

A composite background image featuring a snowy mountain range. In the foreground, there are wind turbines on a rocky outcrop. To the left, a large ship and an offshore oil platform are visible in the water. In the distance, a city skyline is partially visible through the haze. The sky is blue with scattered clouds, and a satellite is seen in the upper right corner.

BIG DATA, DATA SCIENCE, & MACHINE LEARNING

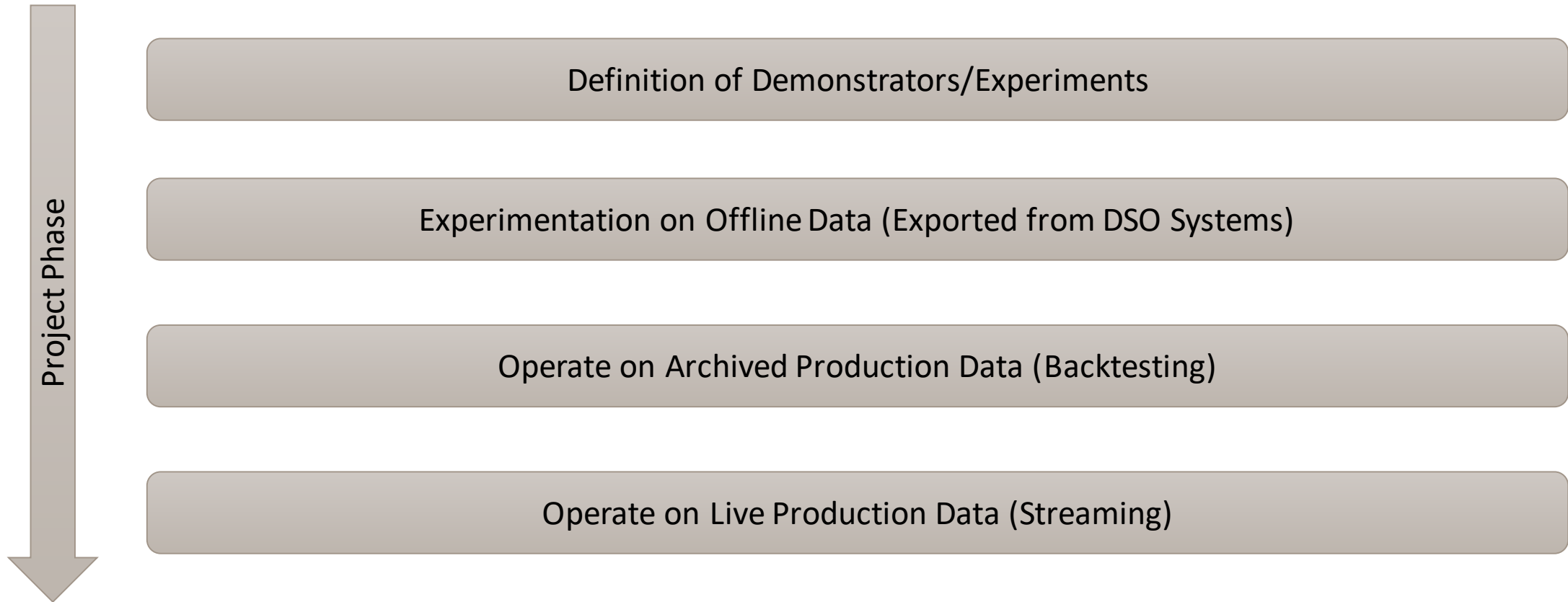
Volker Hoffmann, volker.hoffmann@sintef.no

