# Unit 1: Introduction and Motivation

## Videos:

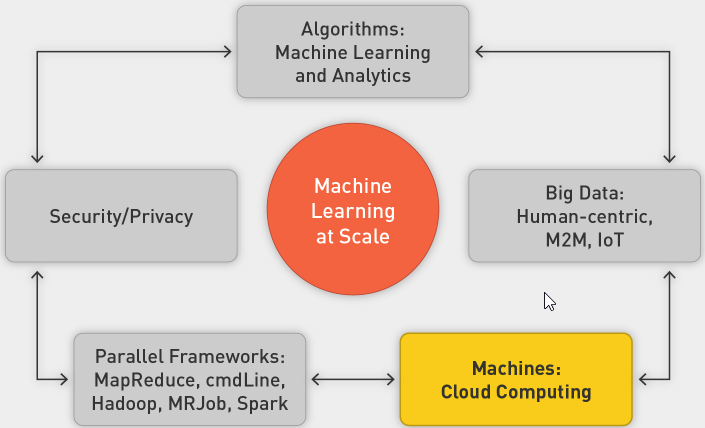
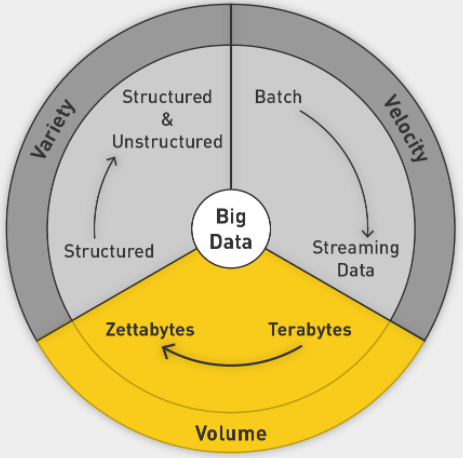
Introduction

1. Intro
2. Big Data
3. Modern-day DS
4. Seven steps in data modelling pipeline
5. Large scale machine learning
6. Tricks of the trade
7. MR using the command line

Readings

1. Doing Data Science Chapter 1 and 2: on Safari
2. An introduction to information retrieval: Chapter 13, downloaded; Info on Naïve Bayes and Text Classification
3. Elements of Statistical Learning, Chapter 7: downloaded and read

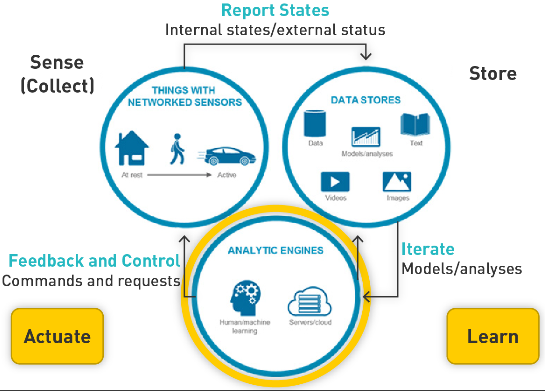
Big Data Definitions

1. ML at Scale
   1. 
2. Big Data Definition
   1. Big data is broad term for data sets so large or complex that traditional data-processing applications are inadequate:
   2. Processing
   3. Laptop (8–16 GB of memory, 1 TB hard drive) overwhelmed with 4–5 GB
   4. Storage
   5. Laptop only 1 TB
   6. Throughput
   7. Three hours to read 1 TB on laptop
   8. Other Challenges
      1. Analysis
      2. Capture
      3. Data curation
      4. Search
      5. Sharing
      6. Storage
      7. Transfer
      8. Visualization
      9. Security
      10. Information privacy
   9. Three Vs
      1. 
      2. Add Veracity: uncertainty of data
3. Sources Driving Big Data
   1. Everything is recorded online
   2. User-generated world, social media
   3. IoT
   4. Scientific Computing
4. Areas of Application
   1. NYSE
   2. FB
   3. CERN
5. Data in Zettabytes (ZB)
   1. 1ZB is 10^21
   2. 2020: 40 ZB

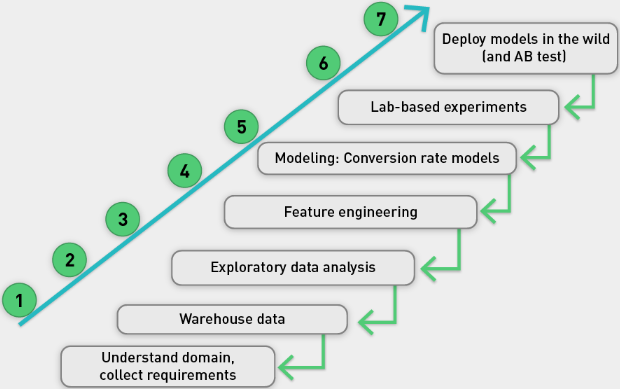
Sources of Big Data

1. Applications of Big Data
2. Societal and Personal Sources of Data
3. Online Society
4. Advertising
   1. Represents 2% of GDP in United States since 1900s
   2. Increased spending online as opposed to on traditional channels
   3. Mobile advertising surpasses others
5. Dating
6. Life Logging
7. Health Care and how tech can help
8. Genomics

