

*Engineering for everyday world*

**AceEngineer**

**DATA SCIENCE**

**Technical Knowledge**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15-Nov-2012 | 01 | Added new features | | | | | VA | ?? | VA |
| DATE | REV | DESCRIPTION | | | | | ORIG | CHK | APPR |
| DOCUMENT CONTROL NO | | Project | Type | Area | Client | - | - | Sequence | Revision |
|  | **DS** | **-** | **-** | **-** | **-** | **0001** | **01** |

**Revision History:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REV | DATE | DESCRIPTION | ORIG | CHK | APPR |
| 01 | 31 Dec 10 | New Document | VA |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Change Log**

|  |  |  |
| --- | --- | --- |
| REV | SECTION | CHANGE DESCRIPTION |
|  |  |  |
|  |  |  |
|  |  |  |

Document Holds

|  |  |
| --- | --- |
| Hold | DESCRIPTION |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

CONTENTS

[1 Introduction 7](#_Toc80443666)

[1.1 Document Purpose 7](#_Toc80443667)

[1.2 Document Scope 7](#_Toc80443668)

[1.3 References 7](#_Toc80443669)

[1.4 Summary 8](#_Toc80443670)

[2 Technical Terms 9](#_Toc80443671)

[2.1 Disruptive Technologies 10](#_Toc80443672)

[2.2 Deep Learning 10](#_Toc80443673)

[2.3 Data Science vs. Machine Learning vs. Artificial Intelligence 11](#_Toc80443674)

[2.4 Open Platform Communications (OPC) 11](#_Toc80443675)

[3 Pipeline data 12](#_Toc80443676)

[4 Math for Data Science 12](#_Toc80443677)

[4.1 Probability and Statistics 13](#_Toc80443678)

[4.2 Distributions 16](#_Toc80443679)

[5 Exploratory Data Analysis (EDA) 16](#_Toc80443680)

[6 Business Analytics 17](#_Toc80443681)

[6.1 Text Mining 20](#_Toc80443682)

[7 Machine Learning Algorithms 21](#_Toc80443683)

[7.1 Basics 22](#_Toc80443684)

[7.1.1 Evaluation 24](#_Toc80443685)

[7.2 Basic Symbols: 24](#_Toc80443686)

[7.3 Regression 24](#_Toc80443687)

[7.3.1 Linear Regression (2-dimensional classification) 24](#_Toc80443688)

[7.4 Logistic Regression 25](#_Toc80443689)

[7.4.1 Project 25](#_Toc80443690)

[7.4.2 Cross Entropy (CE) Method: 26](#_Toc80443691)

[7.4.3 Weights (ML Training) 27](#_Toc80443692)

[7.4.4 Donut Problem 31](#_Toc80443693)

[7.5 Bayesian Statistics 31](#_Toc80443694)

[7.6 Fuzzy Logic 31](#_Toc80443695)

[7.7 Clustering 32](#_Toc80443696)

[7.7.1 K-means Clustering 32](#_Toc80443697)

[7.8 Random Forest 33](#_Toc80443698)

[7.8.1 Algorithms 33](#_Toc80443699)

[7.8.2 Properties 33](#_Toc80443700)

[7.9 Predictions 33](#_Toc80443701)

[7.10 Machine Learning – Practical Issues 34](#_Toc80443702)

[7.11 Big Data Statistical Methods 35](#_Toc80443703)

[7.12 Anomaly Detection 36](#_Toc80443704)

[7.13 References 37](#_Toc80443705)

[8 Scikit-learn ML Examples 39](#_Toc80443706)

[8.1 Introduction 39](#_Toc80443707)

[8.2 Scikit-Learn ML Summary 40](#_Toc80443708)

[8.3 Regression 41](#_Toc80443709)

[8.3.1 Linear Regression 41](#_Toc80443710)

[8.3.2 Logistic Regression 41](#_Toc80443711)

[8.4 Decision Trees 41](#_Toc80443712)

[8.4.1 Ex: Decision Tree Regression 41](#_Toc80443713)

[8.4.2 Ex: Decision Tree Classifier 42](#_Toc80443714)

[8.4.3 Ex: Decision Tree Multi-Output 42](#_Toc80443715)

[8.5 k-NN 42](#_Toc80443716)

[8.5.1 Ex: k-NN Regression 43](#_Toc80443717)

[8.5.2 Ex: k-NN Classification 44](#_Toc80443718)

[8.5.3 Ex: K-NN Classification Nearest Centroid classification 45](#_Toc80443719)

[8.5.4 Ex: K-NN Classification Nearest Centroid classification 45](#_Toc80443720)

[8.6 Naïve Bayes 45](#_Toc80443721)

[8.7 Neural Networks 46](#_Toc80443722)

[8.8 Model Selection & Performance Metrics 46](#_Toc80443723)

[8.9 References 46](#_Toc80443724)

[9 Deep Learning 47](#_Toc80443725)

[9.1 Machine Set-up 47](#_Toc80443726)

[9.2 Stan 48](#_Toc80443727)

[9.2.1 References 48](#_Toc80443728)

[9.3 BUGS 49](#_Toc80443729)

[9.4 Jags 49](#_Toc80443730)

[9.5 Nimble 49](#_Toc80443731)

[10 Time Series Analysis 49](#_Toc80443732)

[10.1 Introduction 50](#_Toc80443733)

[10.2 Time Series Modelling 50](#_Toc80443734)

[10.3 Time Series Exploration 50](#_Toc80443735)

[10.4 Time Series Models 50](#_Toc80443736)

[10.5 Arima vs. FB Prophet 51](#_Toc80443737)

[10.5.1 References 51](#_Toc80443738)

[11 Data Ingestion 52](#_Toc80443739)

[11.1 Data Cleansing 52](#_Toc80443740)

[12 Data Processing 52](#_Toc80443741)

[12.1 Time series analysis 52](#_Toc80443742)

[12.2 Prediction tools 52](#_Toc80443743)

[12.3 References 52](#_Toc80443744)

[13 Visualization 53](#_Toc80443745)

[13.1 How and What to Show 53](#_Toc80443746)

[13.1.1 Simple Visualizations 53](#_Toc80443747)

[13.2 Visualization Programs 53](#_Toc80443748)

[13.3 Spotfire 54](#_Toc80443749)

[13.3.1 Example 54](#_Toc80443750)

[13.3.2 General Working 56](#_Toc80443751)

[13.4 Python Bokeh 58](#_Toc80443752)

[13.4.1 Deliver Reports 58](#_Toc80443753)

[14 GIS 59](#_Toc80443754)

[15 Natural Language Processing (NLP) 59](#_Toc80443755)

[15.1 Packages 59](#_Toc80443756)

[15.1.1 eNLP by Equinor 59](#_Toc80443757)

[16 References 59](#_Toc80443758)

[Appendix 1.0 - Courses 59](#_Toc80443759)

[1.1 Courses Finished 59](#_Toc80443760)

[1.1.1 Udemy Course List 60](#_Toc80443761)

[Appendix 2.0 - Spark 63](#_Toc80443762)

[2.1 Typical Data Process 63](#_Toc80443763)

[Appendix 3.0 – Maven 63](#_Toc80443764)

[3.1 Amazon S3 64](#_Toc80443765)

[3.2 Latency 64](#_Toc80443766)

[Appendix 4.0 - OTC 2017 Big Data Lessons 65](#_Toc80443767)

[4.1 Structured vs. Unstructured data 65](#_Toc80443768)

[4.2 Engineering Algorithms 65](#_Toc80443769)

[4.3 Standardization Solutions 65](#_Toc80443770)

[4.4 Asset Framework 65](#_Toc80443771)

[4.5 Competition 66](#_Toc80443772)

[Appendix 5.0 – Data Science 68](#_Toc80443773)

[5.1 Data science key learnings are: 68](#_Toc80443774)

[Appendix 6.0 - Prognostics 69](#_Toc80443775)

[Appendix 7.0 - NewField 69](#_Toc80443776)

[Appendix 8.0 – Deep Learning Course 69](#_Toc80443777)

[8.1 Binary classification to Multi-class Classification 69](#_Toc80443778)

[8.2 Supervised Learning 70](#_Toc80443779)

[8.3 Back Propagation 71](#_Toc80443780)

[8.4 Course Project 72](#_Toc80443781)

[8.4.1 Engineer at a Inventory company 72](#_Toc80443782)

[8.4.2 Facial Expression Recognition 72](#_Toc80443783)

[Appendix 9.0 - Digital Twins 72](#_Toc80443784)

[9.1 DYSYS 74](#_Toc80443785)

[Appendix 10.0 - R 74](#_Toc80443786)

[Appendix 11.0 - Finance Data 75](#_Toc80443787)

[Appendix 12.0 - Chemistry 75](#_Toc80443788)

[Appendix 13.0 – Sports Case Studies 75](#_Toc80443789)

[Appendix 14.0 - Python for Data Science 75](#_Toc80443790)

[Appendix 15.0 Python and Open Server 76](#_Toc80443791)

[Appendix 16.0 R Studio 76](#_Toc80443792)

# Introduction

Data science is a leading science applied in industrial world during the early 21st century. After designing, engineering and operating equipment, the stakeholders expect to perform data science. The data science objective is to collect data and predict system failures and also help design future systems. This is also the first step for further mechanization and automation.

This document describes the data science basics and latest technologies.

## Document Purpose

## Document Scope

## References

<https://paperswithcode.com/>

https://www.linkedin.com/pulse/100-open-source-big-data-architecture-papers-anil-madan?trk=hp-feed-article-title-like Big Data 100 Big Data papers

https://www.coursera.org/specializations/jhu-data-science Data science paid course

https://www.youtube.com/watch?v=WpxK\_\_SK2a0&index=2&list=PLD0F06AA0D2E8FFBA Machine learning videos

https://www.epmag.com/learning-leverage-artificial-intelligence-oil-gas-1677511 AI in O&G

https://www.datacamp.com/community/tutorials/time-series-analysis-tutorial Time series analysis using Phython

https://www.datacamp.com/community/blog/mobile-data-science Bite size data science and ML lessons

https://www.datacamp.com/community/tutorials/python-string-tutorial Python Strings

https://www.datacamp.com/community/tutorials/active-learning Curious AI Algorithms

https://www.datacamp.com/community/tutorials/google-cloud-data-science Google Data Science and VM

<https://ourworldindata.org/>

Data Engineering Cookbook

<https://github.com/andkret/Cookbook>

https://chrisalbon.com/ Notes on Data Science and Artificial Intelligence

Interpretable Machine Learning

https://christophm.github.io/interpretable-ml-book/storytime.html

Introduction to Machine Learning

<https://openlearninglibrary.mit.edu/courses/course-v1:MITx+6.036+1T2019/about>

<https://github.com/andkret/cookbook>

Artificial Intelligence

“Artificial Intelligence, a New Synthesis" Book by Nils Nilsson.

<https://www.elsevier.com/books/artificial-intelligence-a-new-synthesis/nilsson/978-1-55860-535-0>

Optimizers in Python. Comparison of Optimizers.

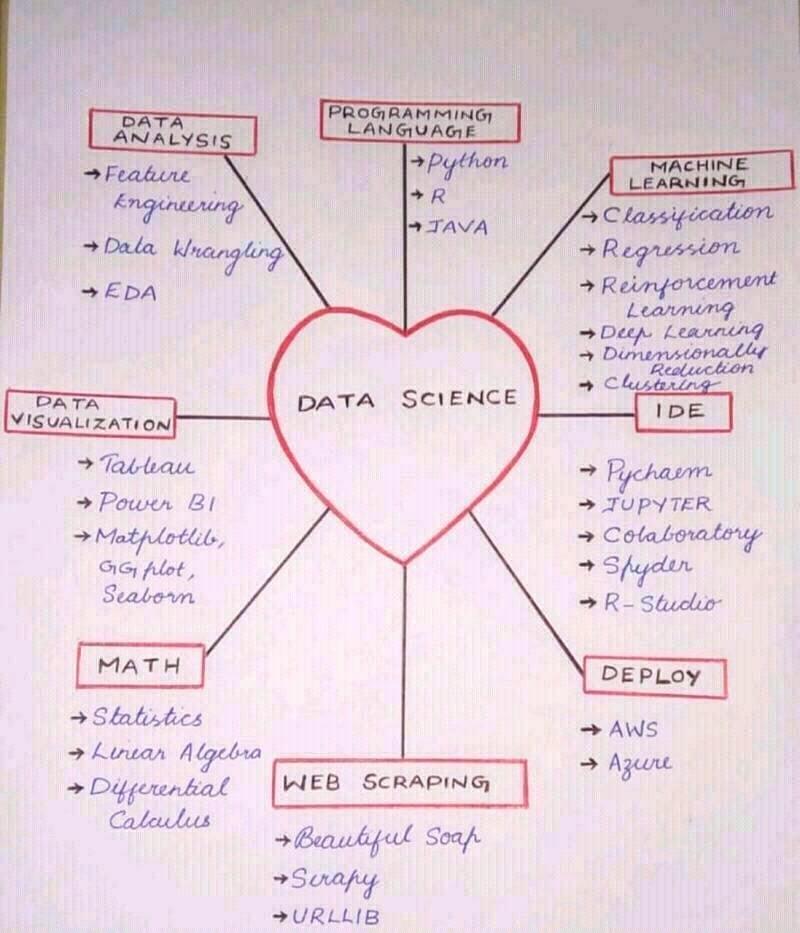
<https://www.microprediction.com/blog/robust-optimization>

<https://www.microprediction.com/blog/optimize>

<https://towardsdatascience.com/practical-machine-learning-tutorial-part-4-model-evaluation-2-764d69f792a5>