Smartgridsenterets Webinar, 26-04-2018

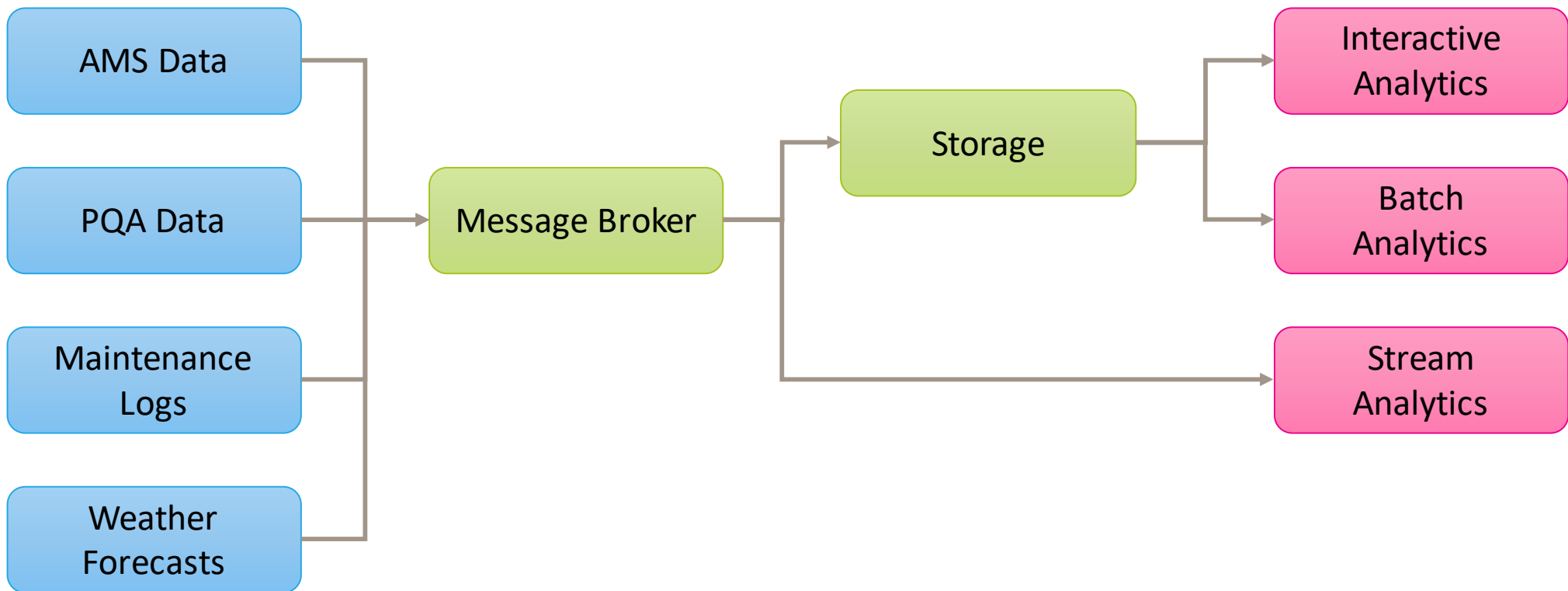
Outline

1. From Experiment to Production
2. Deployed Analytics Infrastructures
3. Data Science & Machine Learning
 1. Tools: Python & Jupyter
4. Reference Slides:
 1. Machine Learning Zoo // Links // Data Science, Optimization // Big Data Landscape