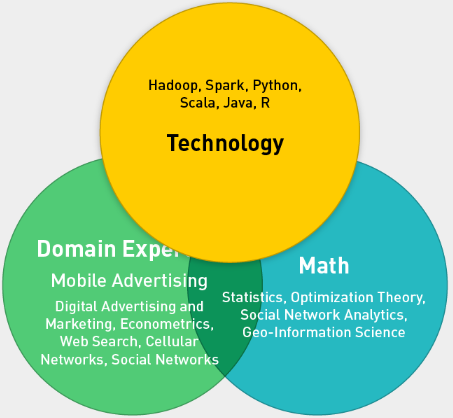
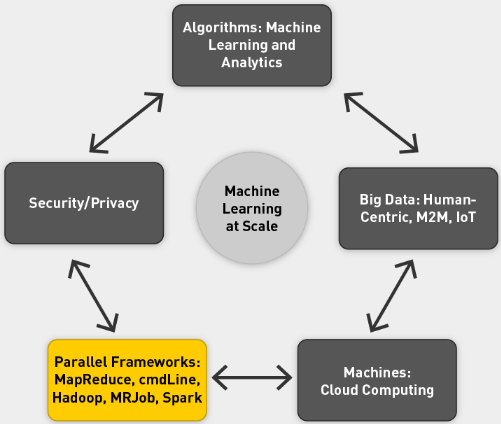
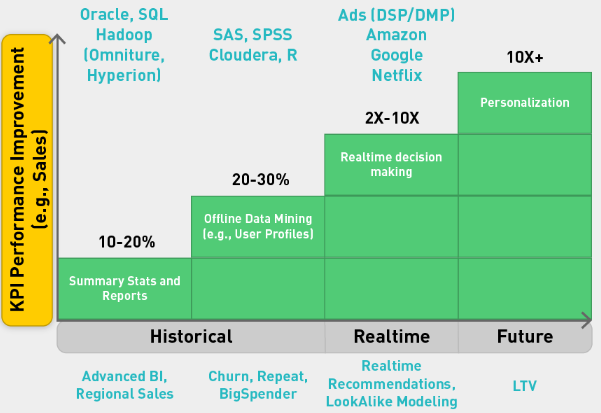
IoT

1. IoT
   1. More data coming from machines than from humans
   2. Machine to machine communication
   3. Attach sensors to all the things
   4. 2020: 26b things on network
   5. Everyone has a unique identifier and data is automatically sent across the internet
2. Japan Earthquake
   1. 14 seconds before, all trains and plants shut down
   2. Mesh of sensors from connected computers to form an early warning system
3. Tracking Nature in the Wild
   1. Sea lions with GPS tracking system
   2. Underwater hub picks up signals from devices
   3. Serengeti
4. Challenges
   1. Battery life
   2. Storage
   3. Network connectivity
   4. Privacy and security
5. Smart homes
6. Smart Cities
   1. 
7. Autonomous Vehicles

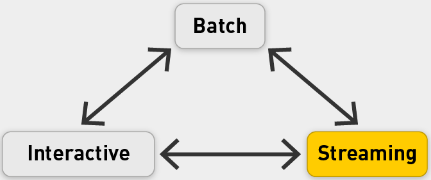
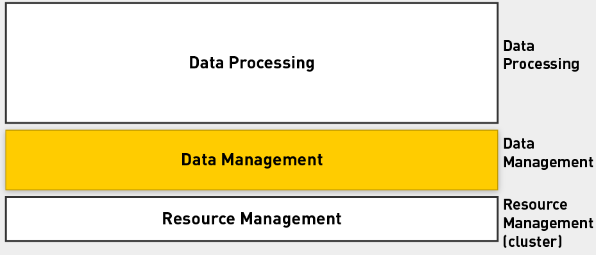
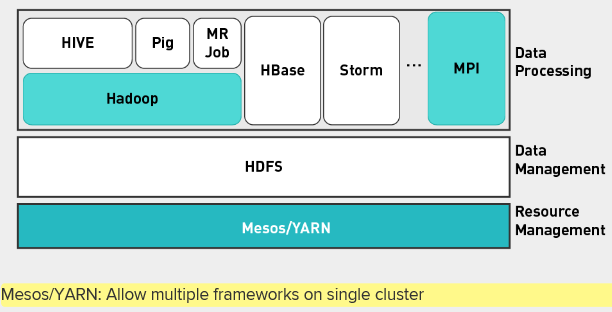
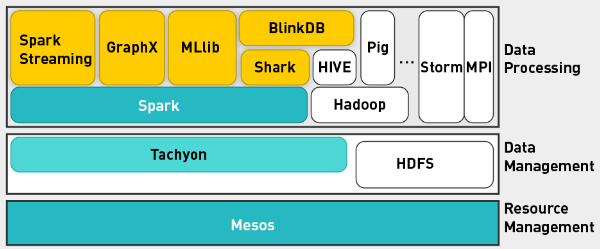
Data Modelling Pipeline

