**So, you wanna start practicing Data Science/Machine learning?**

Contact: Dan Ofer **דן עופר**

[ddofer@gmail.com](mailto:ddofer@gmail.com)

0524600688

101: The point of all this is to learn and to have fun!

**Learn to use:** Sci-kit Learn (Go through tutorials or examples). Python. [Pandas](https://github.com/jvns/pandas-cookbook). (iPython is also useful, in addition to a nice IDE (Spyder = included with anaconda) , Sublime Text 3 or Pycharm).

[scikit-learn: machine learning in Python —](http://scikit-learn.org/) **scikit**-**learn**.org/  
Numpy, scipy are also useful, but you can do without understanding them (for now). Take a look at some tutorials or [examples](http://ipython-books.github.io/featured-04/). [Quick Start Tutorial](http://scikit-learn.org/stable/tutorial/basic/tutorial.html). [Example Gallery](http://scikit-learn.org/stable/auto_examples/index.html). [iPython notebooks tutorial from PyData](https://github.com/bugra/pydata-nyc-2014)

Install everything via the [**Anaconda**](http://continuum.io/downloads) Python distribution. (2.7 and/or 3). 2.7 is Recommended due to increased compatability with some neural network/”Deep learning” libraries (such as Theano, Lasagna).

http://continuum.io/downloads

Read up on [KAGGLE](https://www.kaggle.com) – Look at the wiki, white papers, etc’ .

Each contest takes place over a few months.

We’ll be meeting once a week, for 1-2 hours. More work will be done overall by people at home, as people’s time and schedules permit.

In the future we’ll try working with other packages for “deep learning” and other methods. (e.g. Theano, [Lasagna](http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/), [Breze](http://breze.readthedocs.org/en/latest/), [NoLearn](https://github.com/dnouri/nolearn), Deepy, PY-ELM (Extreme learning machines), Caffe, [Passage](https://github.com/dnouri/Passage)). [**SciKit-Learn Laboratory**](https://scikit-learn-laboratory.readthedocs.org/en/latest/index.html) **(**[SKLL](https://scikit-learn-laboratory.readthedocs.org/en/latest/tutorial.html#titanic-config)). I haven’t practiced with them myself yet.

*Read these articles* (in addition to some of the material in scikit learn’s website, and your own readings):

[A few useful things to know about machine learning - P Domingos](http://dl.acm.org/citation.cfm?id=2347755).

[An introduction to variable and feature selection - I Guyon - Journal of Machine Learning Research](http://dl.acm.org/citation.cfm?id=944968)

When you have time, a really good resource is the Master’s thesis [“Concepts in Predictive Machine learning”](http://www.davidwind.dk/wp-content/uploads/2014/07/main.pdf) , AKA [How to Kick Ass in Competitive Machine Learning](http://machinelearningmastery.com/how-to-kick-ass-in-competitive-machine-learning/)

Current competitions of Interest:

[**MNIST - Digit Recognizer**](https://www.kaggle.com/c/digit-recognizer) – Relatively Easy. Lots of implementations available. Can practice with CNN (Convolutional neural networks), or other methods. Feature engineering fun! Images.

<http://nicklocascio.com/neural-net-mnist-kaggle/>

[Applying deep learning and a RBM to MNIST using Python](http://www.pyimagesearch.com/2014/06/23/applying-deep-learning-rbm-mnist-using-python/)

[Using Scikit-Image and real world digits data - Digit Recognition using OpenCV, sklearn and Python](http://hanzratech.in/2015/02/24/handwritten-digit-recognition-using-opencv-sklearn-and-python.html)

[**Forest Cover Type Prediction**](https://www.kaggle.com/c/forest-cover-type-prediction)  - Semisupervised learning. Larger data, not a “solved” problem (unlike the highly studied MNIST).

<https://shankarmsy.github.io/posts/forest-cover-types.html>

<http://nbviewer.ipython.org/github/aguschin/kaggle/blob/master/forestCoverType_featuresEngineering.ipynb>

There are many more challenges of course. We’ll start with the aforementioned two, then move on (see below for options). Read up on them and Kaggle (rules, guides, blog posts, wiki, rules, etc’).

[**Otto Group Product Classification Challenge**](https://www.kaggle.com/c/otto-group-product-classification-challenge) **–** Small Dataset. 93 Obfuscated Features. Multiclass probabilistic classification. Good opportunity to practice on!

[**Microsoft Malware Classification Challenge**](https://www.kaggle.com/c/malware-classification)  - Prizes. Harder. Lots of work required to create features. ~400GB of raw data.

