**Additional Resources (Joydeep Ghosh)**

1. SGD and Online learning for big data
   1. <http://leon.bottou.org/projects/sgd> (lots of stuff; code, papers, talks) in particular look at   
      leon.**bottou**.org/publications/pdf/compstat-2010.pdf‎
2. Convex Optimization
   1. <https://www.stanford.edu/~boyd/cvxbook/> free book + slide
   2. MOOC by Boyd https://class.stanford.edu/courses/Engineering/CVX101/Winter2014/about
   3. Distributed optimization (technical)

[Distributed optimization and statistical learning via the alternating direction method of multipliers](https://www.stanford.edu/~boyd/papers/admm_distr_stats.html) S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein

1. Text Processing:
   1. Python NL toolkit <http://www.nltk.org/>
   2. Stanford NLP software http://nlp.stanford.edu/software/index.shtml
   3. Prof Mooney’s NLP notes http://www.cs.utexas.edu/~mooney/cs388/
   4. Conditional Random Fields (for text extraction/processing) <http://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf>
2. Modern Recommender Systems
   1. <http://pages.cs.wisc.edu/~beechung/icml11-tutorial>
   2. <https://www.youtube.com/watch?v=bRzOBGLCRbc>
   3. **Content recommendation on web portals** <http://dl.acm.org/citation.cfm?id=2461277>
3. **Deep Learning**  
   Bengio’s book draft <http://goodfeli.github.io/dlbook/>

<http://deeplearning.net/>

videos at http://videolectures.net/deeplearning2015\_montreal/

1. Database side of BIGDATA
   1. See PASS talk 2011 by DeWitt: <http://pages.cs.wisc.edu/~dewitt/includes/passtalks/passtalks.html>
   2. New 2014 book: *Data Just Right: Introduction to Large-Scale Data & Analytics*, by M. Manoochehri, gives a high level overview of system options (hardware/software)
2. Multi-Task Learning (learning many related tasks simultaneously, e.g. spam filtering)
   1. <http://en.wikipedia.org/wiki/Multi-task_learning>
   2. [Feature hashing for large scale multitask learning](http://scholar.google.com/citations?view_op=view_citation&hl=en&user=jsxk8vsAAAAJ&citation_for_view=jsxk8vsAAAAJ:Y0pCki6q_DkC)  
      KQ Weinberger, A Dasgupta, J Langford, A Smola, J Attenberg, ICML, 2009.

Is also related to **Transfer Learning**, where knowledge gained in solving one problem is “transferred” to help solve a related problem.

See <http://www1.i2r.a-star.edu.sg/~jspan/SurveyTL.htm>

**Simpler, Newer Tutorial/Utility Stuff**

**Jupyter** (<http://jupyter.org/>) (old: iPYTHON Notebook) is a web-based interactive computational environment where you can combine code execution, text, mathematics, plots and rich media into a single document

* **ISLR book now has a MOOC:** [https://class.stanford.edu/courses/HumanitiesScience/StatLearning/Winter2014/about](https://class.stanford.edu/courses/HumanitiesScience/StatLearning/Winter2014/about" \t "_blank)
* exploratory data analysis in R mooc (not free) <https://www.udacity.com/course/ud651>

part of their Data Science track. Mostly very basic info.

* Mashup creation: KARMA <http://www.isi.edu/integration/people/knoblock/projects/prj_mashup_construct.html>

**Mailing lists:**

Industry: kdnuggets.com; Datanami;

Academic: JMLR provides a list of abstracts as papers gets published, if you sign up.

**Python Resources: (with input from Tal Yarkoni and Sanmi Koyejo)**

The book *Python for Data Analysis* mostly talks about data handling/pre-processing rather than predictive modeling. http://shop.oreilly.com/product/0636920023784.do

For machine learning in particular, the scikit-learn documentation is really good, and includes a lot of well-commented examples.

For data wrangling etc. in Python, the pandas tutorials are solid:

[http://pandas.pydata.org/pandas-docs/stable/tutorials.html](http://pandas.pydata.org/pandas-docs/stable/tutorials.html" \t "_blank)

For someone coming from a Matlab background, this cheat sheet is invaluable:

[http://wiki.scipy.org/NumPy\_for\_Matlab\_Users](http://wiki.scipy.org/NumPy_for_Matlab_Users" \t "_blank)

also see [http://scipy-lectures.github.io/](http://scipy-lectures.github.io/" \t "_blank)

scipy.optimize for basic optimization ([http://docs.scipy.org/doc/scipy/reference/optimize.html](http://docs.scipy.org/doc/scipy/reference/optimize.html" \t "_blank)).

For vanilla Python it's probably easier to just pick it up in the context of a specific domain (e.g., pandas or sklearn) and then google for anything else that comes up. Given that Python is such a high-level, dynamic language, anyone originally trained in C/C++/Java etc. is going to find it a breeze getting into. Though it probably doesn't hurt to skim the official doc:

[https://docs.python.org/2/tutorial/](https://docs.python.org/2/tutorial/" \t "_blank)

[http://scipy-lectures.github.io/](http://scipy-lectures.github.io/" \t "_blank), plus [http://wiki.scipy.org/NumPy\_for\_Matlab\_Users](http://wiki.scipy.org/NumPy_for_Matlab_Users" \t "_blank) is very useful when moving from matlab.

scipy.optimize for basic optimization ([http://docs.scipy.org/doc/scipy/reference/optimize.html](http://docs.scipy.org/doc/scipy/reference/optimize.html" \t "_blank)).

**“Professional” Alternatives to WEKA.**

Weka is generally considered “Mickey Mouse” in industry. Though it is useful for pedagogical purposes (and I used it for an undergrad course once), many people will laugh if you say that is what you use. However, there are other menu-drive open source packages with similar feel/GUI, that scale to industry sized problems (at least if you purchase a commercial license), and whose statistical functions are much more solid. See

RapidMiner http://rapidminer.com/

Knime <http://www.knime.org/>

**On Hadoop:** I predict the hype of Hadoop being “be all and end all” will die down quite quickly, alternatives such as SPARK <http://spark.apache.org/> and Graphlab (and commercial versions thereof) will gain more traction for large-scale data science, and data scientists will mix-and-match

**MOOCs**: Top Coursera Data Science Specializations

http://www.kdnuggets.com/2015/11/coursera-specializations.html/3