Data Science

A Primer

by Robert Schroll

(TDI data scientist in residence and instructor, age 39)

Dear Reader,

Over the past few years, data science has become increasingly popular. And increasingly complex.

And that's why we're here.

While the tools of data science can be complicated, the goals are pretty straightforward, and are probably the same goals your team or company already have. They're just buried under the new terminology of data science.

The goal of this primer is to demystify common data science terminology and connect it to the goals and processes already present in your organization.

Now, it won't teach you how to do data science, but it should give you a fighting chance to have a meaningful conversation with a data scientist.

And just be forewarned that this primer might be a little opinionated (okay, highly opinionated). It focuses on the tools and techniques that we teach at Pragmatic Institute and The Data Incubator, because these are the ones we think are best and are most commonly used.

Data literacy is a critical skill more professionals need to have, not just for business but for the world in general. I hope this primer helps you along the path to understanding how data can help you.

Enjoy!

Robert S

Two Main Components of Data Science

Data Science is developing a method for taking lots and lots and lots of information to make something useful.

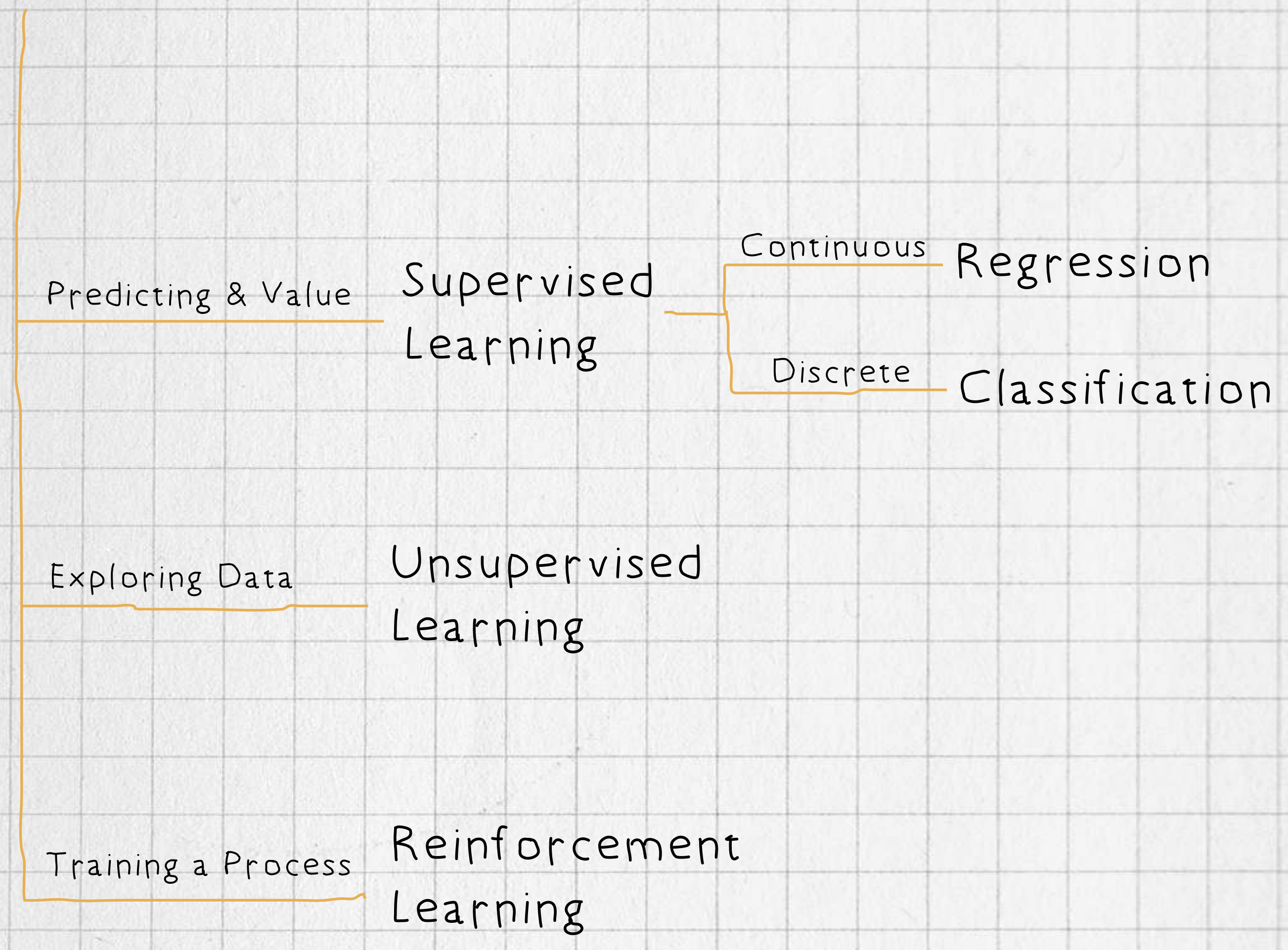
Machine Learning

- Basically model building, but with very complicated models
- Has become possible with modern computers
- Becomes really powerful when you can learn from ("train on") lots of data