From Experiment to Production



Deployed Analytics Infrastructure



Data Science: Tools

 **ANACONDA DISTRIBUTION**
Most Trusted Distribution for Data Science

ANACONDA NAVIGATOR

Desktop Portal to Data Science

ANACONDA PROJECT

Portable Data Science Encapsulation

DATA SCIENCE LIBRARIES

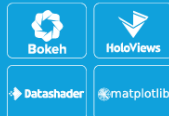
Data Science IDEs



Analytics & Scientific Computing



Visualization



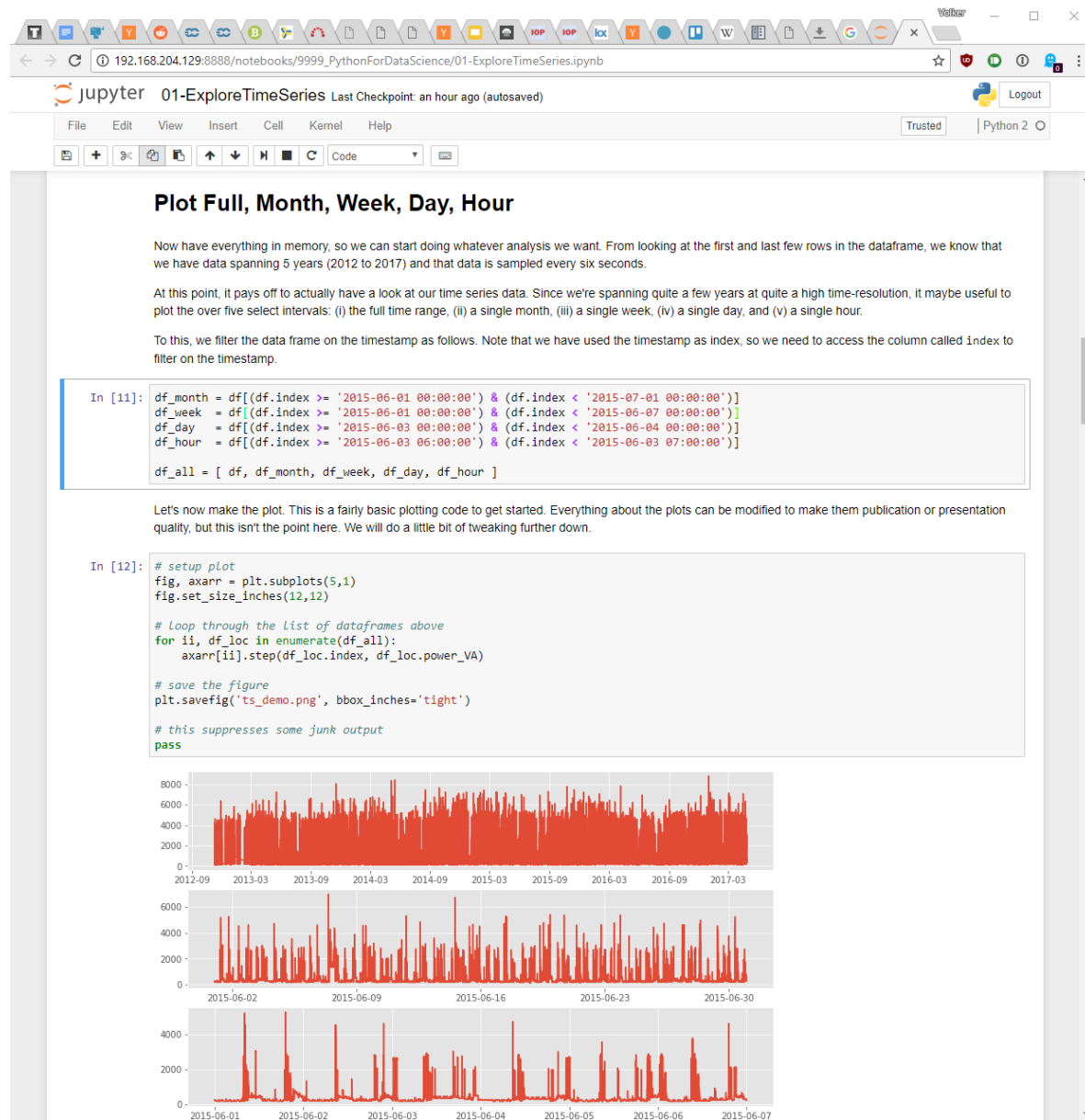
Machine Learning



...and many more!

CONDA

Data Science Package & Environment Manager



Plot Full, Month, Week, Day, Hour

Now have everything in memory, so we can start doing whatever analysis we want. From looking at the first and last few rows in the dataframe, we know that we have data spanning 5 years (2012 to 2017) and that data is sampled every six seconds.

At this point, it pays off to actually have a look at our time series data. Since we're spanning quite a few years at quite a high time-resolution, it maybe useful to plot the over five select intervals: (i) the full time range, (ii) a single month, (iii) a single week, (iv) a single day, and (v) a single hour.