1. Intro
   1. Massive challenge for machine learning
   2. ML is not in a vacuum
   3. ML in real world
2. Modelling
   1. 
   2. Very iterative, might go back at every step
   3. 80% of work is spent in getting, prepping and engineering data
3. Example US insurance
   1. Half of it is property and casualty (auto, home, commercial)
   2. Highly competitive rates
      1. Special rates for good drivers
      2. Score computed through accident history over five years
      3. Static, annual, batch process, easy
4. Progressive’s Snapshot product
   1. Bad driver surveillance through telematics
   2. Small box plugged into steering wheel that records, sends information
   3. Features:
   4. How many miles driven
   5. How many miles driven between midnight and 4 a.m.
   6. Use of sharp or gentle braking
   7. Data used to analyze driving patterns, risk to insurance company
   8. Potential reduced premiums for user
5. Privacy
   1. Huge data generated
   2. Data not transitory
   3. Caution needed with who can access and how is it stored
   4. Who owns the data?

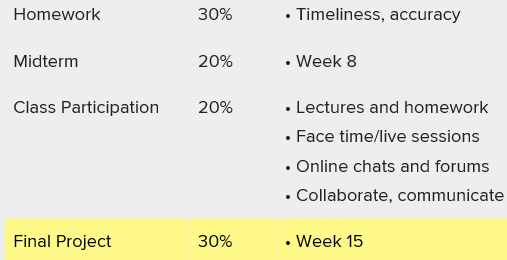
Data Scientist: Post This Class

1. Three skill sets
   1. 
2. Understand Models
3. Understand Security and Privacy
4. ML at scale
   1. 
5. DS Team with different experts in the three fields
6. DS improves KPIs dramatically
   1. 

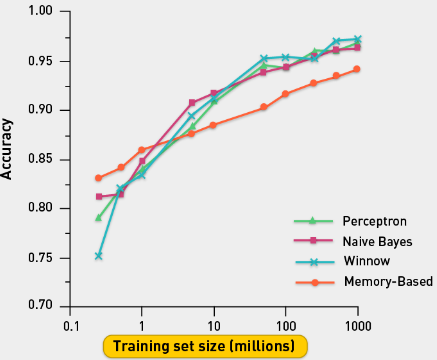
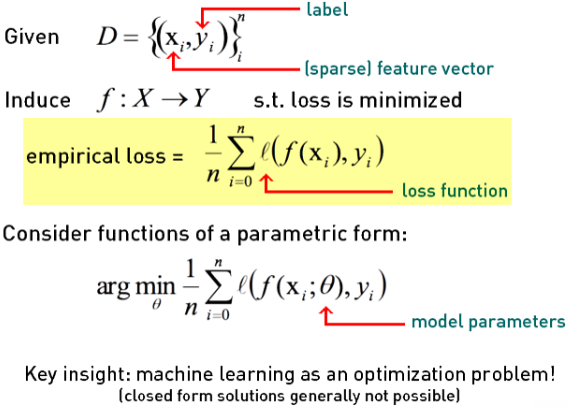
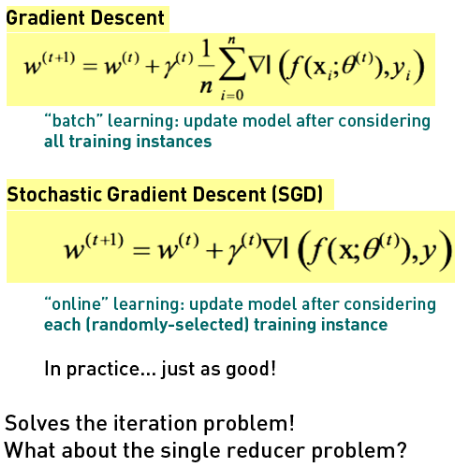
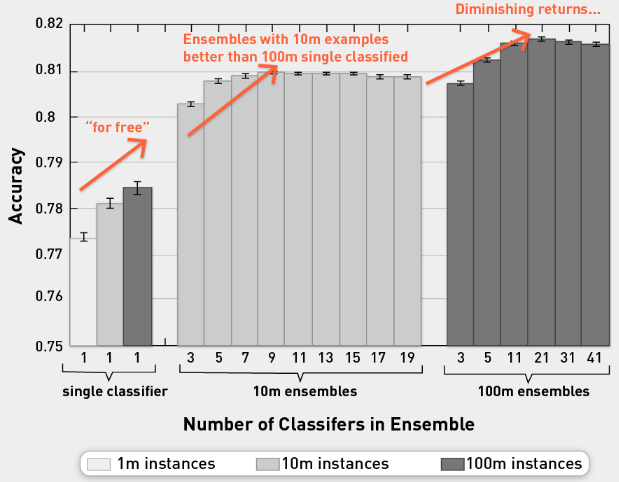
Goals of Class