## Summary

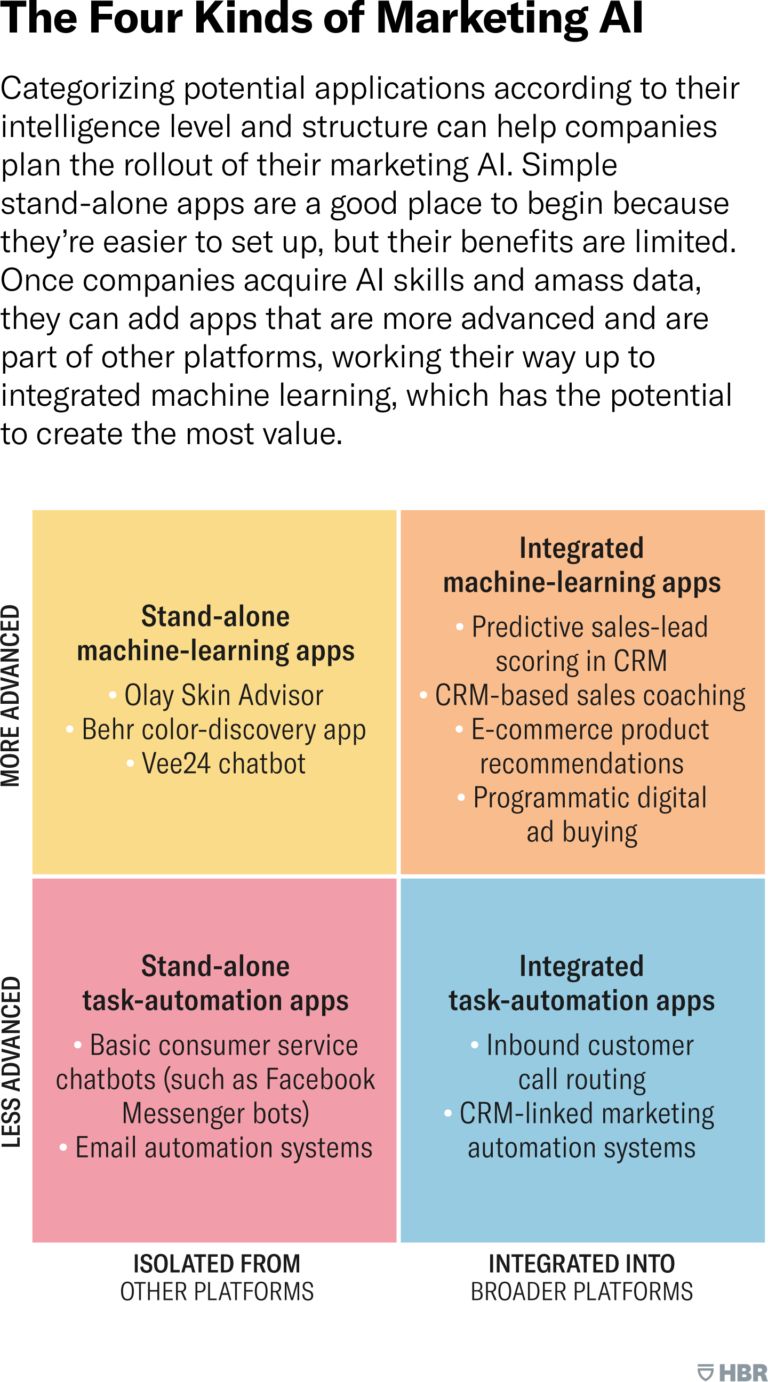
Technologies



Youtube video series

<https://www.youtube.com/playlist?list=PL3N9eeOlCrP6IjkyExZW9oZFwt-A1r0qB>

<https://www.youtube.com/c/AIEngineeringLife>



## Simple Applications

https://github.com/Microsoft/ailab/tree/master/Sketch2Code

Multitask Prediction

<https://github.com/google/ehr-predictions>

<https://academic.oup.com/jamia/article/28/9/1936/6307184#265336640>

<https://ai.googleblog.com/2021/07/multi-task-prediction-of-organ.html>

Deep Learning

<https://github.com/blueberrymusic>

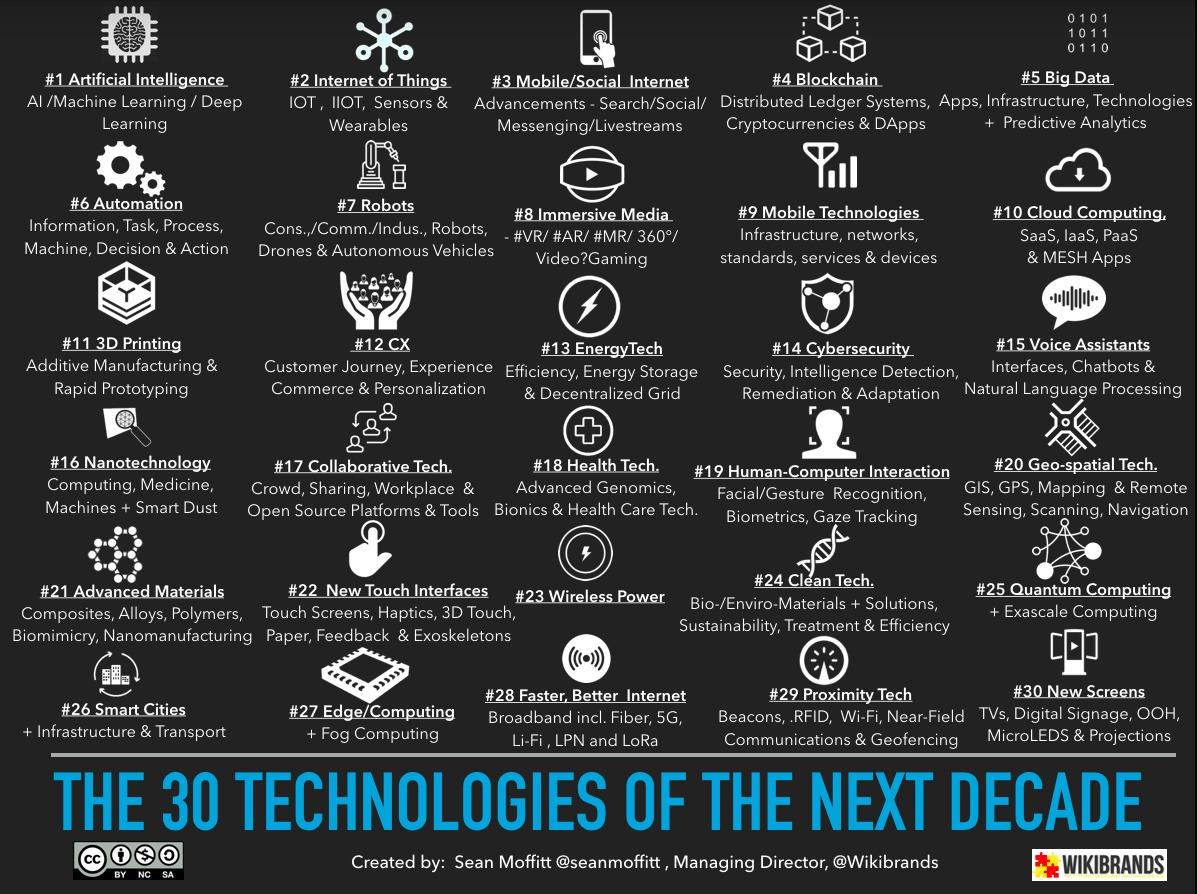
Open Parking Alert

<https://gist.github.com/ageitgey/dee64e0e21d7c1eb3fd20a22e8aebd4f#file-open_parking_alert-py>

# Technical Terms

## Disruptive Technologies

The following disruptive technologies will shape the future of data science



## Deep Learning

Involves optimization and learning

Time Series

IoT – Internet of Things

KPIs – Key Performance Indicators

SCADA : Supervisory control and Data Acquisition

HMI – Human Machine Interface

BMI -

PREDIX (GE System)

Accelerators

HDP Platform (Hortonworks Data Platform)

HIVE and HBASE?

Non value tasks

NIFI

On the Cloud

Zepplien (General Spark Code)

HortonWorks : Hadoop flavour Data Processing and Storage

## Data Science vs. Machine Learning vs. Artificial Intelligence

http://varianceexplained.org/r/ds-ml-ai/

<https://docs.google.com/presentation/d/1kSuQyW5DTnkVaZEjGYCkfOxvzCqGEFzWBy4e9Uedd9k/mobilepresent?slide=id.g168a3288f7_0_58>

<https://cloud.google.com/products/ai/ml-comic-1/>

## Open Platform Communications (OPC)

OPC

<https://en.wikipedia.org/wiki/Open_Platform_Communications>

<https://en.wikipedia.org/wiki/OPC_Foundation>

<http://www.opcdatahub.com/WhatIsOPC.html>

<https://en.wikipedia.org/wiki/OPC_Data_Access>

Kepware

Oil and Gas

Production: Geology; Completion; Production variables

Apache NIFI

# Pipeline data

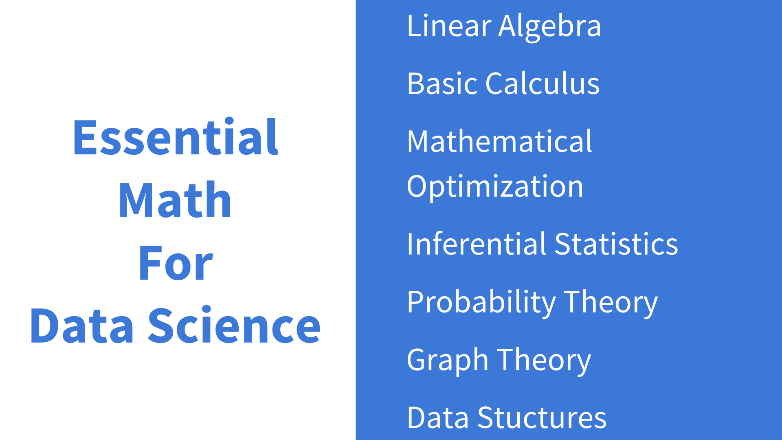
<https://www.data.bsee.gov/Pipeline/PipelinePermits/Default.aspx>

- Enter "Segment number"

- Click "Submit Query"

- Get the MAOP and other required associated data.

# Math for Data Science



<https://www.youtube.com/watch?v=kjBOesZCoqc&list=PL0-GT3co4r2y2YErbmuJw2L5tW4Ew2O5B>

<https://www.youtube.com/watch?v=WUvTyaaNkzM&list=PL0-GT3co4r2wlh6UHTUeQsrf3mlS2lk6x>

<https://www.youtube.com/channel/UCkL2HNDjyhrT6hgWjikmQAg/videos>

<https://www.analyticsvidhya.com/blog/2017/01/comprehensive-practical-guide-inferential-statistics-data-science/>

<https://www.analyticsvidhya.com/blog/2014/04/probability-action-monty-halls-money-show/>

<https://medium.com/basecs/a-gentle-introduction-to-graph-theory-77969829ead8>

<https://towardsdatascience.com/a-data-scientists-guide-to-data-structures-algorithms-1176395015a0>

## Probability and Statistics

https://qz-com.cdn.ampproject.org/c/s/qz.com/1206229/this-is-the-best-book-for-learning-modern-statistics-its-free/amp/ Modern Statistics

Why Statistics?

- It is, with Calculus and Algebra, the core of every Machine and Deep Learning we use for Data Science.

- We need to evaluate and validate our models, make predictions, calculate probabilities and infer possible scenarios, and all of that is possible through Statistics.

- The art of plotting statistical distributions, trends, time series, and make stunning visualizations for the business meetings you'll have to explain your results needs a good statistical base.

There are more reasons but let's see how you can get started with this great topic:

1. Coursera:

https://lnkd.in/gdAZUhQ

https://lnkd.in/gmEsH3a

2. DataCamp:

https://lnkd.in/gNYHAZh

https://lnkd.in/g3PY3Gd

https://lnkd.in/gCu48E9

3. Stanford:

https://lnkd.in/g2qvU2r

https://lnkd.in/gKyAqaK

4. edX:

https://lnkd.in/grDgukz

https://lnkd.in/gKfRRKN

5. Free Books:

https://lnkd.in/ge2MwqX

https://lnkd.in/gyYeeSk

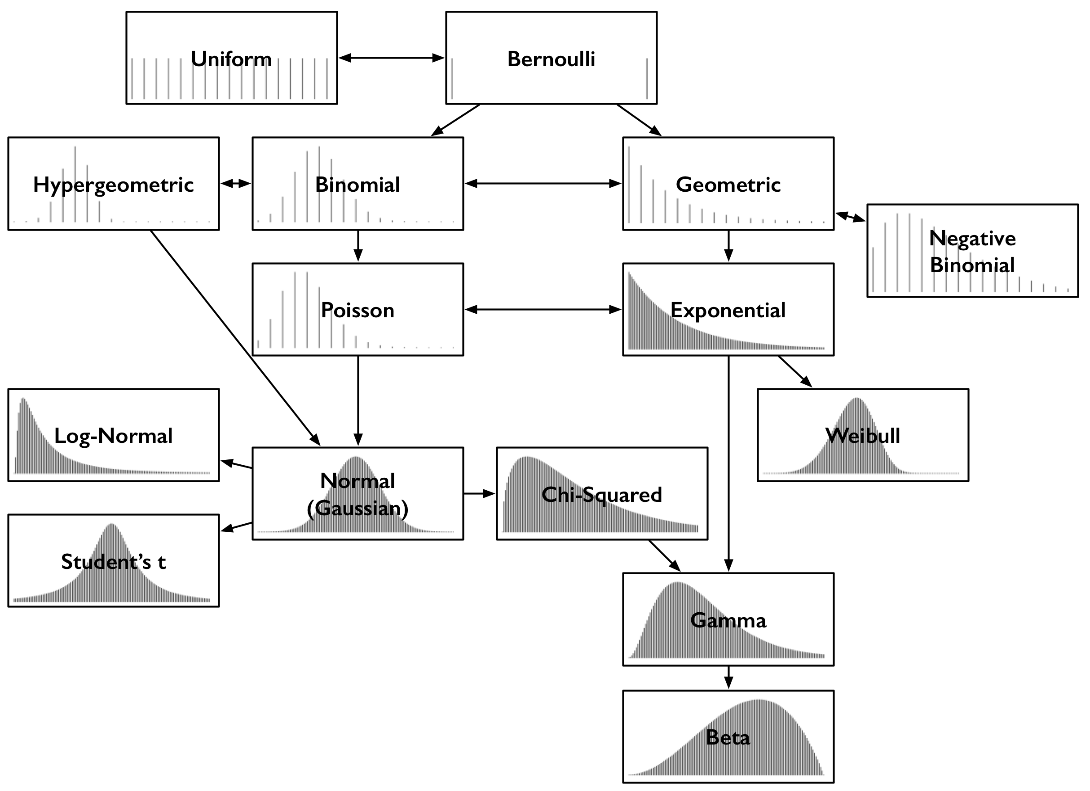
https://lnkd.in/gDwbnGt

https://www.datacamp.com/courses/introduction-to-time-series-analysis-in-python?utm\_source=customerio&utm\_medium=email&utm\_campaign=course\_4267 Free Time Series Analysis

Bayesian and Spatio-Temporal Statistical Modelling



## Distributions



https://medium.com/@srowen/common-probability-distributions-347e6b945ce4

Probability

# Exploratory Data Analysis (EDA)

EDA techniques are simple, efficient, and powerful for the routine testing of underlying assumptions:

* run sequence plot (Yi versus i)
* lag plot (Yi versus Yi-1)
* histogram (counts versus subgroups of Y)
* normal probability plot (ordered Y versus theoretical ordered Y)

A 4-Plot which shows fixed location, fixed variation, fixed normal
 distribution, and no outliers