Distributed Computing

- Needed to deal with BIG DATA - Volume, Variety, Velocity
- When you have too much data to fit on one machine (Dirty secret: very few really need this)
- Need high-powered tools to understand data at this scale

Machine Learning (ML)


Predicting & Value — Supervised Learning

Continuous — Regression

Discrete — Classification

Exploring Data — Unsupervised Learning

Training a Process — Reinforcement Learning

*Aside:
Machine Learning is a subset of Artificial Intelligence, which is the field of making computers look smart.

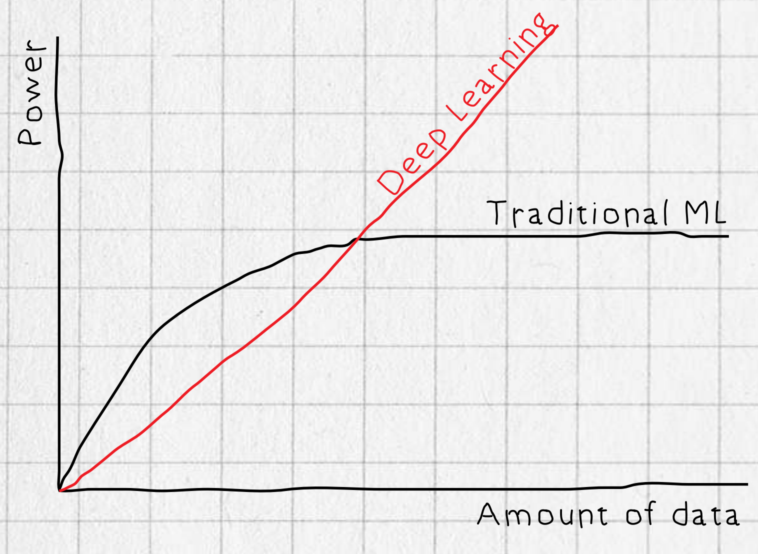
Uses for ML

🟩 Descriptive What happened?

🟩 Predictive What will happen?

🟦 Prescriptive What should I do?

ML Algorithms



Traditional ML

(has many connections to statics)

- Linear Regression
- Logic Regression
- Decision Trees
- Support Vector Machines
- Random Forests
- Gradient Boosting
- Clustering
- Dimensionality Reduction
- PCA (Principal Component Analysis)

Deep Learning

- Mostly equivalent to neural networks
- Powerful, flexible models take a lot of time and data to train
- Lots of cool results recently, thanks to GPUs

ML Tools

(Python focused)

- **Scikit-learn (sklearn):** Traditional ML in Python
  (Used everywhere for exploration)
- **XGBoost:** Gradient Boosting
  (many languages)
- **Spark ML:** Traditional ML in distributed systems
  (Scala, Python, others)
- **TensorFlow:** Neural Networks, from Google
  (Python and other languages)
- **PyTorch:** Neural Networks, from Facebook
  (Python and other languages)

ML Tasks

- **Natural Language Processing (NLP)**
  Understanding human language
- **Image Processing/Computer Vision (CV)**
  Identifying object + segmentation
- **Times Series Analysis**
  Any task where past influences the future
- **Anomaly/Outlier/Novelty Detection**
  Is this data unusual?
- **Recommendation Engine**
  What product would this user like?
- **Churn Prediction**
  Identify customers we will lose
- **Risk Assessment**
  Who is likely to default?
- **Optimization**
  Fastest/cheapest/most efficient way to do X

Distributed Computing

I can speed up a simple repetitive task by having many computers all do a part.

This is a subset of parallel computing.

Generally restricted to tasks that are "easy" to parallelize:

- Processing a bunch of records
- Calculating statistics
- Training a Model

Distributed Computing Tools

○ Hadoop: Framework for organizing a bunch of computers

○ MapReduce: Early library for distributed computing (runs on Hadoop)

○ Spark: Current most popular distributed computing framework (in Scala)

○ PySpark: Lets you write Spark code in Python (Scala, Python, others)

○ Ray: New distributed computing framework from same people who made Spark

○ TensorFlow: Keeps gaining distributed features

Languages and Libraries

1/2

Python: Scripting language popular in DS. Slow, but expressive. Huge library. What we teach

- Pandas: Library for structured data (DataFrames)
- NumPy: Low-level library behind Pandas (and others)
- Matplotlib: Venerable plotting library
- Altair: Modern plotting
- Jupyter: Interactive interface to Python (and other languages)
- Sklearn, TensorFlow, PyTorch, XGBoost, pySpark