To this, we filter the data frame on the timestamp as follows. Note that we have used the timestamp as index, so we need to access the column called `index` to filter on the timestamp.

```
In [11]: df_month = df[(df.index >= '2015-06-01 00:00:00') & (df.index < '2015-07-01 00:00:00')]
df_week = df[(df.index >= '2015-06-01 00:00:00') & (df.index < '2015-06-07 00:00:00')]
df_day = df[(df.index >= '2015-06-03 00:00:00') & (df.index < '2015-06-04 00:00:00')]
df_hour = df[(df.index >= '2015-06-03 06:00:00') & (df.index < '2015-06-03 07:00:00')]

df_all = [ df, df_month, df_week, df_day, df_hour ]
```

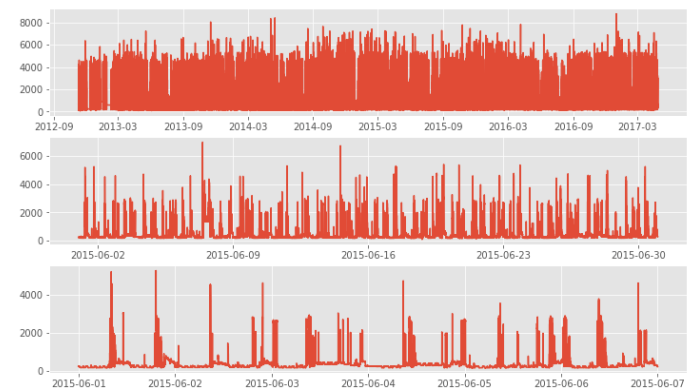
Let's now make the plot. This is a fairly basic plotting code to get started. Everything about the plots can be modified to make them publication or presentation quality, but this isn't the point here. We will do a little bit of tweaking further down.

```
In [12]: # setup plot
fig, axarr = plt.subplots(5,1)
fig.set_size_inches(12,12)

# Loop through the List of dataframes above
for ii, df_loc in enumerate(df_all):
    axarr[ii].step(df_loc.index, df_loc.power_VA)

# save the figure
plt.savefig('ts_demo.png', bbox_inches='tight')

# this suppresses some junk output
pass
```



Data Science & ML: Examples

DEMO 00 – Tour of Python Data Science Packages

<https://github.com/vhffm/PythonForDataScience/blob/master/00-Overview.ipynb>

DEMO 01 – Time Series Analysis

<https://github.com/vhffm/PythonForDataScience/blob/master/01-ExploreTimeSeries.ipynb>

DEMO 02 – Matched Filtering for EV Detection

<https://github.com/vhffm/PythonForDataScience/blob/master/02-MatchedFilter.ipynb>

DEMO 03 – Predictive Maintenance

[MAYBE IN THE FUTURE]

DEMO 04 – Better Time Series Forecasting

<https://github.com/vhffm/PythonForDataScience/blob/master/03-BetterForecasting.ipynb>

Recap

1. From Experiment to Production

1. Define Ideas => Interactive => Batch => Stream

2. Deployed Analytics Infrastructures

1. Sensors => Broker => Database => { Interactive | Batch } Processing
2. Sensors => Broker => Stream Processing

3. Data Science & Machine Learning

1. Tools: Python & Jupyter

4. Reference Slides:

1. Machine Learning Zoo // Links // Data Science, Optimization // Big Data Landscape

Q&A

Should we hire a Data Scientist or try to build simple models first?

In my opinion, you (or somebody on your team) should start playing with your data first. After all, it is you that knows your domain and systems best. This will help you get a feel for the questions you want to ask from your data and the problems that you can tackle.

Once you understand your data, your problems, and the methods, and would like to take things further, then it's time to hire (or get a consultant). Having some basic knowledge already will also make it easier for you to find the right person (for example, you can ask more relevant questions when you interview people).

Q&A

How does Azure compare to the Python (in Data Science)?

Azure offers a huge amount of services, so it can be tricky to find what is relevant to you. The most relevant offerings here are **Machine Learning Studio** [1] and **Hosted Jupyter Notebooks** [2].

The former is a visual way of doing Machine Learning and is great for learning. It is, however, not very flexible if you need to manipulate data in way that Azure has not envisioned. While Python much has packages for (almost) everything (Wavelets, Fourier Stuff, Optimization, ...), Machine Learning studio is more limited.

Azure also offers hosted Jupyter/Python (free!) which is a great way to get started.

[1] <https://studio.azureml.net/>

[2] <https://notebooks.azure.com/>

Q&A

How much data do we need to do something useful?