1. ML at Scale
2. Large-Scale ML
   1. Needs supporting infrastructure
   2. Parallel computing frameworks
3. MapReduce Framework
   1. Allows us to divide and conquer huge data problems
   2. Four frameworks
      1. Cmd
      2. Hadoop
      3. MRJob
      4. Spark
4. Embarrassingly Parallel Problems
   1. Most ML problems
   2. Divide into tasks that can easily be distributed and parallelized
   3. Divided into subproblems
   4. Require little or no communication
   5. Are easily distributed
   6. Have linear computational flow and scale
5. Framework Requirements
   1. Scalable
   2. Fault tolerant
   3. Iterative
   4. Interactive
   5. 
   6. Frameworks that empower us to do above
   7. OS ecosystem
6. Four MR frameworks
   1. CMD based
   2. Hadoop: Storing and manipulating data
   3. MRJob: Convenient Python application programming interface (API) for writing MapReduce programs
   4. Spark: memory backed with integrated framework; Specialized libraries for many things
7. Data analytics stack
   1. 
   2. Populating the stack
   3. 
   4. Add Spark
   5. 
8. Class Phases
   1. Phase 0: CMD
   2. Phase 1: Hadoop/HDFS
   3. Phase 2: MRJob
   4. Phase 3: Spark
9. ML Algorithms
   1. Adopt them for parallel framework
   2. Supervised machine learning (convex optimization, gradient descent, linear regression, decision trees, ensembles of models, support vector machines)
   3. Unsupervised (expectation maximization, matrix multiplication, alternating least squares)
   4. Graphs (random walks, PageRank, graph search algorithms such as breadth-first search, shortest path)
   5. Hybrid algorithms (supervised machine learning and random walks)
   6. Applications (digital advertising, social media, health care, e-commerce, entertainment, metrics, statistics)

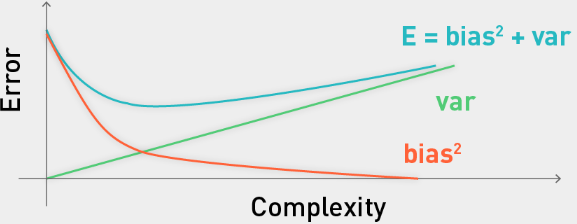
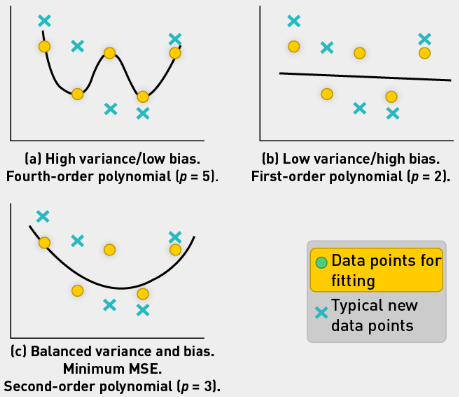
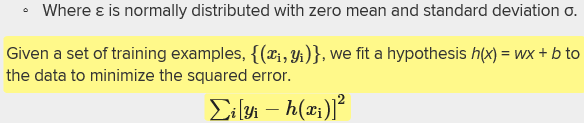
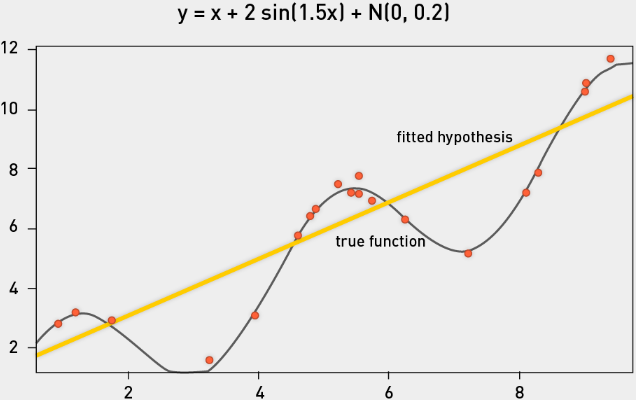
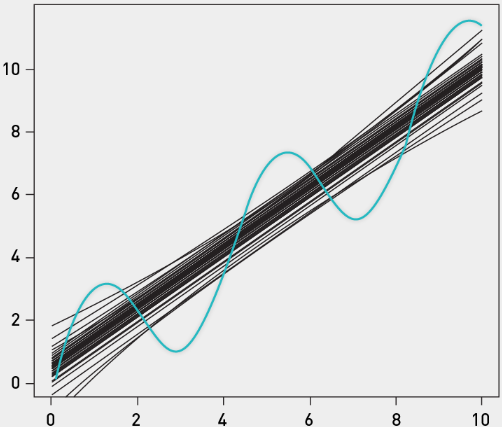
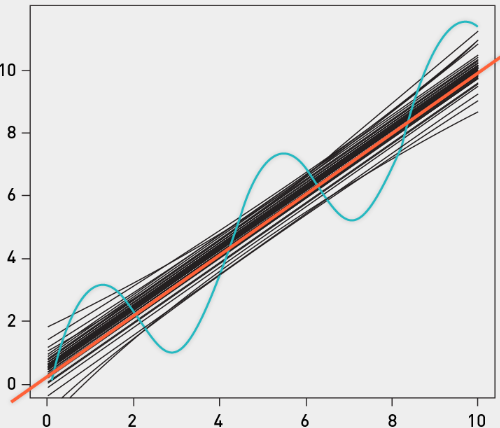
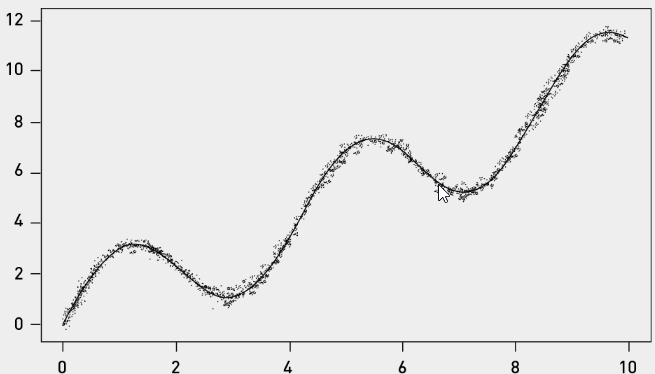
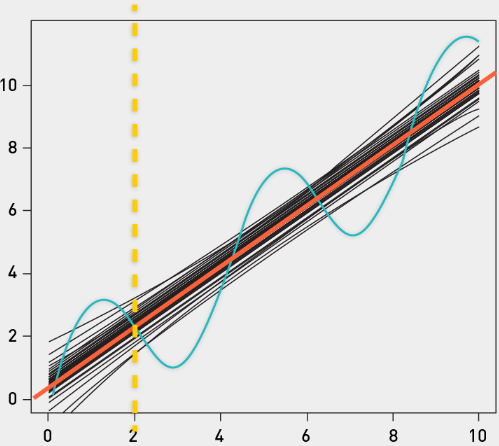
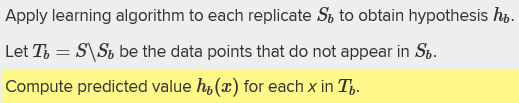
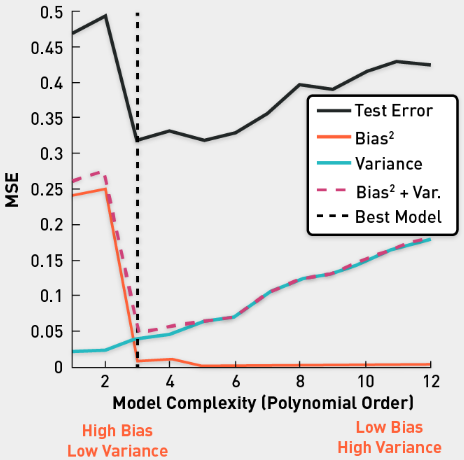
Logistics and Performance Evaluation