R: Language designed for data. Popular with statisticians and biologists

SQL: Database language from 1974 (!). Many tools use it as a common standard. Arguably the most in-demand language

Languages and Libraries

2/2

Scala: Language built on top of Java, a popular enterprise language. Notable for being the native language of Spark

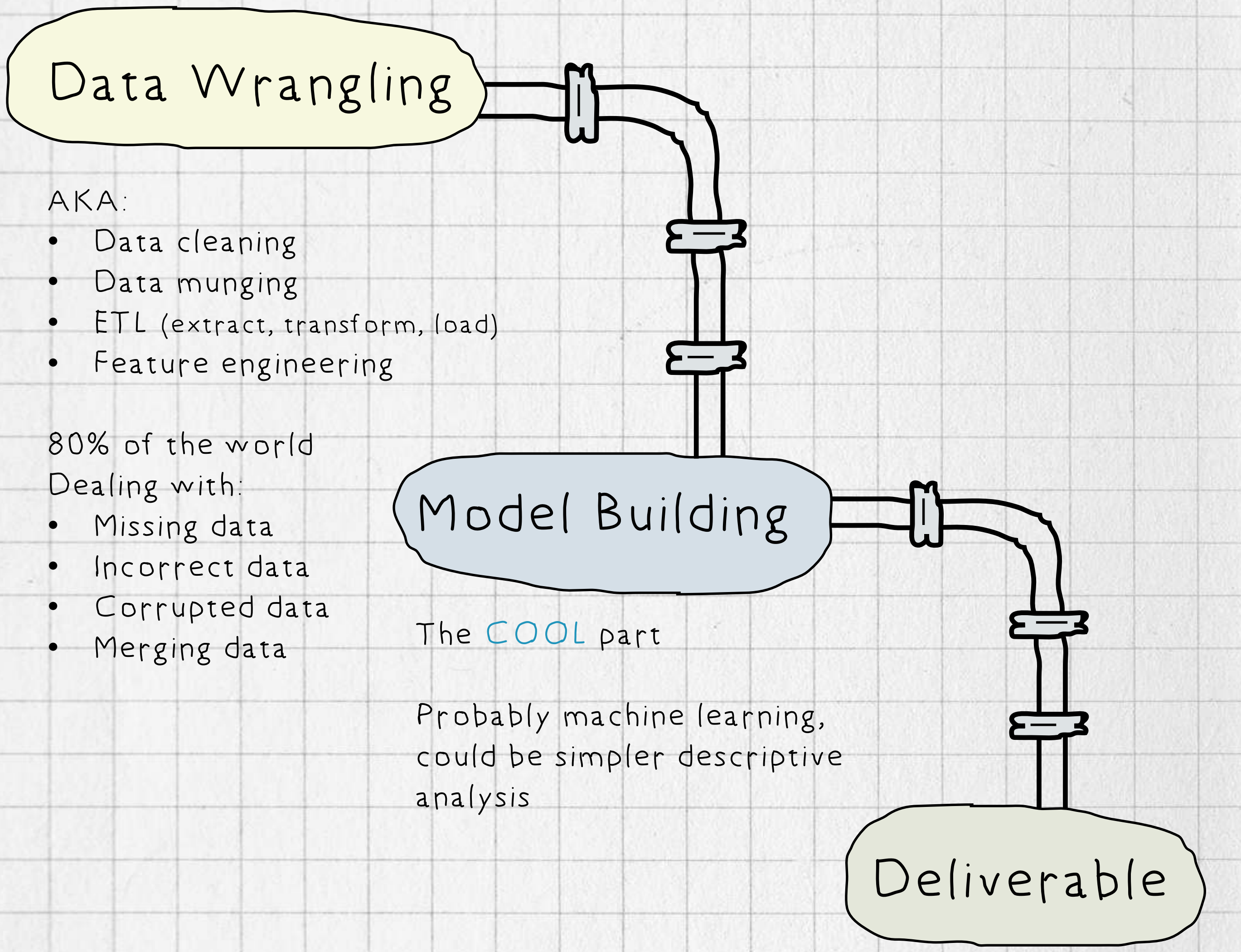
Javascript: Language of web browsers. Not actually related to Java. Many modern visualizations tools use the browser, including DS, Vega

Tableau: Proprietary visualization toolkit. Pretty, but simplistic

Alteryx: Proprietary analytics platform

C, C++, FORTRAN: Low-level systems programming languages. Runs fast, but hard to write. May be needed to put Data Science into production

The Data Science Pipeline



AKA:

- Data cleaning
- Data munging
- ETL (extract, transform, load)
- Feature engineering

80% of the world

Dealing with:

- Missing data
- Incorrect data
- Corrupted data
- Merging data

The COOL part

Probably machine learning, could be simpler descriptive analysis

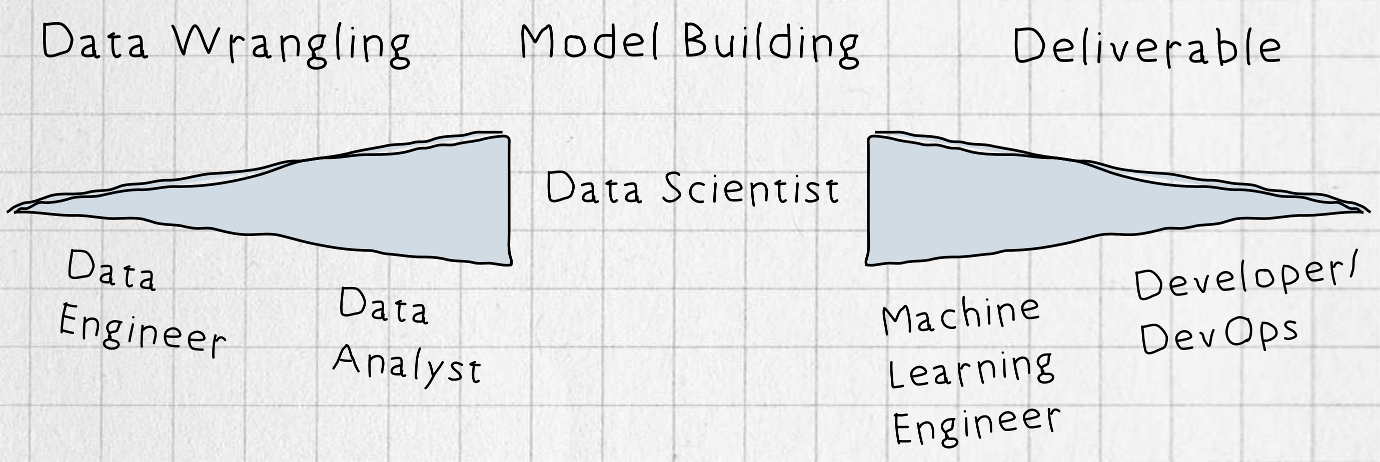
PowerPoint presentation

Visualizations

Interactive Dashboard

Deploying a model into production

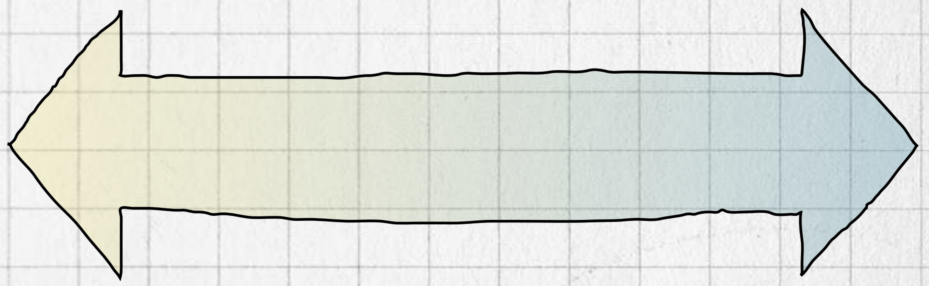
The Data Science Jobs



Other Titles:

- Business Intelligence
- Business Analyst
- Advanced Analytics
- Predictive Analytics
- Marketing Analytics
- Quant (Finance)
- Statistician (Banking, Pharma)
- Actuary (Insurance)

Data Science Applications



Evolutionary	Revolutionary
Make an existing process better with data science Eg: Marketing ads to individuals, not a whole city	Building a product that is only possible because of data science Eg: self-driving cars
Practitioners are not necessarily "data scientists"	Almost definitely data scientists on the team
Lost of gains to be had in the "data wrangling" step	Need full pipeline working for any benefit
Existing staff has domain knowledge, but may not know any data science	Existing staff already data science experts

About the Author

Robert Schroll is a data scientist and instructor for Pragmatic Institute and its sister company, The Data Incubator. Robert studied squishy physics in Chicago, Amherst and Santiago, Chile, before uniting his love of computers, teaching and making pretty graphs at Pragmatic Institute.
In his free time, Robert plays the tuba and right field, though usually not simultaneously.

About Pragmatic Institute

Pragmatic Institute provides comprehensive training, education and certification to data practitioners, product managers and product marketers globally. With a commitment to excellence and a dedication to continued education, Pragmatic Institute's full-service offerings enable organizations to grow revenue, harness the power of their own data, go to market faster, improve customer satisfaction ratings and more.

PragmaticInstitute.com

theDataIncubator.com

Copyright © 2020. Pragmatic Institute, LLC and The Data Incubator. All rights reserved.