A 4-plot which shows fixed location, fixed variation,
 non-randomness, a non-normal U-shaped distribution, and
 several outliers

# Business Analytics

Business analytics is systematic understanding of big data. Uses the following:

* Data
* Statistics
* Quantitative analysis
* Modelling techniques
* Fact based management

Methods used:

* Data mining
* Statistical and quantitative analysis
* Multi-variate testing (A oblique B testing)
* Predictive modelling

Objectives:

* Improved predictability
* Mathematics + Business Knowledge

Answers:

* Why something happened or is happening?
* What is the outcome if this continues over time?
* How best can the future be managed?
* Visualize the data and make the right decisions

Evolution of BA: BA 1.0; 2.0 and 3.0 (Just started)

Types of BA:

* Descriptive Analytics: Past information -> plan for future (Post-mortem analysis)
* Diagnostic Analytics: When, where, how and why a particular crisis occurs. Correlation analysis, Hypothesis, Analysis of Variance (ANOVA) and control charts
* Predictive Analytics: Convert data into information that is actionable. Statistical techniques to forecast future trends. Develop patterns. Assess risk associated with a condition.
* Prescriptive Analytics: More than data insights and feasible actions. Includes machine learning and help predict why a condition happens. Statistics, mathematics and business rules

Data Errors come from:

* Data entry errors : Bad data is inevitable. Minimizing errors is the objective.
* Processing inefficiencies

Having the right data at the right time

Data integration

* Involves transfer of data from one location to another
  + Extract
  + Transform
  + Load
  + The above 3 sub-steps can be named as ETL and can be used for clean up
* Ability to perform change data capture

Importance of Data

* Understand data
* How data is put together
* Unlocking data prospects
* Growth in data volume is exponential
  + Where to start
  + Where is the data
  + In what format
  + What all does it include
* Traditional data handling techniques are inadequate
* Creating value from big data
  + Makes data transparent and usable
  + Detailed information on trends
  + Target products for narrow customer groups
  + Forecast next services required (eg. Pro-active maintenance or preventive maintenance)
  + Forecase next products
* Advantages of business analytics
  + Effective analysis of all data
  + Monitoring of essential parameters that affect the business
  + Deep learning to optimal performance
  + Generate insight into customers
* Real-time Analytics:
  + See what is happening
  + Plan or forecast future events (or) predictive analytics
* Key features of good Analytics solutions
  + Modular
  + Ease of use
* Challenges for Business Analytics:
  + Budgets
  + Knowledge
  + Lack of executive advocacy
  + Data availability, clean up and retention
* Data structures
  + Structured
  + Unstructured
  + Semi-structured
* Data Cleaning:
  + Clean
  + Normalization
  + Transformation
  + Feature extraction
  + Selection
* Data Maturity
  + No useful data
  + Big Data
  + Right Data
  + Predictive data
  + Strategic
* Business analytics vs. business intelligence
  + Valuable new insights

Social Media Analytics

## Text Mining

Structured and unstructured data

Look for key words. For example: Drilling can have the following:

* WOB (Weight on Bit)
* BHA
* Torque
* Depth
* Formation integrity test (FIT)

Text (Data) should be pre-processed.

* Homogenization of Cap
* Remove punctuation
* Sign and numbers
* Lemmation or stemming of words

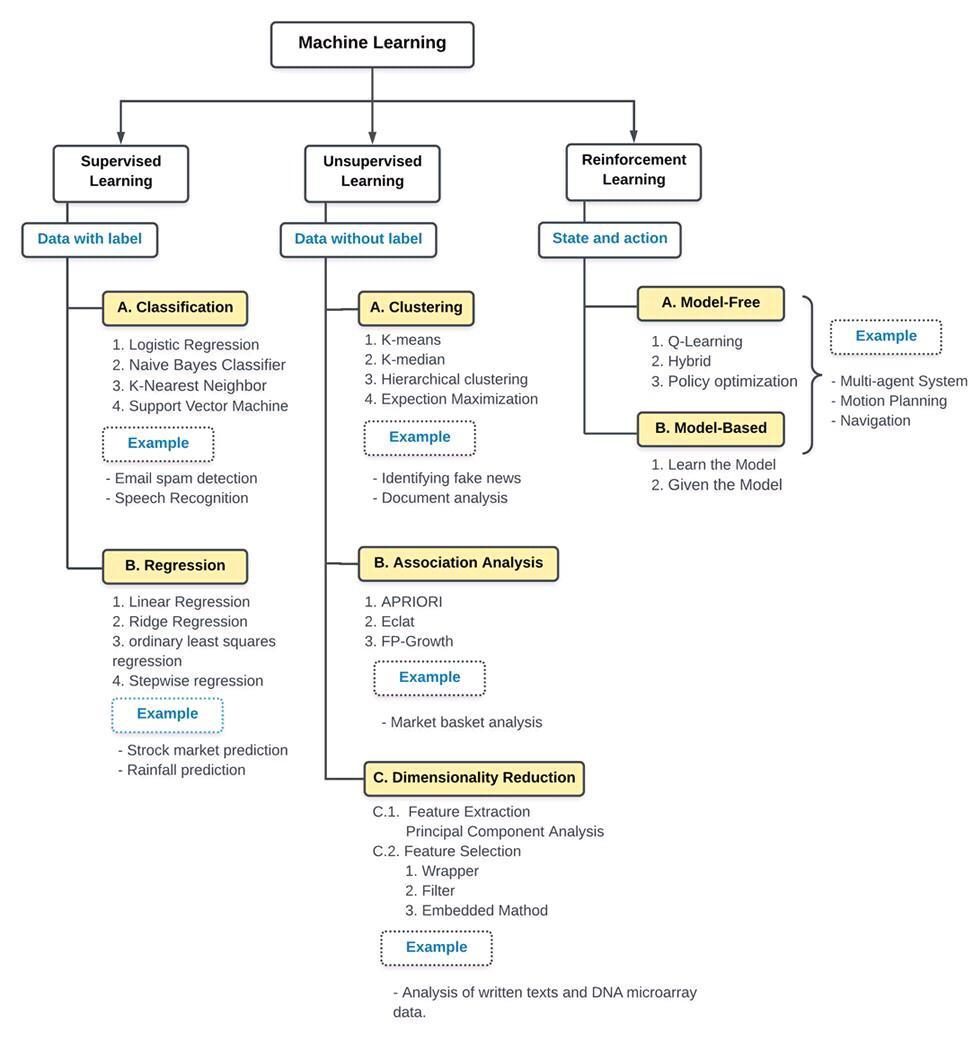
A simple Matrix of Key words found in various documents

A matrix construction of keywords with time to take into importance

Bag of words (Google)

Convert a word to vector to determine its importance.

# Machine Learning Algorithms



Broadly, there are 3 types of machine learning (ML) algorithms

* Supervised learning : Classification (into categories) and Regression (predict value of a function)
  + A target/outcome variable based on given set of predictors (variables).
  + Generate a function to map inputs to desired output
  + Training process continues until a desired accuracy in output is achieved. Training is done by altering the weights of the model.
  + Examples are: Decision tree, Random Forest, K-Nearest Neighbours (KNN), Logistic Regression etc.
* Unsupervised Learning: No data labels. Split data to find patterns
  + No target outcome
  + Used for clustering population or segmentation
  + Examples are: Apriori algorithm, K-means clustering, C-means clustering
* Reinforcement Learning
  + Trained to make decisions
  + Machine is exposed to environment where it continually learns by trial and error

Relationship of AI (Artificial Intelligence), ML (Machine learning) and DL (Deep Learning) is:

<https://semanti.ca/blog/?how-to-prepare-for-a-machine-learning-interview>

## Basics

The 7 basic steps of machine learning are:

* Gather data
* Prepare data
* Choose a model
* Training
* Evaluation
* Hyperparameter Tuning
* Prediction

Tensorflow is an open source project which is very useful for machine learning. Machine learning Sandbox:

<http://playground.tensorflow.org>

<http://www.deeplearningbook.org/> Deeplearnign Book.

The types of training are:

* Supervised
* Unsupervised
* Reinforced : learn by trail and error through reward or punishment

Introduction to Machine Learning

<https://docs.google.com/presentation/d/1kSuQyW5DTnkVaZEjGYCkfOxvzCqGEFzWBy4e9Uedd9k/preview?slide=id.g168a3288f7_0_58>

Image recognition:

<https://youtu.be/aircAruvnKk>

Sigmoid is old school.

Rectified linear Unit, ReLU(a) = max(0,a) works great for Deep nueral networks (DNN)

Gradient Decent

<https://www.youtube.com/watch?v=IHZwWFHWa-w>

<https://docs.google.com/presentation/d/1uYSM7hR8H6aNv6hGkzfS05ojQiuVOgnq5ASLpGy8IGk/preview?slide=id.p> Neural Networks

First of all, why you should learn ML if you want to be data scientist? In the end, we need to create models of the world, models that can help predict "x", or improve "y".

You'll need to deliver these insights to your company or business, and they need to be decanted for their understanding. Right now we'll only talk about ML, and in another post how to explain those models you created.

Learning in the context of ML and DL describes an automatic search process for better representations of the data you are analyzing and studying (please have this in mind, is not making a computer learn).

So let's go:

1. Coursera:

- https://lnkd.in/efN-x\_H

- https://lnkd.in/eazsGPf

- https://lnkd.in/eqV8r7V

- https://lnkd.in/e4DWpZX

2. DataCamp:

- https://lnkd.in/eR7qTrH

- https://lnkd.in/eBHM9Rr

- https://lnkd.in/e-RsiCC

3. Pluralsight:

- https://lnkd.in/e\_-gaaq

- https://lnkd.in/eQ9X3np

- https://lnkd.in/e4GT9BU

4. Kaggle:

- https://lnkd.in/ebPxrMj

- https://lnkd.in/eR4TghZ

### Evaluation

https://www.oreilly.com/ideas/ideas-on-interpreting-machine-learning

## Basic Symbols:

N = number of samples

D = number of dimensions or features per sample

X = Matrix of N by D matrix. Each row is a sample. Each column is the value of feature of 1 feature

T = Target, Nx1 Matrix. Each input X has a target. Eg. For binary classification, there are matrix of 0s and 1s.

Y : output of logistic regression model prior to output was more precise. Precision means probability that P (Y = 1 | X)

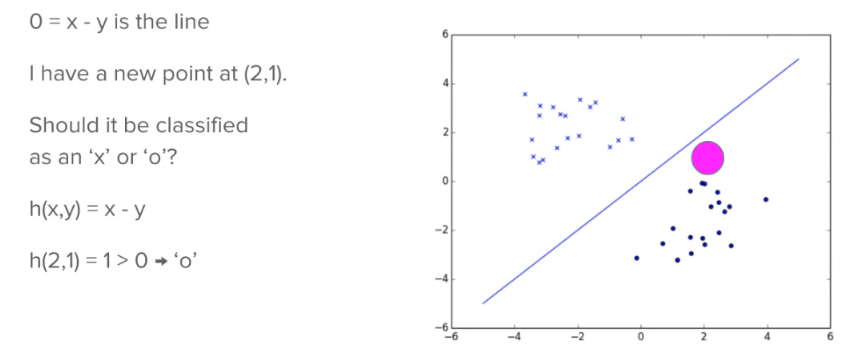
Cost function = Error function = objective function (J). Minimize cost and error with respect to model parameters.

L, log-likelihood. Maximize L = log [P(Data| Model)]. Maximizing the log likelihood is same as maximizing the cross-entropy

## Regression

### Linear Regression (2-dimensional classification)

* Simplest is y = mx +c
* Most complex is multiple variables along with a polynomial fit
* Basically separate a set of data into 2 clusters or groups



* Machine Learning : Hypothesis Test. h is the hypothesis function.
  + h(x) = w0 + w1x1 + w2x2 (or)
  + In vector form h(x) = wTx

## Logistic Regression

Logistic regression is a classification. This is the building block (neuron) of the neural networks. Example problems are:

* Predict the digit to be written by the user (MNIST database for image processing of written digits or letters)
* Image classification. Classify as storm, urban, trees, with cars,
* Logistic regression, Decision trees, k-nearest neighbours are the models. The key difference are:
  + Logistic regression assumes that the data is divided by a line
* Logistic regression is a linear model

### Project

Description: A data scientist at an e-commerse store want to predict user actions on a site. This has direct monetary impact as follow:

* Predict bounce (prompt)
* Discover which areas of the site are weak ex. Mobile, user-friendliness
* Make data driven decisions
* Use science to improve user experience

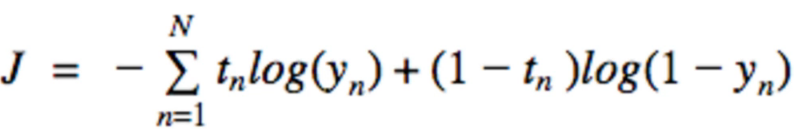
Data:

* CSV Data. First line is header
  + Ismobile (0/1)
  + N\_Products\_viewed (int >=0)
  + Visit\_duration (real >=0\_
  + Is\_returning\_user (0/1)
  + Time of day (0/1/2/3 = 24h split into 4 categories, assumptions?)
  + User\_action (bounce, add to cart, begin checkout/finish checkout
* User\_action is to be predicted
  + Logistic class : binary classification
  + Neural network class : multi-class classification

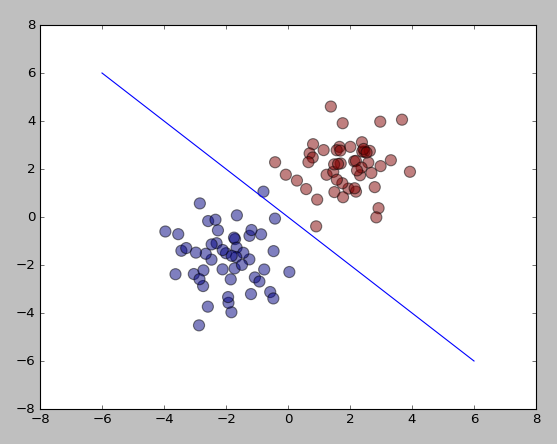
Data Pre-processing

* Logistic regression/neural netowrks work on numerical vectors, not categories.
  + One-hot encoding.
  + Time of day will become 4 columns (12am-6am, 6am-12pm, 12pm-6pm, 6pm-12am) with 1 and 0 as appropriate.
* Binary categories can be kept as is. Eg. Is\_mobile; is\_returning\_user can be handled for bias
* Numerical columns : eg: n\_products\_veiwed and visitor\_duration
  + n\_products\_veiwed. Int is category or ordinal. Can we treat it as real. Eg. 1.5, 2.5 etc.
  + Scale and distance is important. i.e. normalization (0 mean, 1 standard deviation). Z = (x- mean)/standard deviation
  + Eg. Sigmoid = exp(x)/[exp(x)+1]
    - But sigmoid(10) = sigmoid(11) = sigmoid(12) = 0.999. Not so useful. Sigmoid is useful only after data is normalized.