"It depends, but as much as possible." A (somewhat empirical) rule of thumb is ***ten times*** as much as you have of the thing you're interested in. Consider examples:

1/ In time-series forecasting, training data should cover 10 times the forecast range.

2/ To correlate power failures with weather, have at least 10 observed failures per region (and "region" depends on the typical scale of weather conditions.)

3/ Machine learning algorithms have parameters that are learned. To do well, we need ten times as many samples as parameters. For some methods, this means 40 or 50 samples. For others, this means thousands of samples.

We can also have too much data. In this case, you need to rethink visualization (e.g., plot densities instead of points) and look at dimensionality reduction (e.g., PCA).

Q&A

What industry is most sophisticated in Data Science and ML?

Publicly, the most sophisticated uses are in speech recognition (Alexa, Siri), profiling of user behavior and targeted advertisement (Facebook, Google), image recognition (Facebook, Google, Microsoft).

Out of the public eye, there's a lot happening in different industries. Manufacturing uses image recognition for quality control of products, the fisheries industry has done some work on fish swarm tracking/prediction, and automated processing of satellite images in Earth Observation (useful in disaster recovery, econometrics, or geohazard forecasts) is a ramping up. Another heavy use is in predictive maintenance in asset-intensive (oil and gas) and safety-critical (aircraft turbines) industries.

References

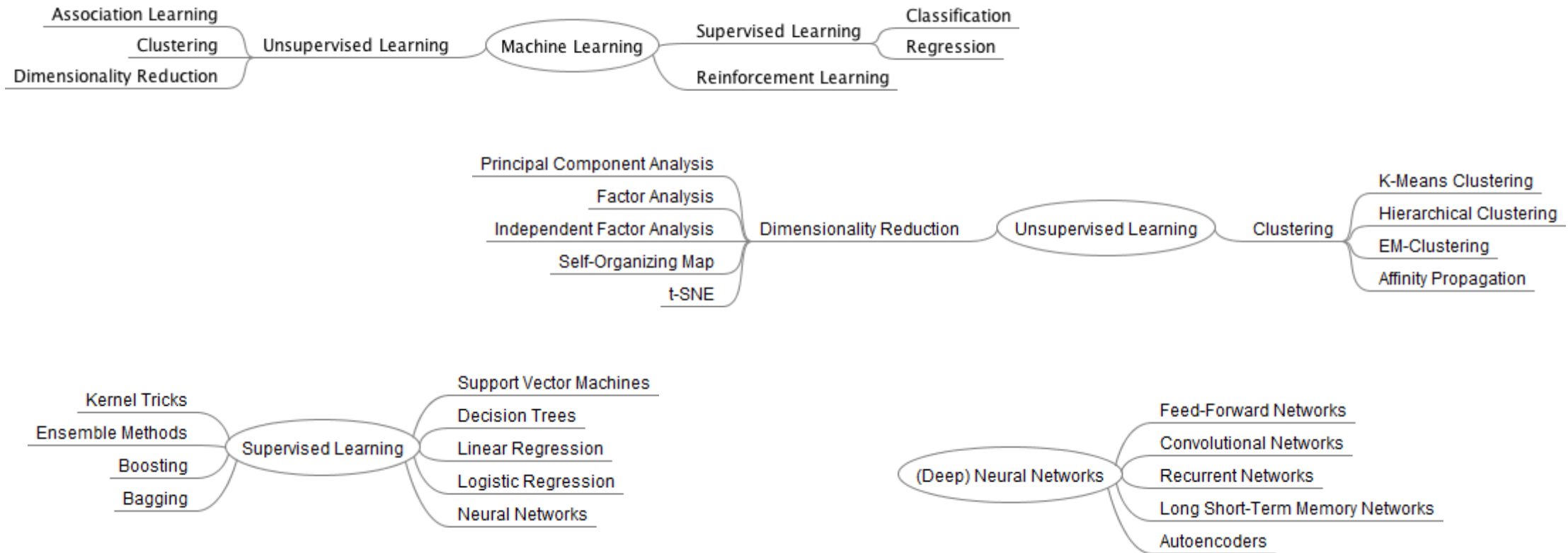
Links

- Anaconda Python: <https://www.anaconda.com/distribution/>
- Jupyter Notebook: <http://jupyter.org/>
- Demo Notebooks: <https://github.com/vhffm/PythonForDataScience/>
- Important Python Packages:
 - Pandas: <https://pandas.pydata.org/>
 - Scikit-Learn: <http://scikit-learn.org/>
 - Matplotlib: <https://matplotlib.org/>
 - Numpy/Scipy: <http://www.numpy.org/> <https://www.scipy.org/>
 - Scikit-Image: <http://scikit-image.org/>