1. The more you put in, the more you get back
2. Structure
   1. 13 lectures over 14 weeks
   2. Two exam weeks
   3. Week 8 midterm
   4. Week 15 final project
3. Performance evaluation
   1. 

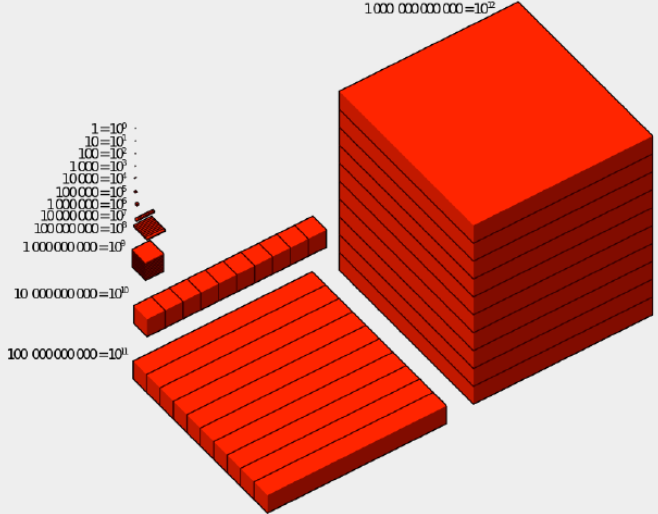
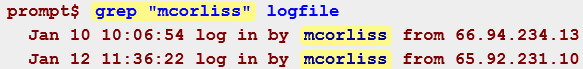
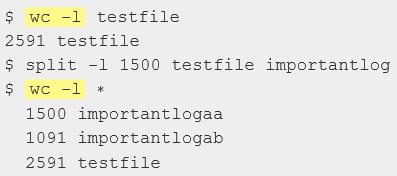
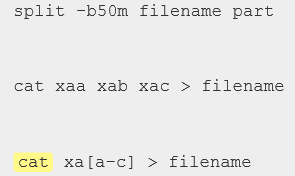
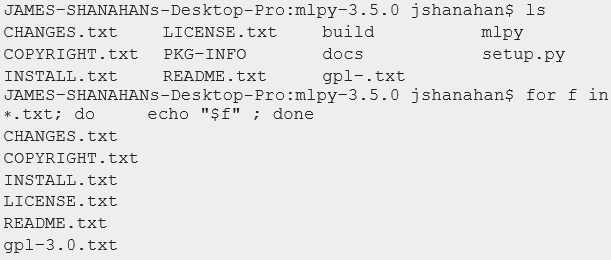