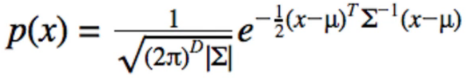
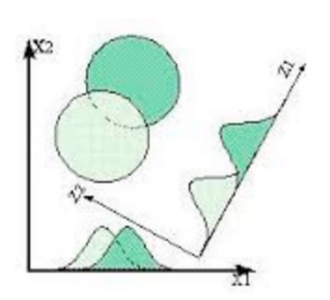
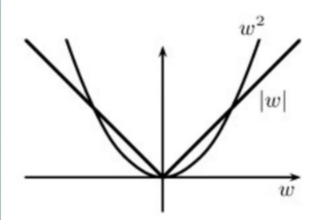
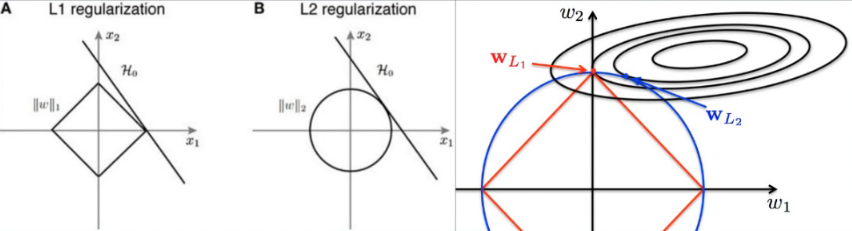
### Cross Entropy (CE) Method:

* Example Error functions
  + Linear regression – squared error, J = Ʃn(tn-yn)2. This assumes a Guassian distributed error. Log (Gaussian) = squared function.
* Logistic regression cannot be Gaussian distributed as the target is only 0/1 and the output is only a number between 0 and 1.
  + 0 if correct, >0 if not correct, more wrong implies more cost
* Cross entropy error function
  + J = -[ t log(y) + (1-t) log(1-y) ] where t is target, y is output of logistic
  + Log(y) number between 0 and inf
  + Some examples
    - t = 1; y = 1; J = 0
    - t = 0 ; y = 0; J = 0
    - t = 1; y = 0.9; J = 0.11
    - t= 1 ; y = 0.5 ; J = 0.69
    - t = 1, y = 0.1 ; J = 2.3
  + for multiple training examples. Note a bracket is missing
    - 
* Generate random data sample according to a specified mechanism
* Update the random mechanism parameters based on data produced to better sample the next iteration i.e. minimizing the cross-entropy (or) Kullback Liebler divergence.

### Weights (ML Training)

* Step 1: Using random function for finding weights, the prediction is not good ~30% on average. Reference codes are below
  + ecommerce\_process.py
  + logistic\_predict.py
  + logistic1.py
* Step 2: Using a closed form solution, w is determined and defined.
  + Logistic2.py
  + The cross-entropy error decreased significantly from ~100 to below 2 for majority of the instances using the closed form w solution.
  + Visualize the plot data points and the discriminating line
  + 
* Step 3: Use gradient decent method to determine weights:
  + See Logistic3.py
  + The output of this is different from closed form solution i.e. the solution is off by a factor (i.e. 3 times higher).
  + Further training is required:
* Step 4:
* How to find weights?
  + Logistic regression needs probability and statistics
  + Finding w is machine learning training
* Train or learn to find parameters for the model so predictions are accurate
* Interpreting the weights for logistic model:
  + wi is the amount log odds increase when xi is increases by 1.

A typical problem:

* 2 different data sets defined by 2 Gaussian clouds
* Same covariance
* Multivariate Gaussian PDF
  + 
  + 
* An example Data:
  + µ0 = (-2,-2)T
  + µ1 = (2,2)T
  + σ2 = 1
  + Solution is
    - w = (4,4)
    - b = 0 (assuming equal number of samples for each class)
  + This method is also called linear discriminant analysis (LDA)
  + If covariance is a diagonal matrix : Naive bayes
  + If covariances are different: Quadratc discriminant Analysis (QDA)
* Optimizing weights
  + If we assume data is Gaussian distributed with equal covariance, we can use bayes method.
  + For linear regression, we can take the derivative and solve for maximizing the likelihood.
  + However, for multiple weights, taking derivative is not possible. 1 equation and too many variables.
* Gradient descent
  + Take small steps in the direction on the derivative
  + Step size is the learning rate (1). There is no scientific method to know the learning rate. Have to guess/experiment/train
  + Ex:
  + w = -2
  + w = -2-1\*(-1) = -1
  + Slope is zero at the bottom, so no more changes will occur to the weight.
* Gradient Descent Derivation
  + Derivative of J = XT(Y-T)
  + <http://mccormickml.com/2014/03/04/gradient-descent-derivation/>
  + <http://eli.thegreenplace.net/2016/understanding-gradient-descent/>
  + <https://sebastianraschka.com/faq/docs/linear-gradient-derivative.html>
* Regularization.
  + There may be multiple solutions
  + Regularization is required
  + Training data should not be overfitting. It should nicely fit the existing data. If training is representative of the range of predicted data, then the data is good.
  + Model overfit will have to guess the output in a space never seen before – Not good.
  + Logistic regression will try to go to w = (0, ∞, ∞)
  + Regularization will penalize very large weights. A smoothing parameter λ ranging from [0.1 to 1].
  + Jreg = J + λ/2||w||2 = J + λ/2 wTw. Note the L2 Norm of the weights – L2 regularization (or) Ridge Regression, JRIDGE
  + Regularization cost, reg\_cost = λ/2 wTw.
  + d (reg\_cost) = λ wi. i.e. d (reg\_cost)/dw = λ w.
  + d Jreg/dw = J = XT(Y-T) + λ w.
  + Probabilistic Perspective (Bayesian)
    - Posterior likelihood x prior
    - Our prior belief about w is that it is Gaussian distributed with variance 1/λ.
* Without regularization, we maximize the likelihood
* With regularization, we maximize the posterior. This is also known as “maximum a poteriori” or MAP estimation.
* See “logistic4.py” for this method implemented in code.
* L1 regularization (or) (or) Lasso Regression, JLASSO
  + Skinny data (D<<<N) preferable.
  + Fat (D> N) not preferable.
  + Select small number of important features that predict the trend
  + Remove noise. We call this sparsity as most weights will be zero while only a few are non-zero
  + Laplace distribution on weights.
  + d Jreg/dw = J = XT(Y-T) + λ sign(w).
    - if w>0, sign(w) = 1; w<0, sign(w) = -1; w=0, sign(w) = 0
  + See “l1\_regularization.py” for this method implemented in code.
* L1 vs. L2 regularization
  + L1 objective: few w’s non-zero while many equal to 0
  + L2 objective: all w’s close to zero but NOT exactly 0
  + Both methods help prevent overfitting by not fitting to noise
    - L1 accomplishes by choosing the most important features
    - L2 accomplishes this by making the assertion that none of the weights are extremely large (influential)
  + L2 penalty is quadratic, L2 is absolute function. Note that we use gradient descent i.e. derivative.
    - JL2 = J + λ1 |w|2
  + Quadratic : as w tends to be zero, derivative tends to be 0
  + Linear,
  + 
  + 
* L1 and L2 simultaneously, ElasticNet
  + JELASTICNET = J + λ1 |w| + λ1 |w|2

### Donut Problem

## Bayesian Statistics

Naïve Bayesian Classifiers

Conditional probability

Design of experiments and ‘influence of prior beliefs’

https://en.wikipedia.org/wiki/Bayesian\_statistics

k-Nearest Neighbours

## Fuzzy Logic

Fuzzy Logic: Degrees of truth rather than true or false. Similar to how our brain works.

- Aggregate data and form number of partial truths by forming rules

- Aggregate rules further into higher truths

- When thresholds exceed, further results as motor reaction (Action).

- Believed to be important to develop human-like capabilities for AI

Fuzzy Matrix: Simple Classification to

Fuzzy Congitive Mappings: Graphical representation of the system. (D3.JS)

Implemented 4 rules using Scikit-Learn Fuzzy

Latitude/Longitude;

Operation & depth;

Well name and ID;

## Clustering

### K-means Clustering

Clustering based on number of groups.

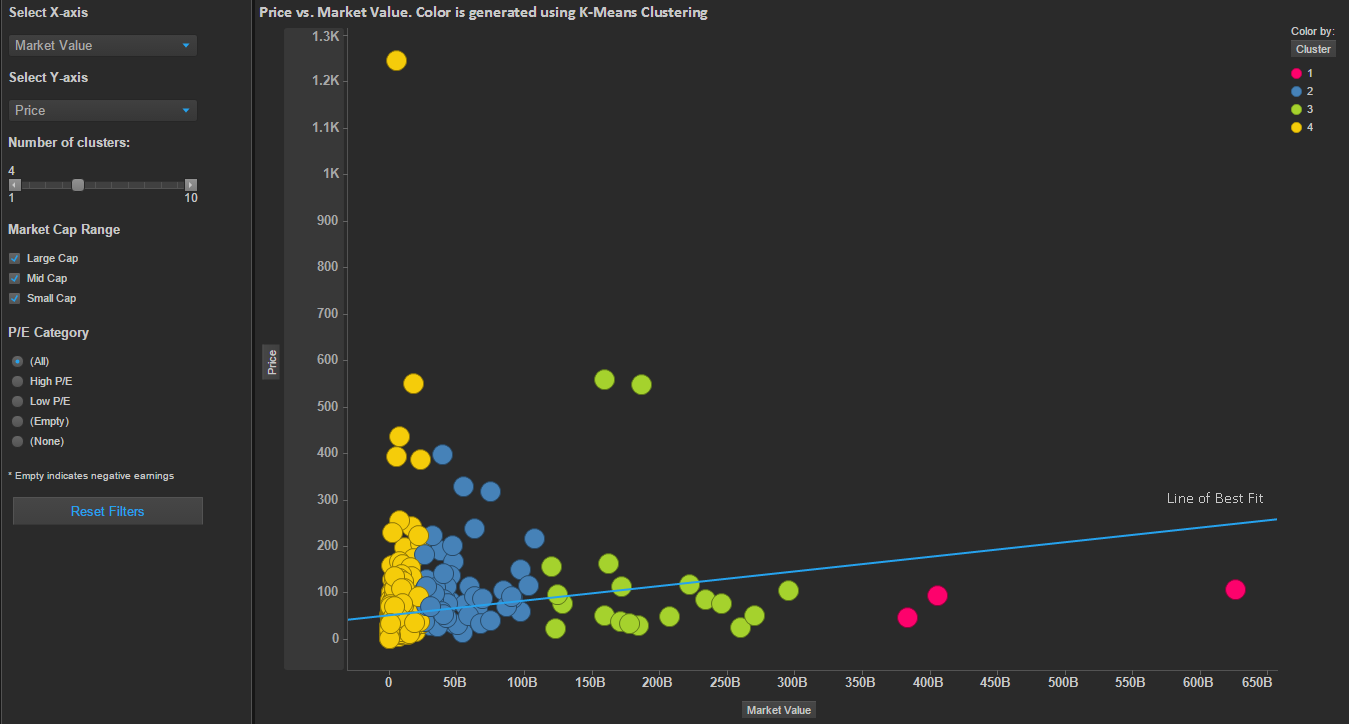


Figure 7.1 – K-Means Cluster ( 4 Clusters)

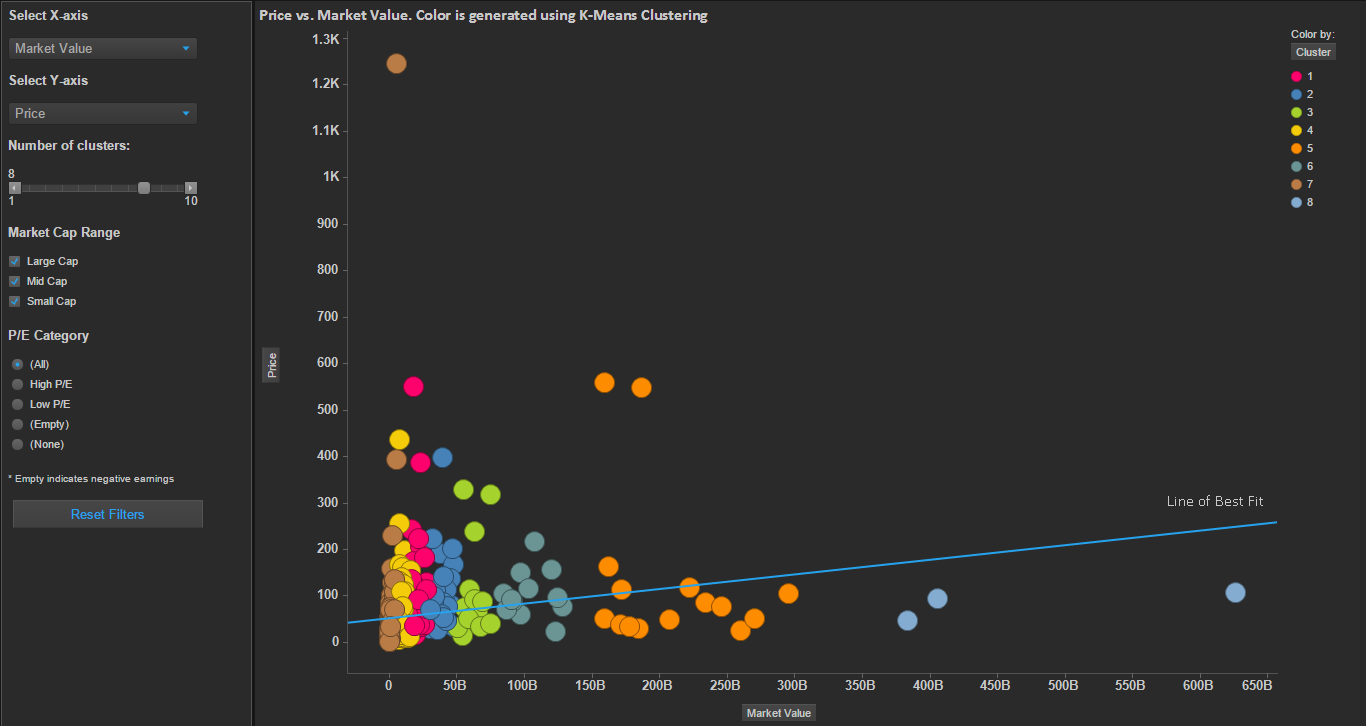


Figure 7.2 – K-Means Cluster ( 8 Clusters)