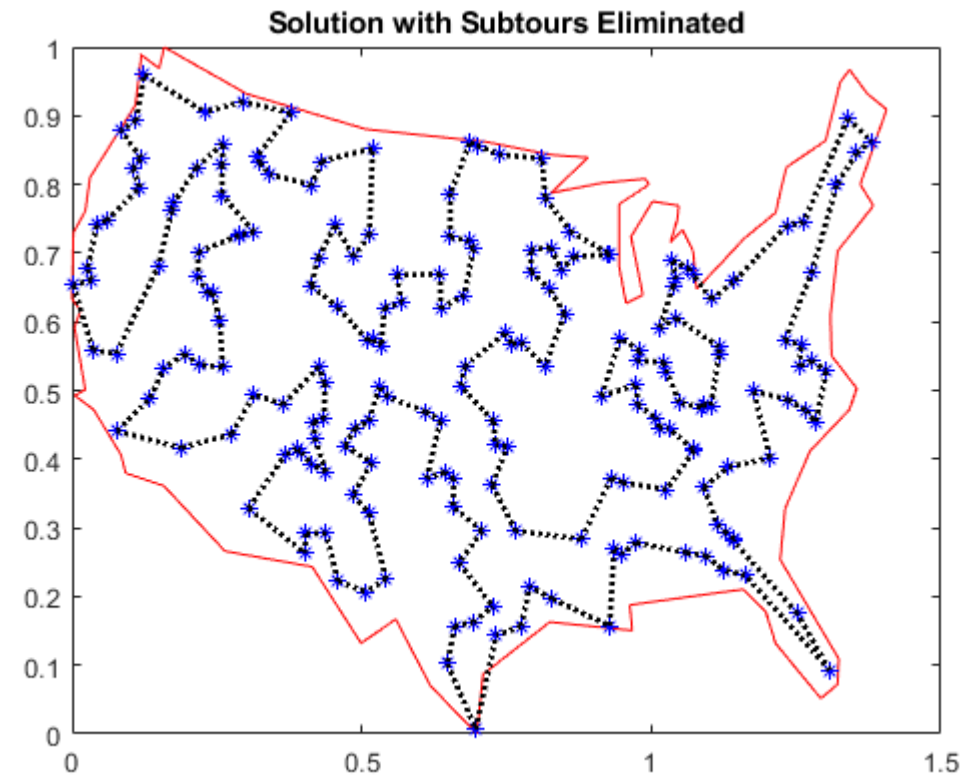
Good Books

- Python for Data Analysis
 - <http://wesmckinney.com/pages/book.html>
- An Introduction to Statistical Learning
 - <http://www-bcf.usc.edu/~gareth/ISL/>
 - Public PDF: <http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Seventh%20Printing.pdf>
- The Elements of Statistical Learning
 - <https://web.stanford.edu/~hastie/ElemStatLearn/>
 - Public PDF: <https://web.stanford.edu/~hastie/ElemStatLearn/download.html>