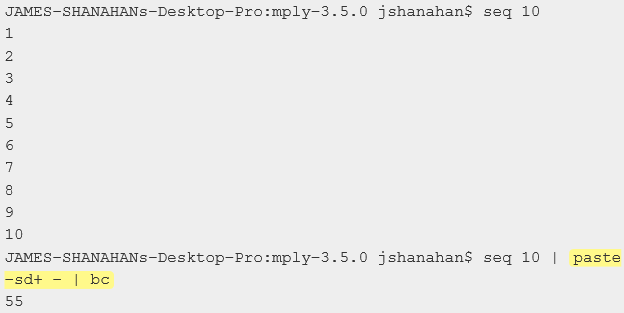
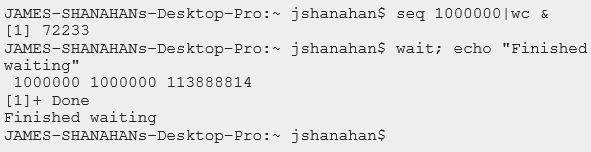
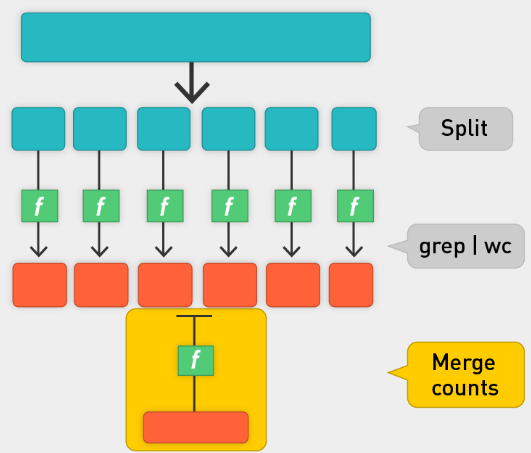
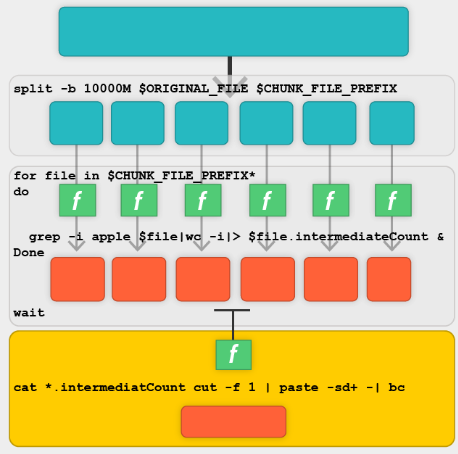
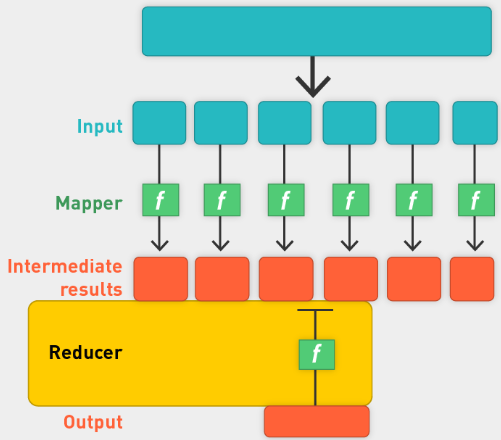
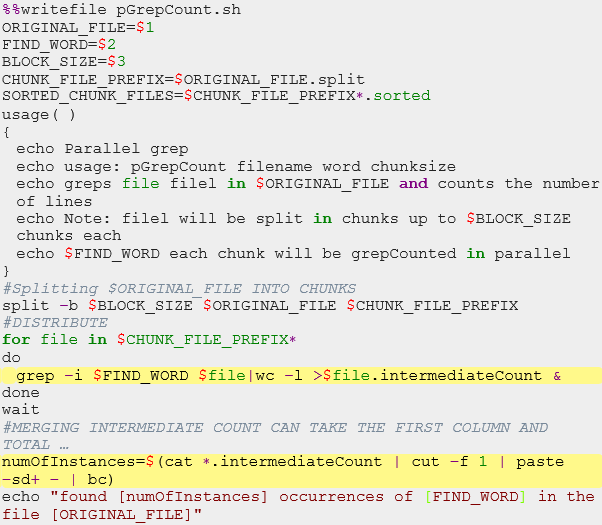
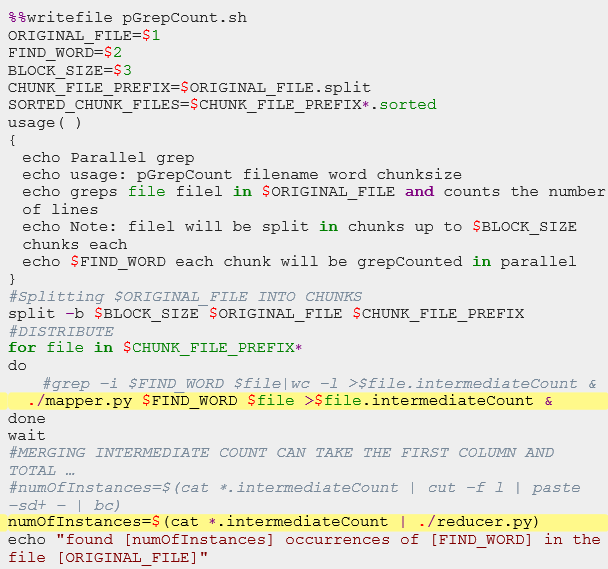
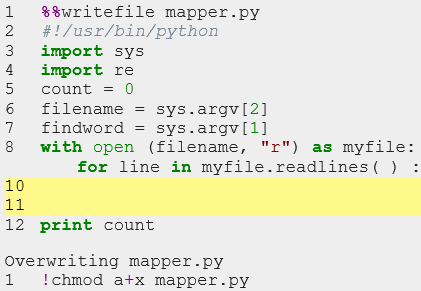
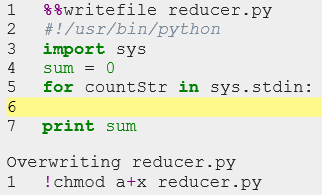
Large Scale ML

