictions



Categories of Machine Learning

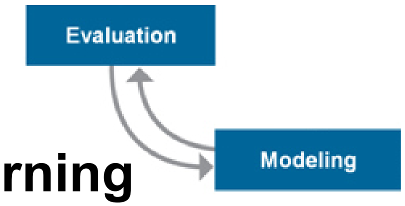
	Discrete Output	Continuous Output
Supervised Learning (require Ground-Truth)	• **Classification** (outcome is discrete) • Binary Classification • Detecting Fraud • Predicting defaults on loans • Discovering spam • Predicting users who might churn • Multi class Classification • Classifying images, sounds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

Categories of Machine Learning

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

Recommendation Engines
- Content Filtering
- Collaborative Filtering

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

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

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

Recommendation Systems

- **Recommendation systems seek to predict the rating (or preference) that a user would give to an item**

- **Recommendation systems attempt to improve customer experience through personalized recommendations based on prior user feedback**

- **Recommender systems have become extremely common in recent years, and are applied in a variety of applications**
  - movies, music, news, books, research articles, search queries, social tags, …
  - products in general

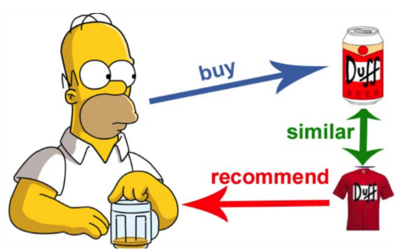
- **Collaborative filtering is a technique that is commonly used for recommender systems**
  - employs a form of wisdom of the crowd approach

Collaborative Filtering with Spark ML

- **Forms of Collaborative Filtering**
  - Explicit matrix factorization - preferences provided by users themselves are utilized
  - Implicit matrix factorization - only implicit feedback (e.g. views, clicks, purchases, likes, shares etc.) is utilized

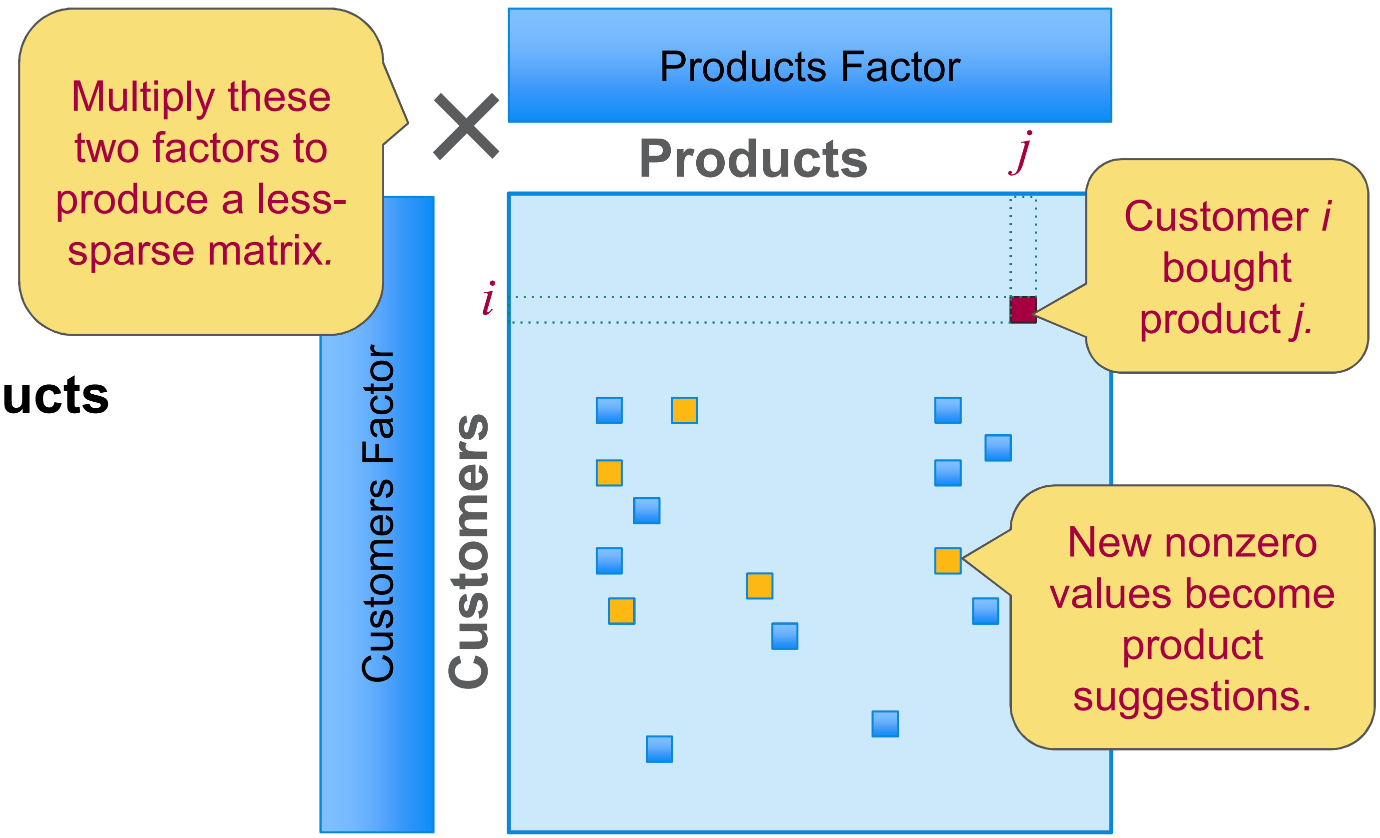
- **Spark ML supports an implementation of matrix factorization for collaborative filtering**
  - Matrix factorization models have consistently shown to perform extremely well for collaborative filtering

- **Collaborative filtering aims to fill in the missing entries of a user-item association matrix in which users and items are described by a small set of latent factors that can be used to predict missing entries**



Running Example:
Alternating Least Squares

- **Problem: Recommend products to customers**



Lab 3 Flow

1. Download compressed CSV data and load into an RDD

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/10 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/10 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/10 8:26	2.75	17850	United Kingdom

2. Prepare the data

– Remove header
– Only keep rows that have
  - a purchase quantity greater than 0
  - a non blank customer ID
  - a non blank stock code after removing non-numeric characters

4. Create a DataFrame from the resulting RDD

– Add a label column

5. Split the dataset

– 80% for training
– 10% for testing
– 10% for cross validation

Lab 3 Flow (continued)

5. **Build a recommendation model using the training dataset**
   - Two models using different hyperparameters
     - rank
     - maxIter

6. **Test the two models using the cross validation dataset**

7. **Evaluate the two models using mean squared error**
   - Confirm “best” model against the test dataset

8. **Use the “best” model to make predictions for a particular user**
   - Top 5 recommendations

	description
0	YELLOW FLOWERS FELT HANDBAG KIT
1	MIDNIGHT BLUE COPPER FLOWER NECKLAC
2	TEA TIME TEA TOWELS
3	BLACK DROP CRYSTAL NECKLACE
4	COPPER/OLIVE GREEN FLOWER NECKLACE