C-means clustering: Will also weight for the proximity of the data point into consideration. Computationally more intensive.

https://vpetro.io/fuzzylogic/fuzzy\_mehaan\_joy.html

https://stackoverflow.com/questions/2345903/whats-is-the-difference-between-k-means-and-fuzzy-c-means-objective-function

## Random Forest

### Algorithms

Decision tree learning

Tree bagging

Bagging to Random forests

Extra trees

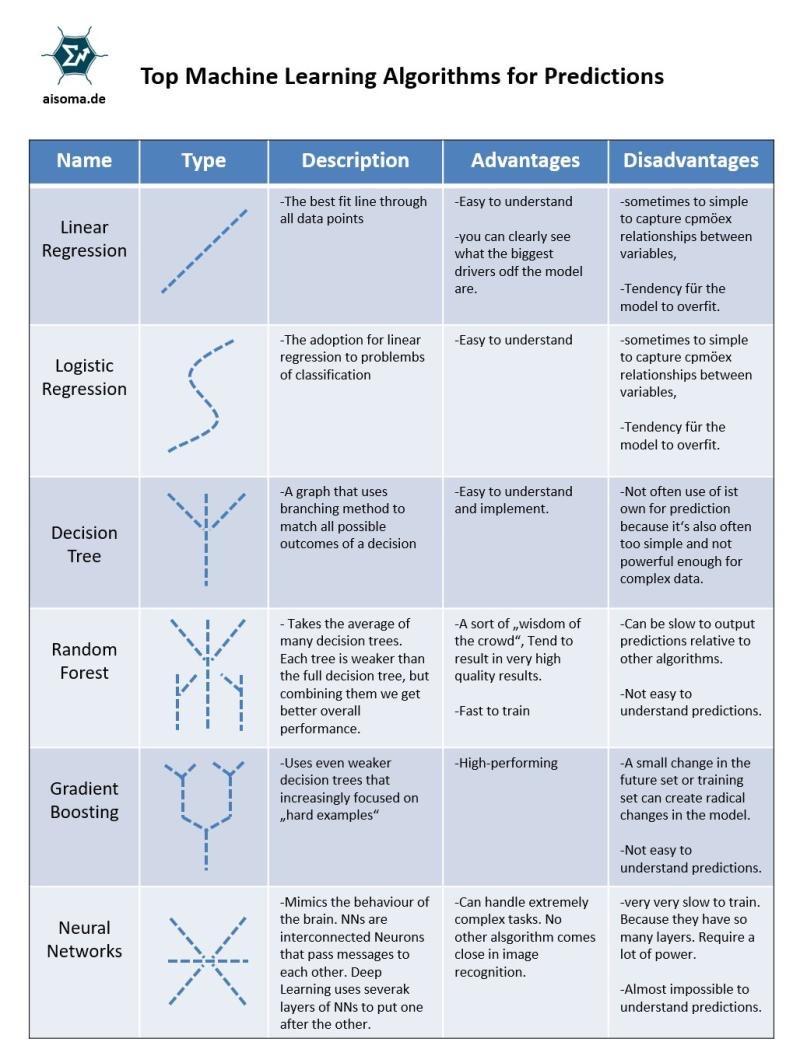
### Properties

Variable importance

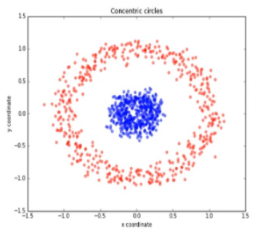
Relationship to nearest neighbours

Unsupervised learning with random forests

## Predictions



## Machine Learning – Practical Issues

* Regularization and overfitting:
  + Model performs too well on training data but not well on test or future data
  + Can happen because of irrelevant inputs
* Cost function
  + Infinite weights give an ideal solution
* Some data sets cannot be solved using a straight line (limited). See example below
  + Human pattern can recognize but not a logistic regression model as a line can not separate 2 classes
  + 

## Big Data Statistical Methods

\*\*\* Statistically, Overfitting: Old Problem, New Solution \*\*\* \*\*\*

~ Overfitting, a problem akin to model inaccuracy, is as old as model building itself, as it is part of the modeling process. The effect of overfitting is an inaccurate model.

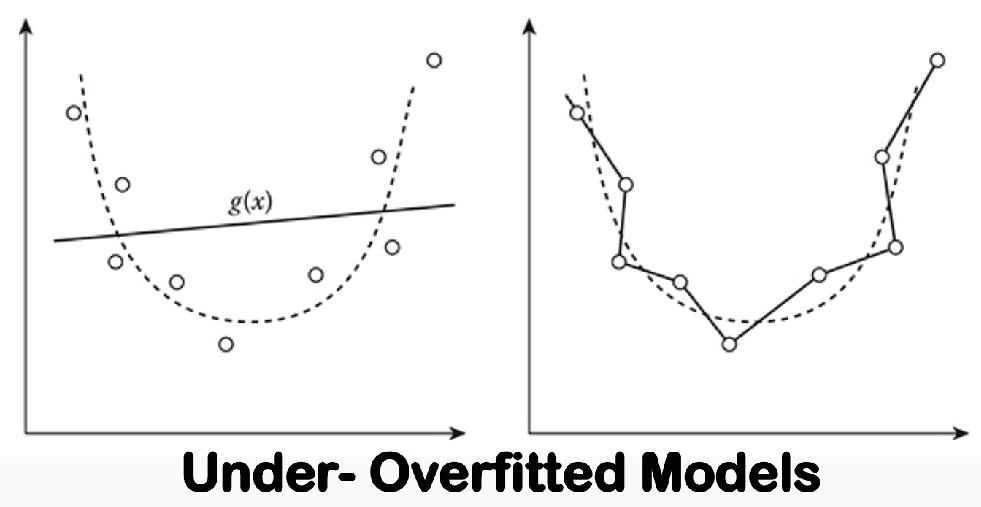
~ In Chapter 38 in my newly released book (See Comment A, below), I introduce a new solution, based on the data-mining paradigm of any machine-learning technique, to the old problem of overfitting.

~ I illustrate how a given machine-learning technique can identify the complexity of the idiosyncrasies and

subsequently instruct for deletion of the individuals that contribute to the complexity of the data under consideration.

~ I deliver a simple solution to the old and always lurking problem of overfiitting.

--- B. Noted

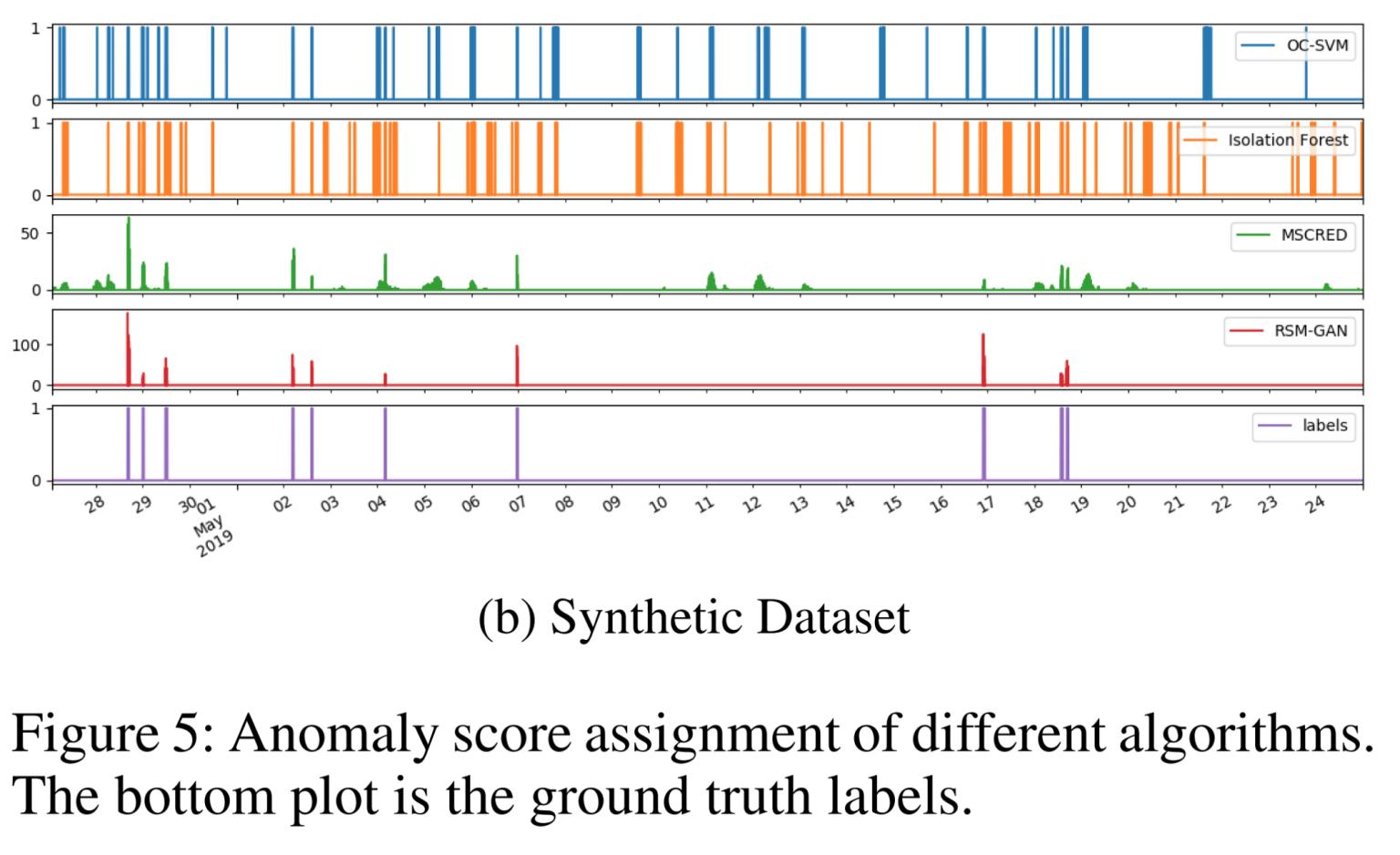


## Anomaly Detection

Robust anomaly detection is a requirement for monitoring complex modern systems with applications such as cyber-security, fraud prevention, and maintenance.

There are 3 main types of detection methods:

* classical time series analysis (TSA) based methods;
* classical machine learning based methods;
* deep learning based methods



## References

https://www.linkedin.com/pulse/5-big-data-statistical-analysis-methods-naveen-joshi Big Data Statistical Methods

https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/

<http://scikit-learn.org/stable/supervised_learning.html#supervised-learning>

https://ai.google/education#?modal\_active=none Machine Learning with Google

Predictions



# Scikit-learn ML Examples

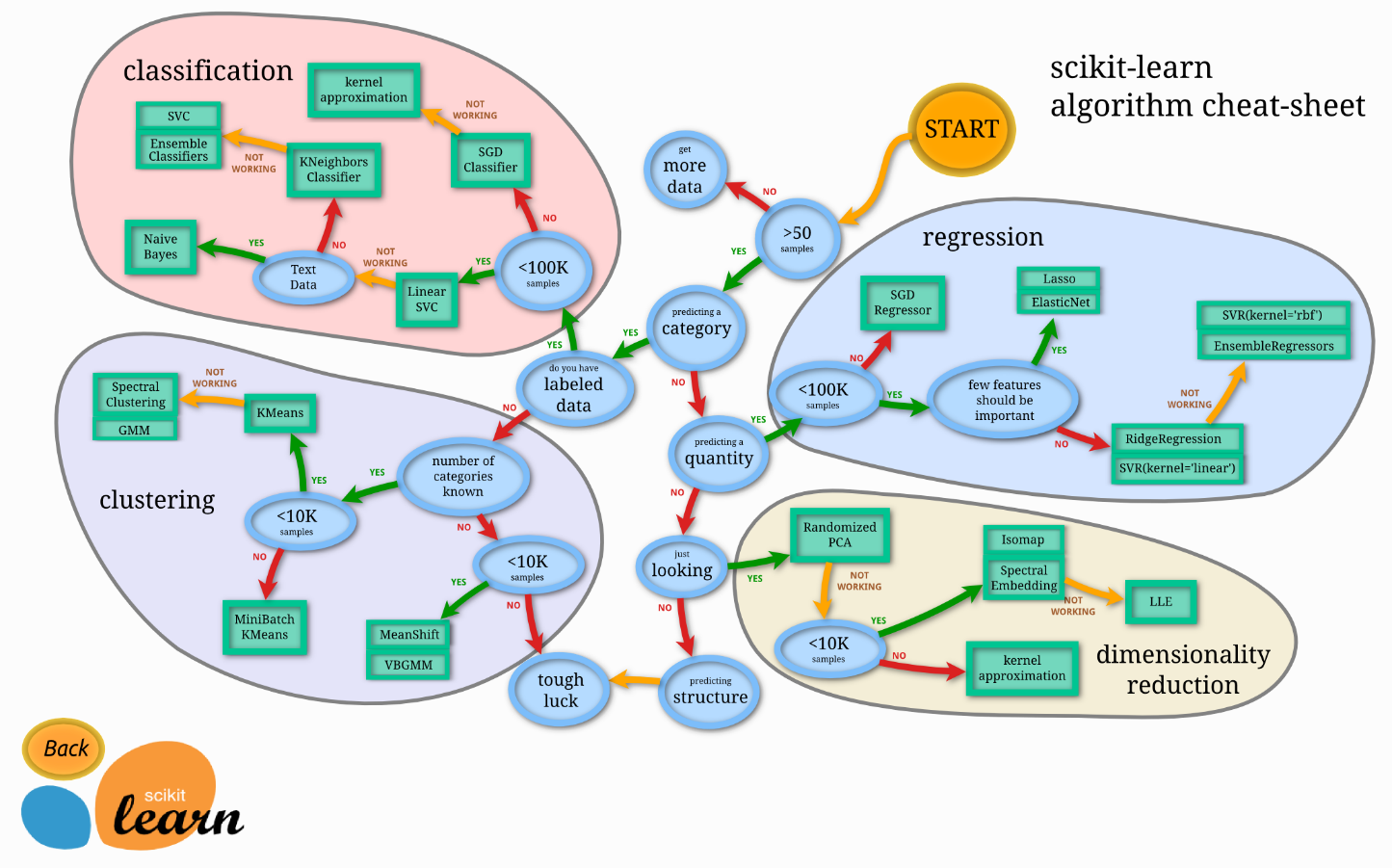
The examples provided as explanation in this section are based on below:

http://scikit-learn.org/stable/auto\_examples/index.html

## Introduction

* Python has rich and healthy ecosystem for data analysis. Scikit-Learn is the best and most effective library
* Scikit-Learn provides supervised and unsupervised algorithms and is built over SciPy