[**Diabetic Retinopathy Detection**](https://www.kaggle.com/c/diabetic-retinopathy-detection)- Biomedical image data. ~100GB. Image classification. Hard. Noisy.

Take a look at some iPython notebooks on Pandas/data wrangling / machine learning 101 with scikit learn and the like.

Just for example:

[Introduction to Machine Learning with Python and Scikit-Learn](http://kukuruku.co/hub/python/introduction-to-machine-learning-with-python-andscikit-learn) - <http://kukuruku.co/hub/python/introduction-to-machine-learning-with-python-andscikit-learn>

<http://kachkach.com/data-processing-and-machine-learning-with-python/>

<http://nbviewer.ipython.org/github/halflings/python-data-workshop/blob/master/data-workshop-notebook.ipynb>

[Random Forests with scikit-learn](http://nbviewer.ipython.org/github/shankarmsy/practice_Random_Forests/blob/master/Random%20Forests%20with%20scikit-learn.ipynb) (Has examples of plotting learning curves, Feature importance).

<https://github.com/bugra/pydata-nyc-2014Pydata%20NYC%202014%20Scikit%20Learn%20Tutorial%20%20-%20https:/github.com/bugra/pydata-nyc-2014>

<http://nbviewer.ipython.org/github/shankarmsy/pandas-cookbook/tree/master/cookbook/> - Pandas cookbook – examples for data munging ,reading in a CSV, etc’ . (Separate from the [Actual 10 minutes to pandas cookbook](http://pandas.pydata.org/pandas-docs/dev/cookbook.html)). (Lots of other very good [tutorials](http://nbviewer.ipython.org/github/shankarmsy/sklearn_pycon2014/tree/master/notebooks/) !)

[Learning Python for Social Scientists](http://nealcaren.github.io/python-tutorials/) (List of tutorials) - <http://nealcaren.github.io/python-tutorials/>

[Data Visualization using Seaborn, matplotlib, pandas – Example](http://nbviewer.ipython.org/github/olgabot/pycon2014_dataviz/blob/master/pycon2014_dataviz.ipynb) - <http://nbviewer.ipython.org/github/olgabot/pycon2014_dataviz/blob/master/pycon2014_dataviz.ipynb>

[https:/github.com/rasbt/pattern\_classification/blob/master/resources/python\_data\_libraries.md](https://github.com/rasbt/pattern_classification/blob/master/resources/python_data_libraries.md)

<http://www.metacademy.org> - Gives a big list of ML topics, and how to study them. (Including links to textbooks and courses)

Some Practice Questions:

תענו ***בקצרה*** וברמה “high level”.   
תעברו על הכל (לא חובה לענות על הכל. בפרט אם אתם לא יודעים\לא מבינים) . אל תעתיקו מ Wikipedia. כן תרגישו חופשיים להיעזר ב Google.

**תענו *לפחות* על מה שרשום באדום!** (תרשמו או תשרטטו ותשלחו לי) = Please DO answer at least everything that’s in RED, and send me your answers.

Do not write overly long answers, I’m looking for a sign that you know the concepts or looked them up, not חפירות or proofs. (sketches or doodles are fine to add).

You can also ask about the questions when we meet, including the other people. The point is to find about some of these things, not to quiz you to death.

e.g. “What’s the idea behind Machine learning?” = *Teaching machines/algorithms to learn from data.*

e.g. “PCA gives us?” = *Principle components containing the most variance/variability (שונות) in the data. It’s an unsupervised method. Very useful for dimensionality reduction. (See Eigenvectors, Eigenvalues, covariance, SVD)*