Machine Learning (Petting) Zoo

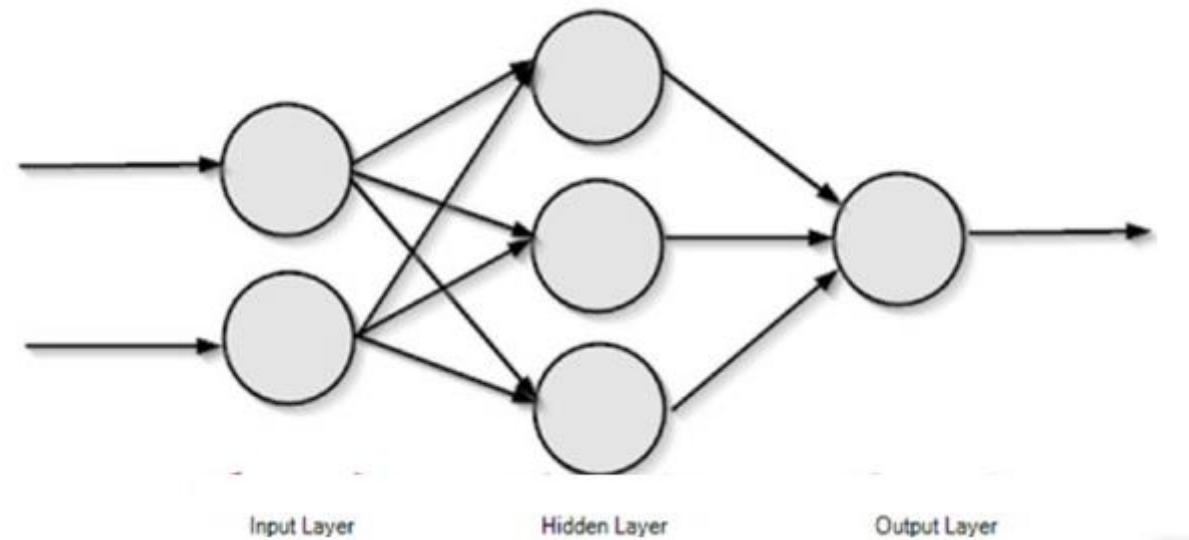
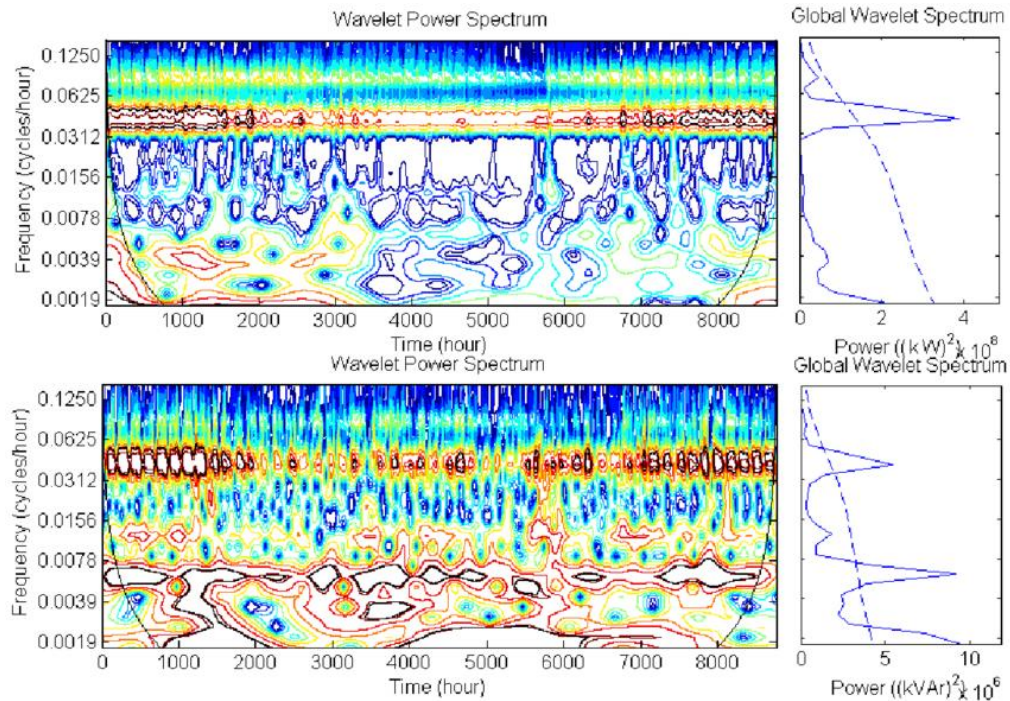