<https://www.analyticsvidhya.com/blog/2016/12/cheatsheet-scikit-learn-caret-package-for-python-r-respectively/>



## Scikit-Learn ML Summary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **Model** | **Supervised/ Unsupervised** | **Create Model/ Estimators** | **Model Fitting** | **Prediction** |
| 1 | Linear Regression | Supervised | LinearRegression( normalize = true) | lr.fit(X,Y) | Lr.predict(X\_test) |
| 2 | Support Vector Machines (SVM) | Supervised | SVC( kernel=’linear’) | svc.fit(X\_train, y\_train) | svc.predict |
|  | Decision Tree - Regression | Supervised | DecisionTreeRegressor( max\_depth=2) | regr\_1.fit(X, y) | regr\_1.predict( X\_test) |
|  | Decision Tree - Classification |  |  | DecisionTreeClassifier( ).fit(X, y) | clf.predict(np.c\_[ xx.ravel(), yy.ravel()]) |
| 3 | Naïve Bayes | Supervised | GaussianNB() |  |  |
| 4 | k-NN Classification | Supervised | KneighborsClassifier( n\_neighbors = 5) | knn.fit(X\_train, y\_train) | knn.predict\_proba( X\_test) |
|  | k-NN Regression | Supervised | neighbors. KNeighborsRegressor( n\_neighbors=15) | knn.fit(X, y) | knn.predict(T) |
| 5 | PCA | Unsupervised | PCA( n\_component=0.95) | pca.fit\_transform( X\_train) |  |
| 6 | K Means | Unsupervised | KMeans(n\_clusters=3, random\_state=0) | k\_means.fit(X\_train) | k\_means.predict( X\_test) |
|  |  |  |  |  |  |

Table 8.1 – ML Model Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Metric Class** |  |  |
| 1 | Classification Metrics | Accuracy Score  Classification Report  Confusion Matrix |  |
| 2 | Regression Metrics | Mean Absolute Error  Mean Squared Error  R2 Error |  |
| 3 | Clustering Metrics | Adjusted Rand Index  Homogeneity  V-measure |  |
| 4 | Cross-Validation | cross\_val\_score |  |

Table 8.2 – ML Model Performance Metrics

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Method** |  |  |
| 1 | Grid Search | GridSearchCV(estimator=knn, param\_grid=params)  Best\_score\_  Best\_estimator\_n\_neighbors |  |
| 2 | Randomized Parameter Optimization | RandomizedSearchCV(estimator=knn, param\_distributions=params, cv=4, n\_iter=8, random\_state=5)  Best\_score\_ |  |

Table 8.3 – ML Model Tuning

http://scikit-learn.org/stable/tutorial/index.html

## Regression

### Linear Regression

A simple process for linear regression is given below:

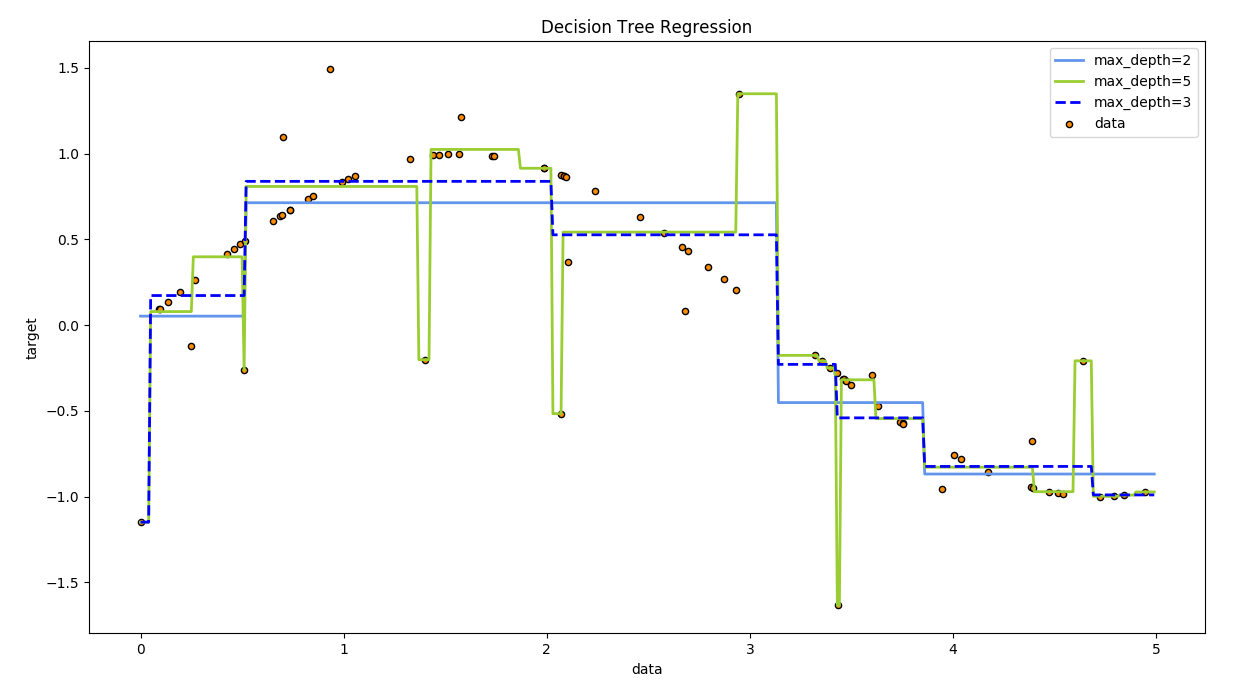
### Logistic Regression

## Decision Trees

Consists of :

* Decision tree regression
* Decision tree classification

### Ex: Decision Tree Regression

* Filename: tree/plot\_tree\_regression.py
* A 1D regression with decision tree
* Simple 1 dimensional data considered for analysis
* Lessons:
  + If maximum depth of tree (‘max\_depth’) is set too high, the decision trees fit the noise (overfit)
  + Simple to understand and interpret. Trees can be visualized
  + Cost of using tree is logarithmic in number of data points used to train the tree
  + Ability to handle multi-output problems
  + More useful in white box model. Difficult to interpret in black box model (eg. Artificial neural network)
* Example output 1-D regression output is shown below:
* 

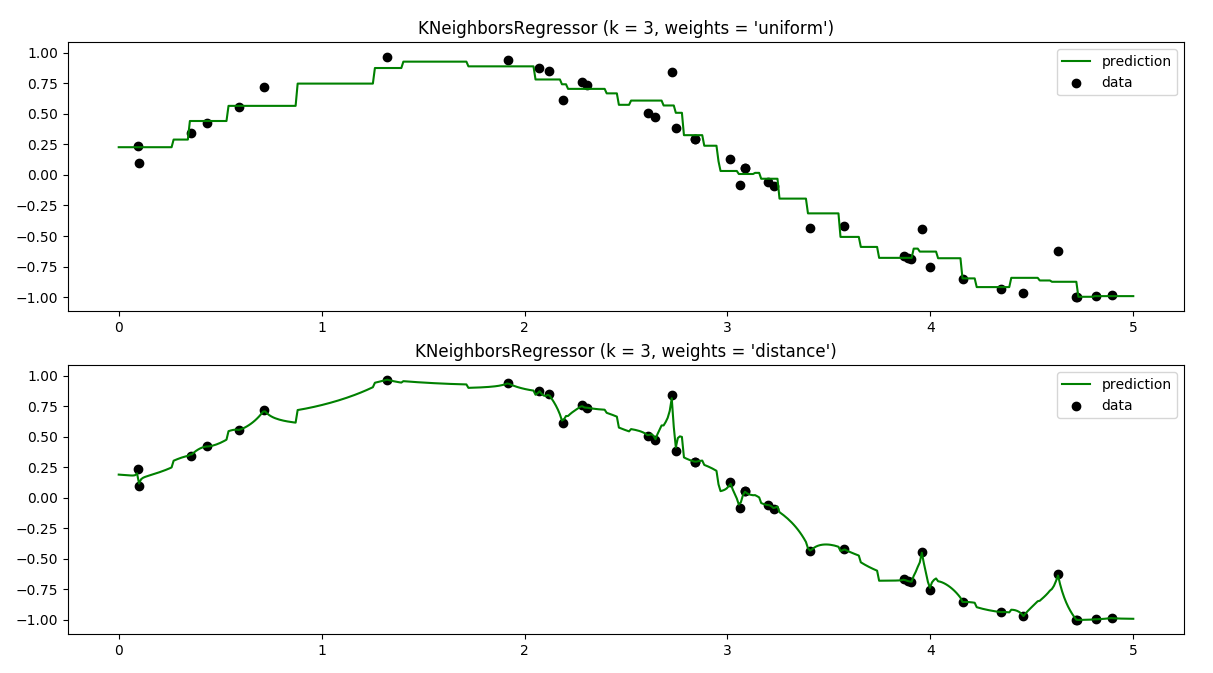
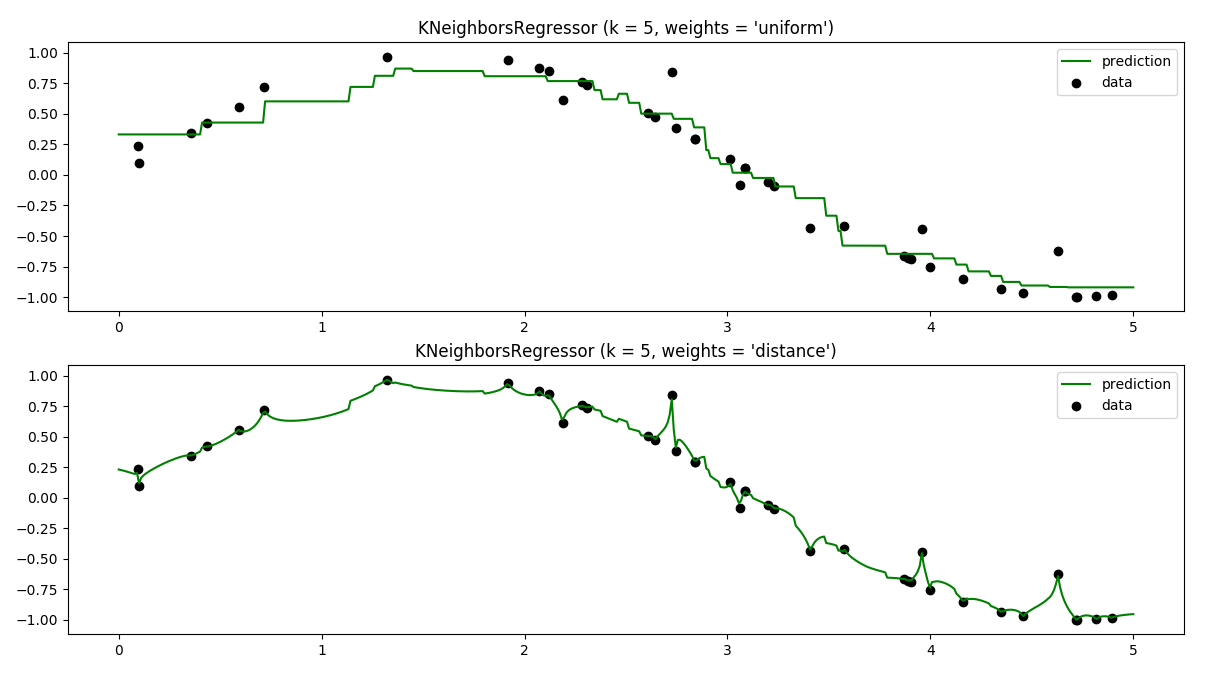
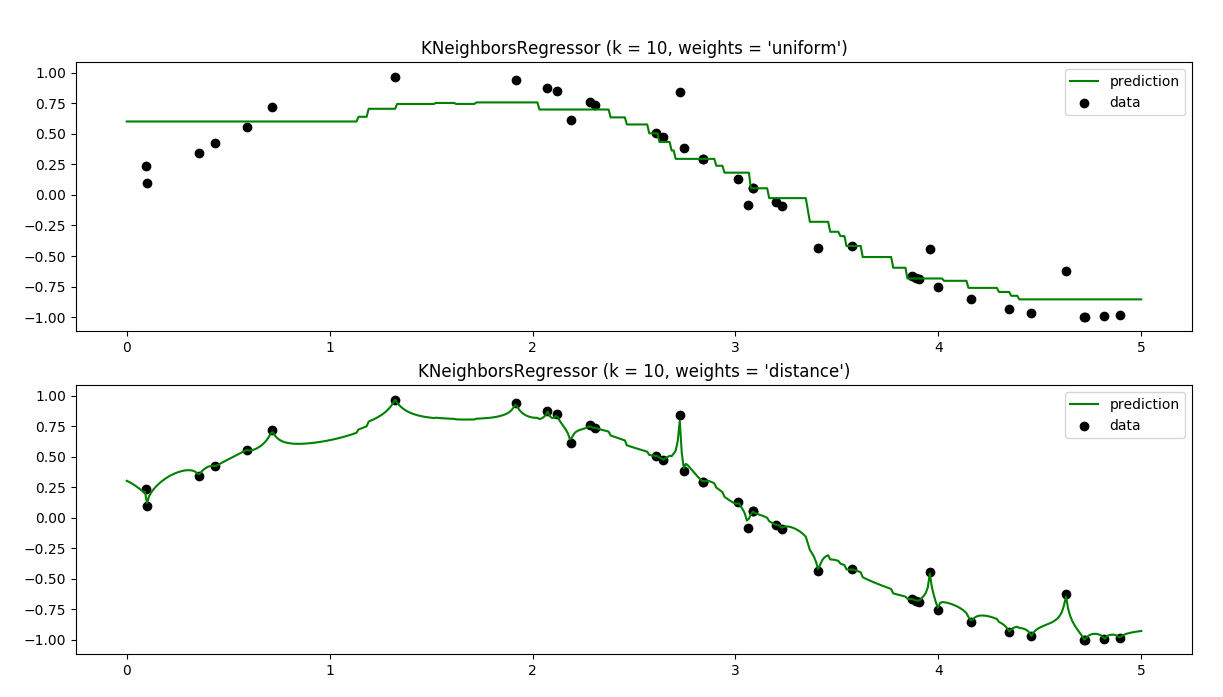
### Ex: Decision Tree Classifier