1. How do you test model performance? (AKA "How do you validate your findings?")
2. Explain what *regularization*is and why it is important.  In general, What happens when using L1 vs L2 regularization?
3. Feature Selection: What are the benefits? Name 2-3 different approaches, and an example for each approach.
4. In general, what do you get from PCA? (Principle Components Analysis).
5. X = m X n matrix of m samples, n features. Y = vector of labels for each sample. Clf = a ML model from scikit learn. How do you train your model on the data, and test its performance? Write out the scikit-learn code/pseudocode. (Hint – look at the tutorials. It’s cut and paste)
6. How would you try to improve a given model’s performance over the default? Give pseudocode! (Hint: Tune Hyperparameters. GridSearchCV. Hint – look at the tutorials)
7. Explain what precision and recall are. How do they relate to the ROC curve?
8. What do Accuracy and the F1 metric measure?
9. Supervised Classification, Regression, Unsupervised learning are used to predict ??? . What is the difference between each of them? Give examples. ([hint](http://www.astroml.org/sklearn_tutorial/general_concepts.html))
10. Median = ? Standard Deviation = ?
11. What are feature vectors?
12. Give an example of an ensemble method. Why use Ensembles?
13. What’s a kernel? (e.g. in the context of SVMs). What’s good about the kernel trick? (תשובה כללית)
14. "How would you decide which modeling technique to use?". (Give your Personal opinion/thought process).
15. Give 2 examples of how you would find important features from your data/model.
16. What’s a common winning strategy for Kaggle contests? (Hint: It’s in the blog and their “White papers”. And interviews. ~5 words MAX).
17. What’s FDR and what is it used for? (Hint: P-values).
18. Bagging, Boosting = ?
19. What’s the Bias–variance dilemma?
20. What’s overfitting and underfitting?
21. How can you avoid overfitting ?
22. What are model Hyperparameters?
23. What’s the idea behind autoencoders (Neural networks)?
24. Given a problem-set from Kaggle (e.g. MNIST) – What features would you add? How would you choose which features to keep?
25. How would you approach the Netflix Prize?
26. What are Random Forests? SVM? Neural Networks?
27. What are GitHub and Dropbox?
28. How can we “Standardize” or “normalize” features? (e.g. Prior to using them as input to a SVM)? [hint](http://kukuruku.co/hub/python/introduction-to-machine-learning-with-python-andscikit-learn).
29. Given a problem, Why would you use random forests (or any other “basic” model) instead of a deep convolutional (or recurrent) neural network? (hint – Practical concerns).
30. What are some common feature extraction techniques for: Text. Audio/time-series. Images? ([Hint/Ideas](http://www.astroml.org/sklearn_tutorial/general_concepts.html))
31. What’s TF-IDF? How is it different from the basic “Bag of words” approach? What’s the benefit?
32. Write pseudocode to count the number of (unique) words in a document using Python. (Try this for bi-grams!).

Some general learning resource recommendations:

*Nobody is expecting you to read all (or even most) of these!!! This is for anyone whose looking for some resources or starting points..*

Blogs.

<http://blog.kaggle.com/2011/03/23/getting-in-shape-for-the-sport-of-data-sciencetalk-by-jeremy-howard/>

Kaggle White papers + Forums. (<http://blog.kaggle.com/> )

<https://github.com/savarin/pyconuk-introtutorial> - Practical introduction to pandas and scikit-learn via Kaggle problems

<http://www.datatau.com/newest>

<http://www.reddit.com/r/MachineLearning/wiki/index>

[**DataRobot**](http://www.datarobot.com/blog/)– Lots of VERY good basic guides.

e.g.: <http://www.datarobot.com/blog/a-primer-on-deep-learning/>

[Educational Resources for Getting Started in Data Science](http://www.datarobot.com/blog/educational-resources-for-getting-started-in-data-science/)

[GBRT Guide (Gradient Boosted Regression Trees)](http://www.datarobot.com/blog/gradient-boosted-regression-trees/)

[**Yhat**](http://blog.yhathq.com/)– Lots of very good blog posts. (Older ones more so). EG:

[Predicting customer churn with scikit-learn](http://blog.yhathq.com/posts/predicting-customer-churn-with-sklearn.html) – examples of visualization, feature importance, etc ‘

[Data Science in Python](http://blog.yhathq.com/posts/data-science-in-python-tutorial.html) - IPython notebooks introduce using Python for data science.

[Random Forests in Python](http://blog.yhathq.com/posts/random-forests-in-python.html)

https://darshanhegde.wordpress.com/2014/08/19/learn-machine-learning-the-hard-way/

* [Machine Learning 101](http://www.erogol.com/large-set-machine-learning-resources-beginners-mavens/) [- Large set of ML resources for beginners through Mavens.](http://www.erogol.com/large-set-machine-learning-resources-beginners-mavens/) (Great blog!)
* [Random Forest Explained](http://citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics/) - Write up about Random Forest in layman's terms.

<http://www.zipfianacademy.com/blog/post/46864003608/a-practical-intro-to-data-science> - GOOD resource

<http://radimrehurek.com/data_science_python/> - Good end to end code example of building a SPAM detector, with code and model tuning in scikit learn.