1. Do you need more data scientists or do you need more data?
2. ML and Data Study
   1. Filling in confusable words in sentences
   2. Use ML
   3. Vary size of training data
   4. 
   5. Different algorithms perform differently if they have small number of training examples, but similarly with large number of examples
   6. Conclusion: more data leads to 10-20% boost in performance
3. Supervised Classification in a nutshell
   1. 
   2. 
4. Complexity of models
   1. 
   2. Moving from single to ensemble, you get huge boost in performance
   3. However, if you use 100m examples, you get further boost of 1-2%.
   4. Using more data we get amazing improvements, but we hit diminishing returns
5. More data or more DS?
   1. Trade-off between bias and variance
   2. Bias from data scientists
   3. Variance from more data
   4. With more data can reduce variance

Bias-Variance Background

1. B-V
   1. Empirical studies: more data leads to big improvements
   2. More data or more DS?
   3. Bias-variance trade-off
2. ML objectives
   1. Minimize error term
   2. Minimize loss function
   3. Reduce model complexity
   4. Better generalization
   5. For linear regression, we use mean squared error
3. Loss: Irreducible and reducible error
   1. Irreducible:
      1. Has inherent uncertainty
      2. Associated with natural variability in system
      3. Noisy sensors
   2. Reducible:
      1. Can and should be minimized further to max accuracy
4. Reducible error
   1. Includes both "error due to squared bias" and "error due to variance"
   2. Goal: Simultaneously reduce bias and variance as much as possible in order to obtain as accurate a model as is feasible
   3. Trade-offs in selecting models:
      1. Flexibility and complexity
      2. Selecting appropriate training sets
5. Complexity of model
   1. 
   2. Try to minimize error
   3. E is from two components: bias^2 and variance (noise factor as well)
   4. Bias decreasing as complexity increased
   5. Usually, bias is a decreasing function of complexity, while variance is an increasing function of complexity.
6. BV tradeoff in model selection
   1. 
   2. First plot shows high variance as we fit the data too well.
   3. Second plot shows **high bias as we cannot fit the training data well**.
7. Error due to squared bias
   1. Definition: Amount by which expected model prediction differs from the true value, **over the training data**
   2. Bias is introduced at model selection
   3. Repeat model-building process (through resampling) to obtain average of prediction values
   4. If average prediction values are substantially different than true value, bias will be high
8. Error due to variance
   1. Definition: Amount by which prediction **over one training set differs from expected predicted value over all training sets**
   2. Repeat modeling, measure inconsistency over different training data sets
   3. Variance measures how inconsistent predictions are from one another, over different training sets, not whether or not they are accurate (Manning et al., 2008)
9. BV Analysis in Regression
   1. 
   2. 
10. Example
    1. 
    2. We fit linear model
    3. If we resample 50 times and fit 50 linear models and plot them, we get:
    4. 
    5. Use bootstrap sampling, see average model
    6. 
    7. Avg model introduces a bias. Prediction for x=3 is way above actual value for 3.
    8. From variance perspective, models are variable from training set to training set
    9. Noise: low since artificial
    10. 
11. Variance, Bias and Noise
    1. Variance: 
    2. Model prediction minus average model prediction
    3. Bias: 
    4. Average model prediction vs true function value
    5. Noise: 
    6. Observed target value and true function value
12. Distribution of Predictions at x=2.0
    1. 
    2. High variance between 50 models in this case, but true and mean prediction aren’t very different.
    3. X=5 has big difference between true prediction and mean prediction.
13. Measuring Bias and Variance
    1. Simulate multiple training sets by bootstrap replicates.
    2. Construct B bootstrap replicates of S (e.g., B = 200).
    3. 
    4. Use to est bias and variance
14. Model Complexity
    1. 
    2. Experiment where you take training data and use different models: from linear to polynomial regression (go from 1 to 12 polynomial order)
    3. Linear models have very high bias (don’t fit data well)
    4. High polynomial models fit data precisely, lowering bias, but they increase variance (the green line)
    5. Polynomial regression model of order 3 is the best here
15. BV Decomposition
    1. Can be extended to classification problems
    2. Pedro Domingos (2000a; 2000b): Developed unified decomposition that covers both regression and classification
16. BV Tradeoff code:
    1. [Model Selection: Underfitting, Overfitting, and the Bias-Variance Tradeoff](https://theclevermachine.wordpress.com/2013/04/21/model-selection-underfitting-overfitting-and-the-bias-variance-tradeoff/)
    2. [Ask a Data Scientist: The Bias vs. Variance Tradeoff](http://insidebigdata.com/2014/10/22/ask-data-scientist-bias-vs-variance-tradeoff/)
    3. [polyfit: Polynomial curve fitting](http://www.mathworks.com/help/matlab/ref/polyfit.html)
17. Model Selection
    1. Trade off between b and v
    2. Practical method for selecting model
       1. Minimize error
       2. Function of model complexity
18. Effect of algorithm parameters
    1. K-Nearest Neighbor: Increasing k typically increases bias, reduces variance
    2. Decision trees of depth D: Increasing D typically increases variance, reduces bias
    3. Radial basis function (RBF) support vector machines (SVM) with parameter σ: Increasing σ increases bias, reduces variance
    4. Bagging tends to reduce variance without increasing bias
19. Wrap-up
    1. Bias and variance of an estimator are related to squared prediction error.
    2. These concepts can be applied to classification problems.
    3. An optimal estimator will have both low variance and low bias.