### Ex: Decision Tree Multi-Output

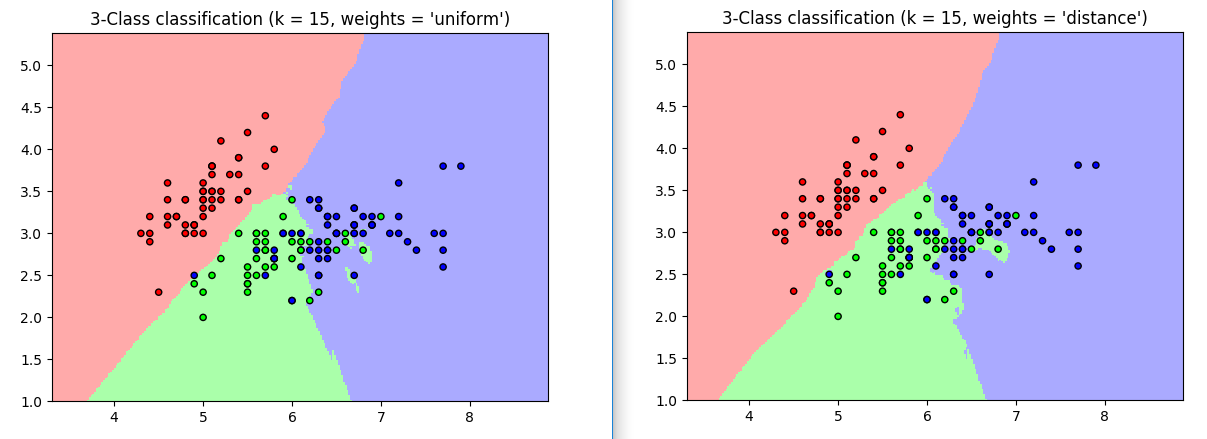
## k-NN

* A simplest machine learning algorithm
* A non-parametric method for classification and regression
* The output can be classification or regression
* Classification
  + Classification output is a class membership
  + A object being assigned to the class most common among its k nearest neighbours
  + Example, k =1 is assigned to the class of the single nearest neighbour.
* Regression
  + Ouptut is a property value for the object
  + This value is the average of the values of its k nearest neighbors
* Methods used for k-NN
  + ‘uniform’ : uniform weights. All points in each neighborhood are weighted equally.
  + ‘distance’ : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
  + [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

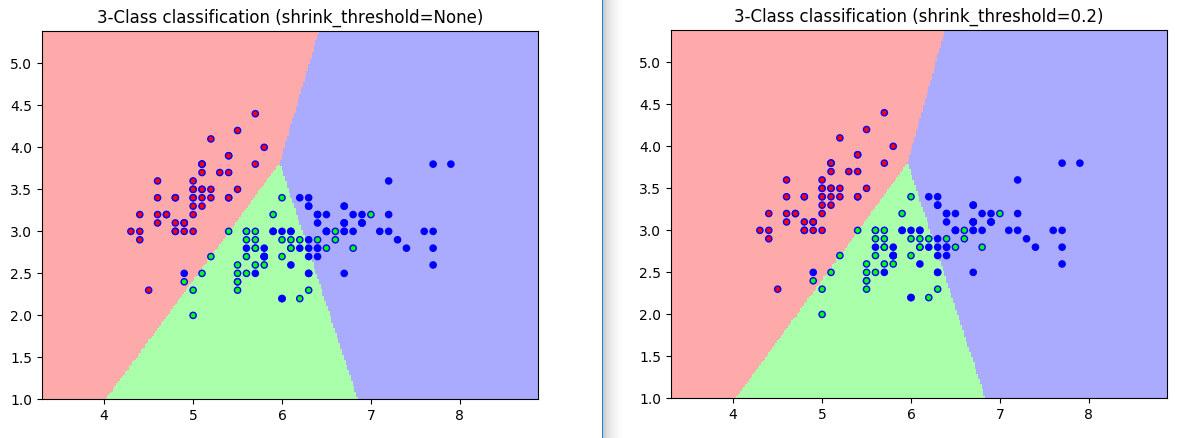
### Ex: k-NN Regression

* Filename : plot\_regression.py
* For number of neighbours = 3
* 
* For number of neighbours = 5
* 
* For number of neighbours = 10
* 

### Ex: k-NN Classification

* Filename : plot\_regression.py
* For number of classes= 3. The listed colormap seems to be defining tis.
* Number of neighbours = 15
* Nearest neighbours method:
* 

### Ex: K-NN Classification Nearest Centroid classification

* Filename : plot\_nearest\_centroid.py
* 

### Ex: K-NN Classification Nearest Centroid classification

X = [[0, 4], [1,-3], [2,6], [3,2]]

y = [0, 0, 1, 1]

from sklearn.neighbors import KNeighborsRegressor

neigh = KNeighborsRegressor(n\_neighbors=2)

neigh.fit(X, y)

print(neigh.predict([[1.5, -9]]))

## Naïve Bayes

Study Bayesian gausian mixture models

## Neural Networks

## Model Selection & Performance Metrics

Selecting the model is an important aspect of machine learning.

* ADFA?
* AS/

## References

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Link** | **Description** |
|  | http://scikit-learn.org/stable/index.html# |  |
|  | https://www.datacamp.com/community/blog/scikit-learn-cheat-sheet |  |
|  | http://scikit-learn.org/stable/modules/tree.html#tree | Decision trees |
|  |  |  |

# Deep Learning

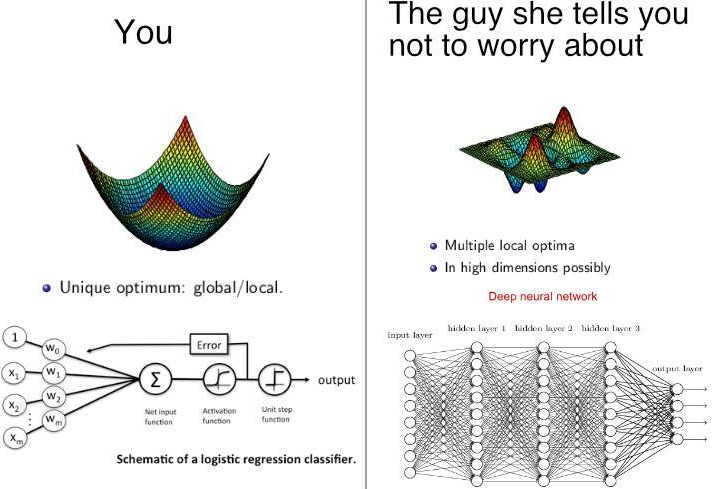


Figure 9.1 - Deep Learning vs. Machine Learning

## Machine Set-up

A linux machine is the best way to start the work as most ML programs are supported in this environment.

* Create a Lubuntu machine in Windows to learn ML
  + Username: ml
  + Password: mlpassword
  + sudo apt-get update (To update firmware)
  + sudo apt-get install python-numpy python-scipy python-matplotlib ipython python-pip python-dev python-setuptools
* Installing Tensorflow:
  + sudo apt-get install python3-pip python3-dev python-virtualenv
  + pip3 install tensorflow-gpu

Courses

Data Learning Prerequisites: Logistic Regression in Python

Deep Science: Deep Learning in Python

Deep Science : Practical Deep Learning in Theano + Tensorflow

## Stan

Stan (Sampling Through Adaptive Neighborhoods) is a package for Bayesian inference using the No-U-Turn sampler, a variant of Hamiltonian Monte Carlo.

Stan is a program for general Bayesian analysis, developed recently at Columbia University. It uses a modelling language inspired by BUGS and superficially similar, but it is conceptually different in many ways.

Stan is expected to fit the following models:

* Varying intercept
* Varying slope
* Multilevel models with multiple coefficients per group

Python Stan development:

<https://statmodeling.stat.columbia.edu/2020/08/09/regression-and-other-stories-translated-into-python/>

<https://github.com/bambinos/Bambi_resources>

<https://github.com/bambinos/Bambi_resources/tree/master/Rethinking>

<https://www.amazon.ca/Regression-Other-Stories-Andrew-Gelman/dp/1107676517/ref=sr_1_1?dchild=1&keywords=9781107676510&qid=1597108159&sr=8-1>

### References

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Comment** |
|  | http://mc-stan.org/users/documentation/index.html | Stan documentation |
|  | http://andrewgelman.com/2011/11/30/stan-uses-nuts/ |  |
|  | https://github.com/stan-dev/stan/wiki/Stan-Best-Practices | Stan best practices |
|  | https://pystan.readthedocs.io/en/latest/getting\_started.html |  |
|  | <https://pystan.readthedocs.io/en/latest/windows.html> | Installation instructions for compilers if varsalls.bat error persists |
|  | http://scikit-learn.org/stable/auto\_examples/index.html | Scikit learn examples |
|  |  |  |
|  |  |  |

## BUGS

Bayesian Inference Using Gibbs Sampling

NIMBLE can also be used to fit general models written in the BUGS language, and includes the ability to write novel sampling algorithms.

<http://www.stat.ufl.edu/system/man/BUGS/manual05/manual05.sec-1.html>

## Jags

JAGS (Just Another Gibbs Sampler) by Martyn Plummer is an open source program which was developed independently of the BUGS project. JAGS uses essentially the same model description language, but it has been completely re-written. This runs natively on Windows, Mac, Linux and several other varieties of Unix.

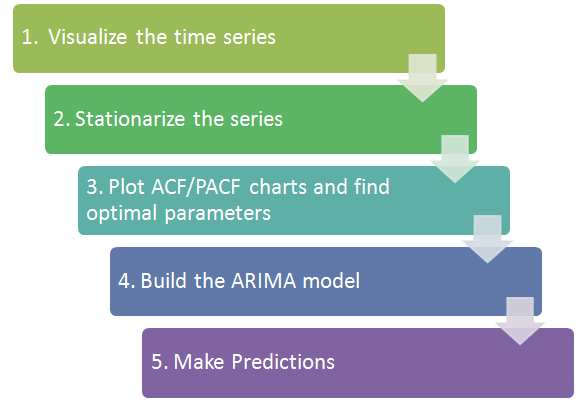
http://mcmc-jags.sourceforge.net/

## Nimble

https://r-nimble.org/

# Time Series Analysis

The time series analysis methods are given in this section. The objective of these methods is



## Introduction

Very high level time series analysis are:

* Trends
* Seasonality
* Irregularity
* Cyclic

https://towardsdatascience.com/analyzing-time-series-data-in-pandas-be3887fdd621

https://www.machinelearningplus.com/time-series/time-series-analysis-python/

## Time Series Modelling

This involves building a time series model. This is a complicated task

## Time Series Exploration

## Time Series Models



## Arima vs. FB Prophet

https://towardsdatascience.com/time-series-prediction-using-prophet-in-python-35d65f626236

https://neuralnetwork.guru/time-series-in-no-time/

https://pythondata.com/forecasting-time-series-data-with-prophet-part-1/

https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/

### References

<https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/>

REF-DS-Time\_Series\_Forecasting.pdf

REF-DS-Time\_Series\_Analysis\_For\_Beginners.pdf

REF-DS-Lasso and Ridge\_Regularize ML Models to prevent overfitting.pdf

High Frequency Time series analysis

https://github.com/RamiKrispin/USgrid

# Data Ingestion

Structured/Semi-structured/unstructured data

## Data Cleansing

How to clean the CSV files. Any techniques used?

Missing data, Errors. Remove unrealistic data points.

# Data Processing

Principle Component Analysis

## Time series analysis

https://www.xenonstack.com/blog/anomaly-detection-of-time-series-data-using-machine-learning-deep-learning Anomalies in time series data using machine learning

## Prediction tools

http://www.presenso.com/single-post/2017/05/24/The-Economics-of-the-Smart-Factory-How-does-Machine-Learning-Lower-the-Cost-of-Asset-Maintenance-Part-1 Why ML predictive Maintenance?

## References

https://onlinecourses.science.psu.edu/stat505/node/49 Principle Component Analysis

# Visualization

## How and What to Show

### Simple Visualizations

A guide for simple visualization is shown in Figure 13.1.

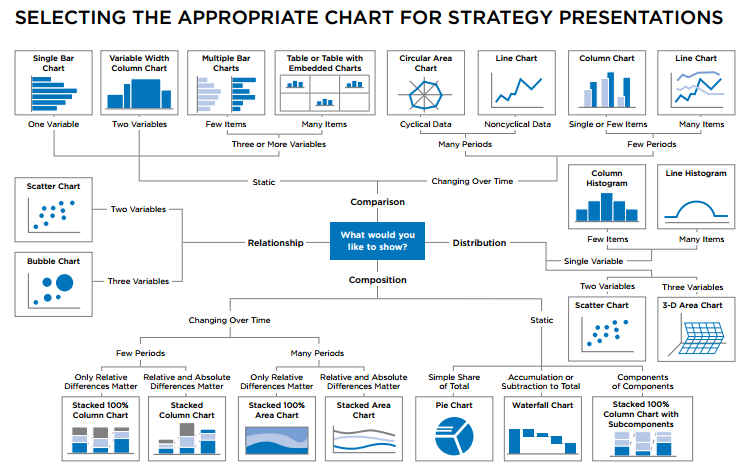


Figure 13.1 – Simple Data Visualizations

## Visualization Programs

The following visualization programs can be used for analysis:

* D3JS (Data driven documents using Javascript)
* Power BI
* TIBCO Spotfire
* Tableau
* Python Matplotlib
* Python Bokeh

Tools are used to help unravel data insights.