[FastML](http://fastml.com/) - **fastml**.com/ - **Great** blog, great articles, includes code. Look at [Deep learning made easy](http://fastml.com/deep-learning-made-easy/), [The secret of the big guys](http://fastml.com/the-secret-of-the-big-guys/) , [ELM (extreme learning machines)](http://fastml.com/extreme-learning-machines/) and his Git code/implementations.

Also links to a lot of other good blogs - <http://fastml.com/links/> :

<http://hunch.net/>

[http://blog.echen.me](http://blog.echen.me/)

<http://colah.github.io/>

<http://sebastianraschka.com/> - Lots of REALLY good blog posts and tutorials. Very step by step.

<http://mlwave.com/> - Lots of posts on Kaggle competitions

[Kaggle Blog](http://blog.kaggle.com/)

<http://trevorstephens.com/>

http://www.johndcook.com/blog/

<https://www.coursera.org/learn/machine-learning>

[**Stanford Unsupervised Feature Learning and Deep Learning Tutorial** – (ufldl) deeplearning.stanford.edu/tutorial/](Stanford%20Unsupervised%20Feature%20Learning%20and%20Deep%20Learning%20Tutorial%20-%20deeplearning.stanford.edu/tutorial/)

<http://snippyhollow.github.io/blog/2014/08/09/so-you-wanna-try-deep-learning/> - Good blog, good GitHub code

http://www.kdnuggets.com/2015/03/10-steps-success-kaggle-data-science-competitions.html

<http://www.quora.com/How-do-I-learn-machine-learning-1>

[The Hitchhiker’s Guide to Kaggle](http://www.slideshare.net/ksankar/oscon-kaggle20)

[Data Wrangling for Kaggle Data Science Competitions - An etude](https://us.pycon.org/2014/schedule/presentation/61/)

<https://news.ycombinator.com/item?id=1055042>

<http://datasciencemasters.org/>

<http://danielnouri.org/notes/2014/12/17/using-convolutional-neural-nets-to-detect-facial-keypoints-tutorial/> - Lasagna. (Look also at [Using deep learning to listen for whales](http://danielnouri.org/notes/2014/01/10/using-deep-learning-to-listen-for-whales/) - VERY interesting ideas I plan to use).

<http://www.quora.com/What-are-the-key-skills-of-a-data-scientist>

<http://www.dataists.com/2010/09/a-taxonomy-of-data-science/>

<http://blog.ericnovik.com/2012/08/09/a-better-way-to-learn-applied-statistics-got-zat/>

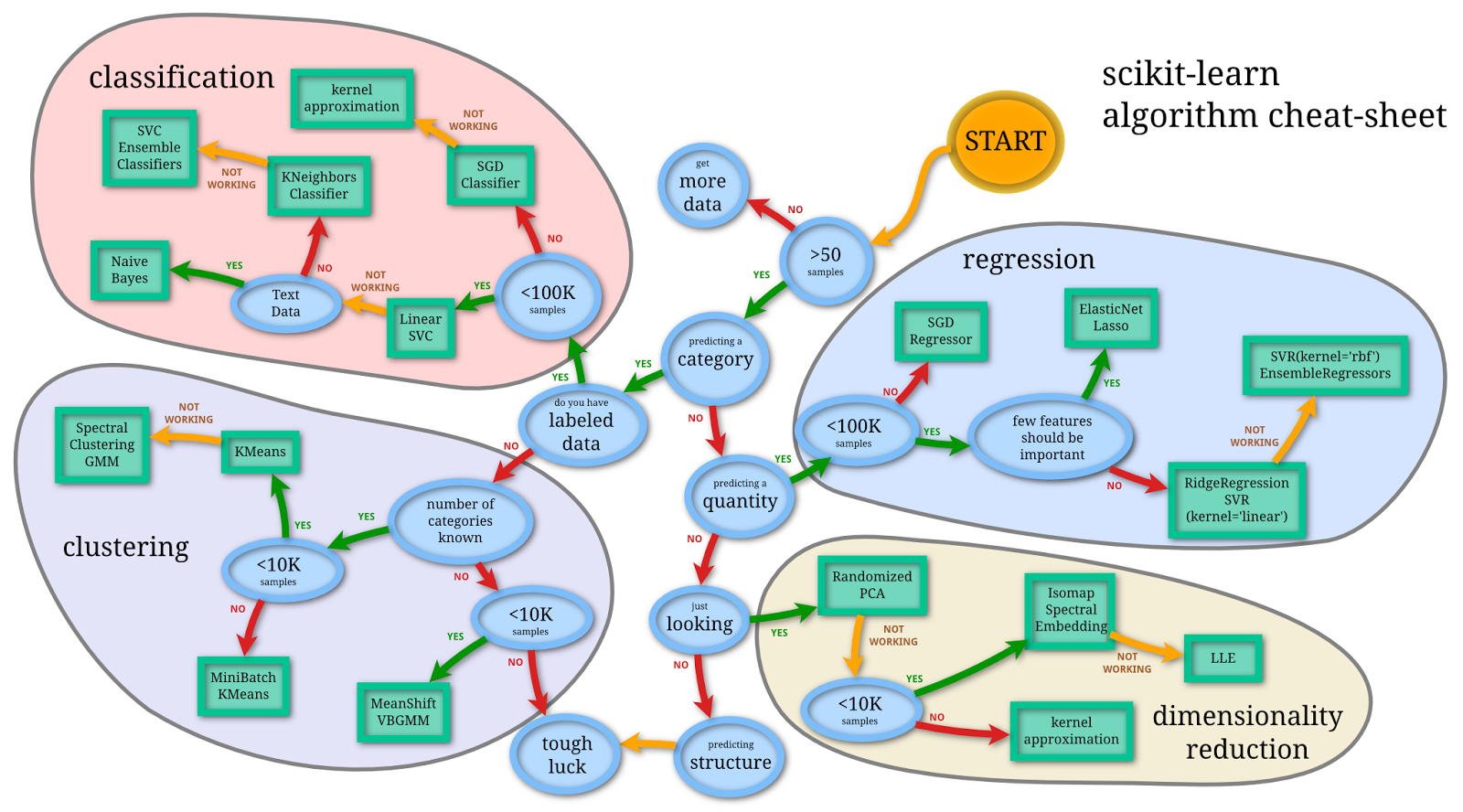
[**Feature Engineering**: How to transform variables and create new ones?](http://www.analyticsvidhya.com/blog/2015/03/feature-engineering-variable-transformation-creation/?utm_content=bufferee3ef&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer)