Time Series Analysis & Optimization



<https://se.mathworks.com/help/optim/examples/travelling-salesman-problem.html>

Signal Processing & Machine Learning

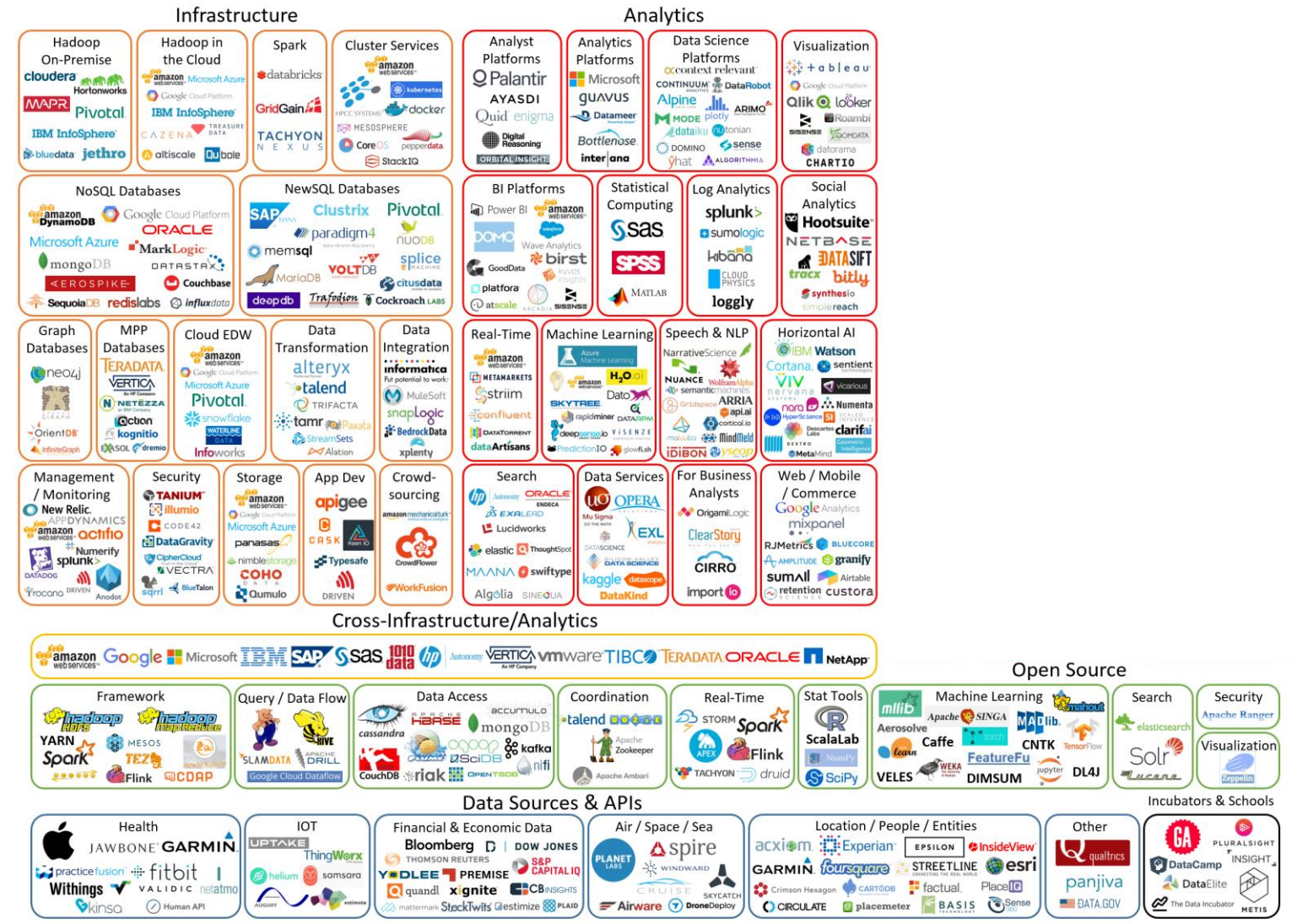


Avdakovic & Bosovic (2017)

<https://dzone.com/articles/feed-forward-neural-network-with-mxnet>

Big Data Processing

Big Data Landscape 2016 (Version 3.0)



Last Updated 3/23/2016

© Matt Turck (@mattturck), Jim Hao (@jimrho), & FirstMark Capital (@firstmarkcap)

FIRSTMARK



Teknologi for et bedre samfunn