MR with CMD

1. Overview
   1. Aim for ML on billions of examples
   2. Use parallel computational framework based on cmd
   3. Divide and conquer strategy
2. Goal
   1. Large scale problem
   2. Decompose into nonoverlapping subproblems
   3. Non-ML problem
   4. Related to NB
3. Task
   1. Find all occurrences of the word apple (by date) in Facebook posts log file.
   2. Parallel grep for FB posts available in a log file
4. Back-of-the-Envelope Calculations
   1. 10^9 active FB users on a daily basis
   2. One post per day = 10^9 posts per day
   3. 
   4. Each post = 1000 bytes (text and metadata)
   5. 1TB of data (1000 \* 10^9 = 10^12 bytes)
5. Processing Times
   1. On one node:
      1. if you scan at 100 MB/s (10^8), it will take you
      2. 
      3. This is way too long
   2. On a 96 multicore machine
      1. Scanning at 100MB/s, assume 100 cores has no latency
      2. 
      3. How do you write code to leverage this power?
6. Strategy: Calculate Term Popularity
   1. Unix-based divide and conquer strategy
   2. Poor man’s distributed computational perspective
7. Ground Rules
   1. Use only below commands
   2. split, grep, wc, merge, cat, for, echo
   3. cut, paste, bc
   4. |, &, wait
8. GREP command
   1. "grep <string> <filename>" returns all lines in file that contain string
   2. Each line corresponds to one record (Facebook post)
   3. Date, time, body of post
   4. 
9. Pipe command: |
   1. Chain together various commands
   2. 
   3. Find lines that contain all three words
10. Split command
    1. Split by number of lines: "split -l 30" splits file into chunks of 30 lines or less
    2. Split by number of MB: ◦"split -b 24m" splits file into chunks of 24MB or less
    3. Example Split command:
       1. 
       2. Wc counts number of lines in this case
       3. 
       4. Cat displays the data or feeds it into another file
11. For loop and Echo
    1. 
12. Cut, Paste, bc
    1. Seq just generates list of numbers
    2. 
    3. Paste takes the numbers produced by the sequence and creates a string with a plus sign between each of the numbers.
    4. Bc then takes that string and executes it as a calculator, summing up the count
13. Parallel Computing with & and Wait
    1. "&" causes parent process to spawn off parallel processes
    2. "wait" will cause parent process to wait until child processes have finished
    3. 
    4. "seq 1000000|wc &" causes the parent terminal process to spawn line-count task off as child process
    5. "wait" causes the shell to freeze until child process finishes
    6. It will print Finished waiting when wait is over
    7. Once child command finishes, shell can continue and run echo command
14. Schematic of Parallel Processing
    1. Split file in chunks of 10G each
    2. Take grep function and grep each chunk | wc to count how many matches the grep generated in each chunk. Output is number of matches per chunk
    3. Merge these counts together
    4. 
15. Unix Commands
    1. 
16. MapReduce Pattern
    1. 
17. MapReduce Framework
    1. 
    2. How about using python?
    3. 
    4. Write mapper.py and reducer.py
    5. Change execution privileges
18. Python: Mapper for Grep
    1. 
    2. If findword in line
    3. Count = count + 1
19. Python Reducer for Grep
    1. Barrier pattern: we cannot start reduce function until all mapper processes are done
    2. 
    3. Sum = sum + int(countStr)