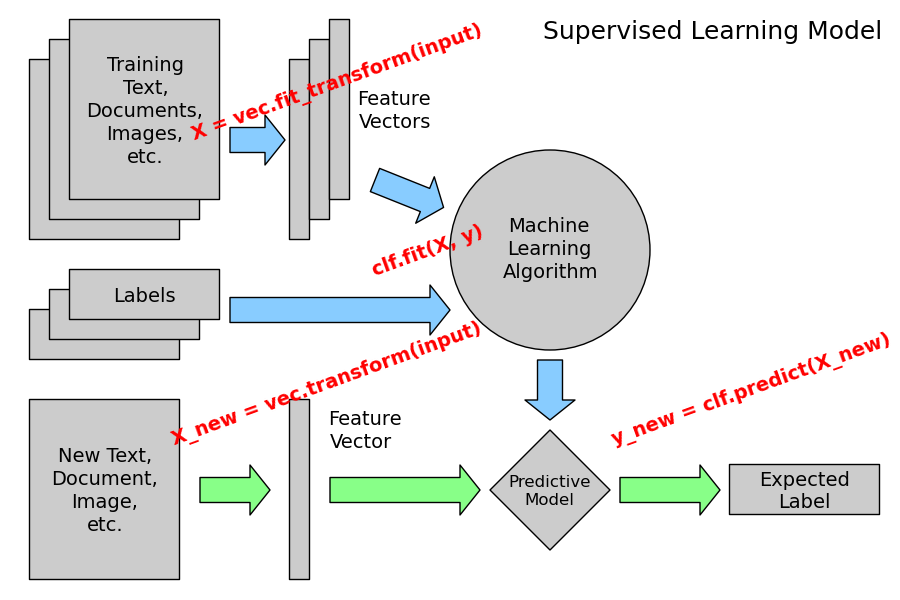
<https://github.com/jonathan-bower/DataScienceResources#machine-learning>

<https://github.com/josephmisiti/awesome-machine-learning#python>

[**Support Vector Machines and Kernels for Computational Biology**](http://www.ploscollections.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1000173) - Asa Ben-Hur, , Gunnar Rätsch. **PLOS Computational Biology:** | info:doi/10.1371/journal.pcbi.1000173

And countless more….



[](http://www.astroml.org/sklearn_tutorial/auto_examples/plot_ML_flow_chart.html)

*Overview of supervised Learning with scikit-learn*

~3 Hour tutorial on youtube. There are many,. <https://www.youtube.com/watch?v=HjAB45qsx_c> (Haven’t watched it myself)