## Spotfire

TIBCO Spotfire is built from ground up to be the fastest data discovery tool. The key features of Spotfire are:

* An example of a sophisticated dashboard using world bank data is shown in Figure 13.2.
* This chart also contains the “Marking” feature to study any data in greater detail.
* Live filtering is also a good feature
* Creating visualizations:
  + Can be based on spotfire recommendations (or)
  + Start from scratch

### Example

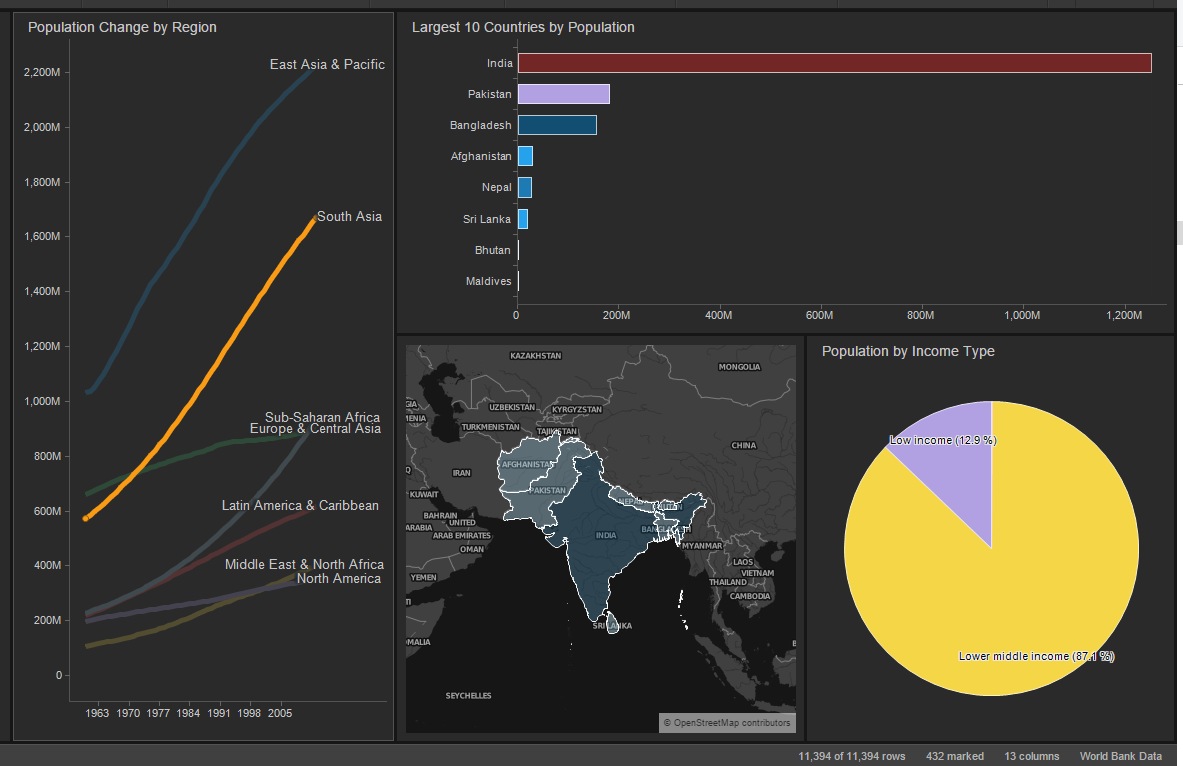


Figure 13.2 – Spotfire Dashboard with Marking Feature

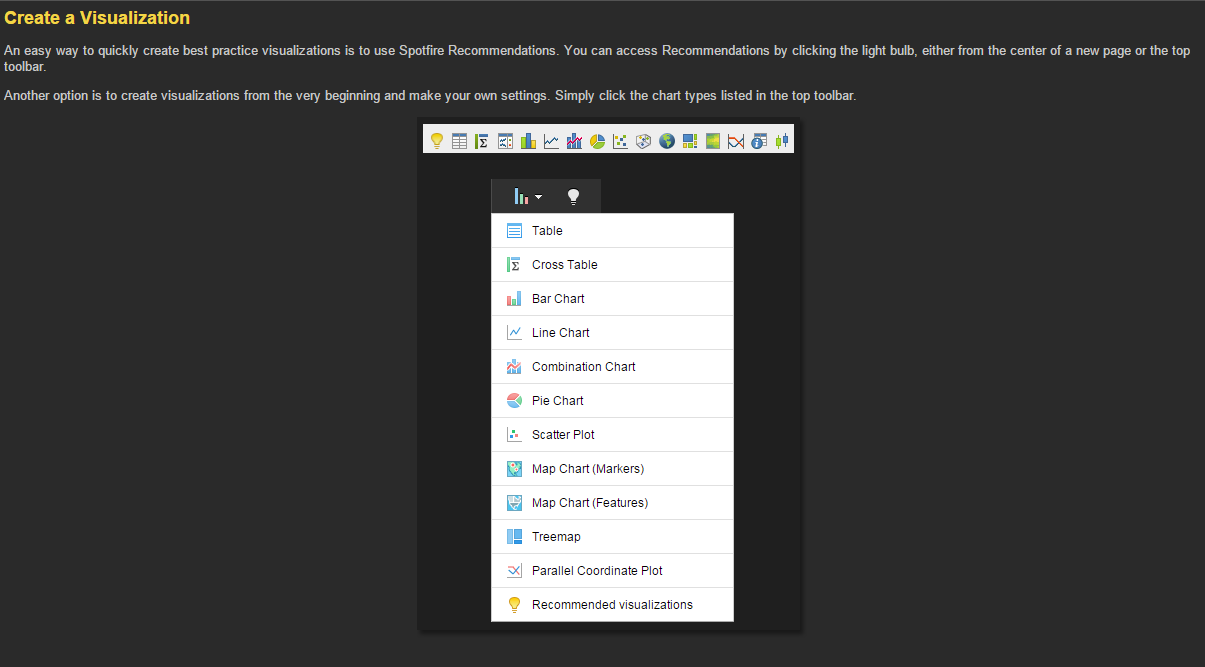
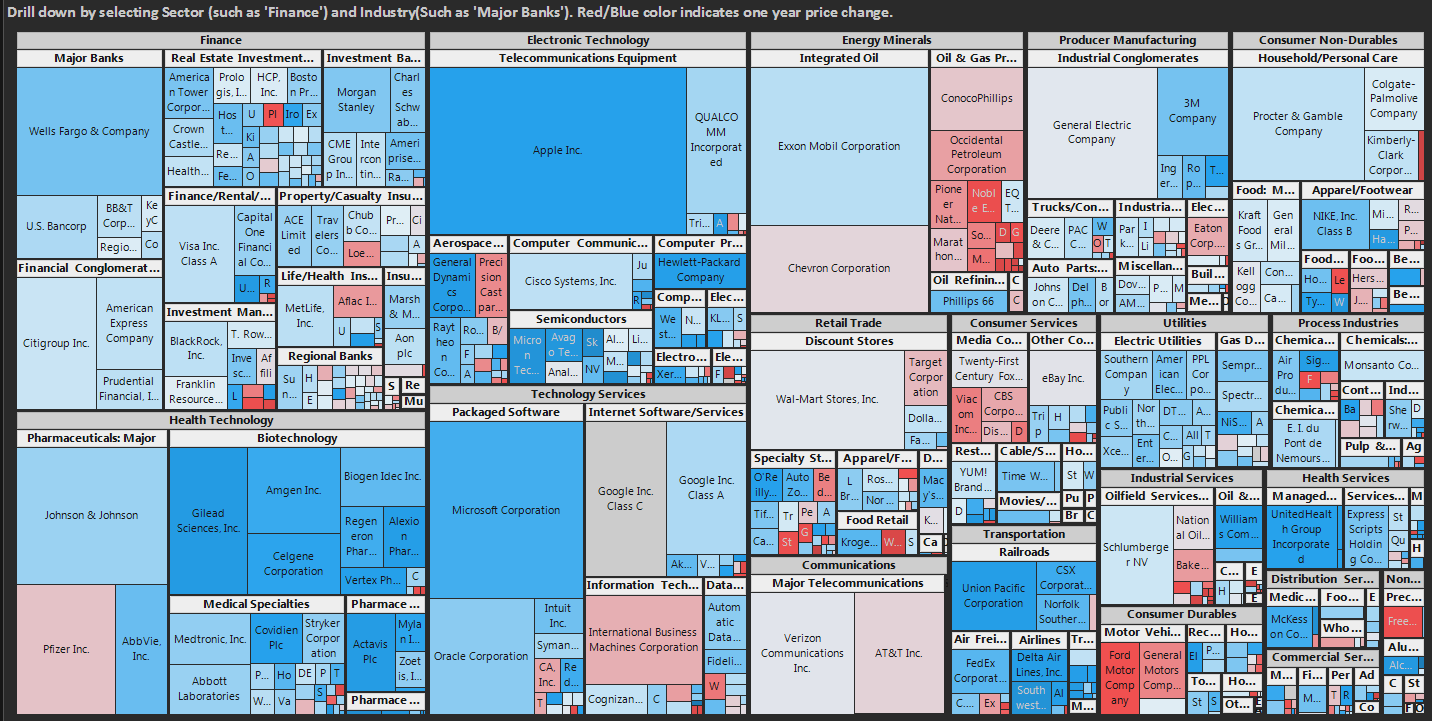
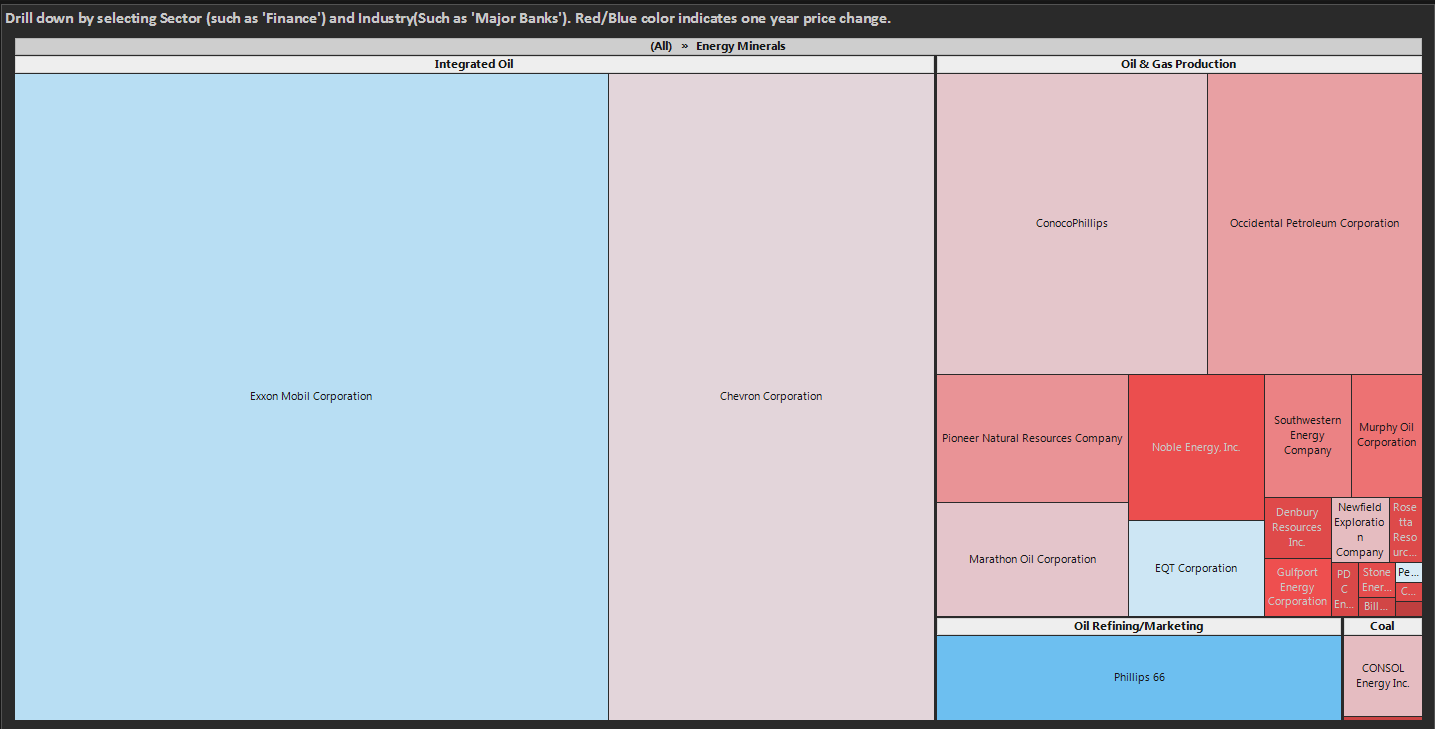


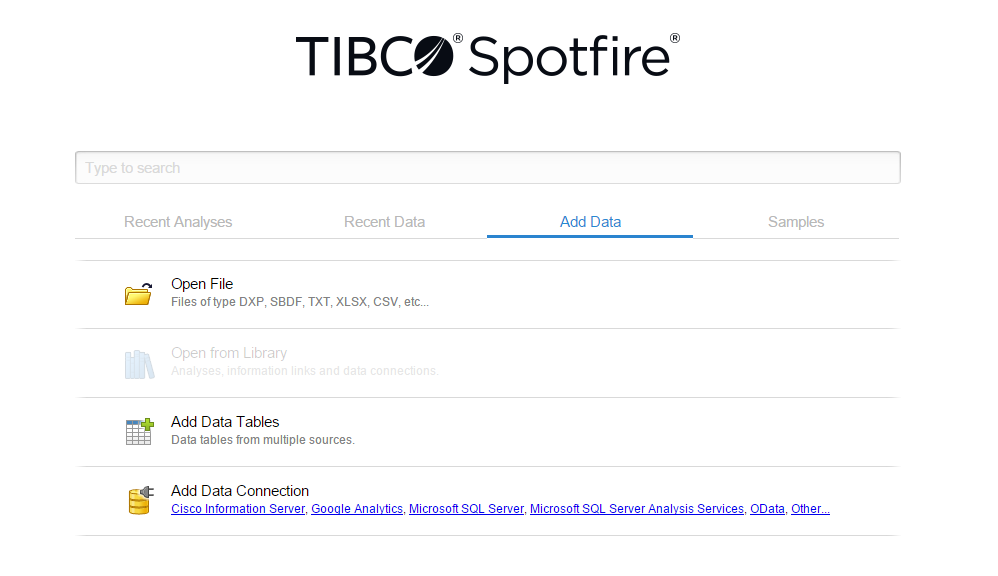
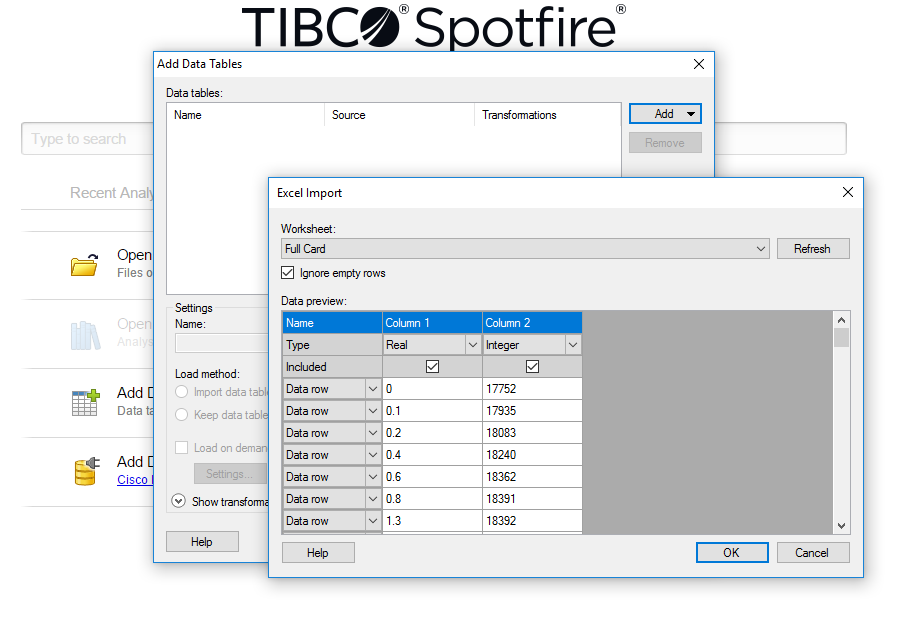
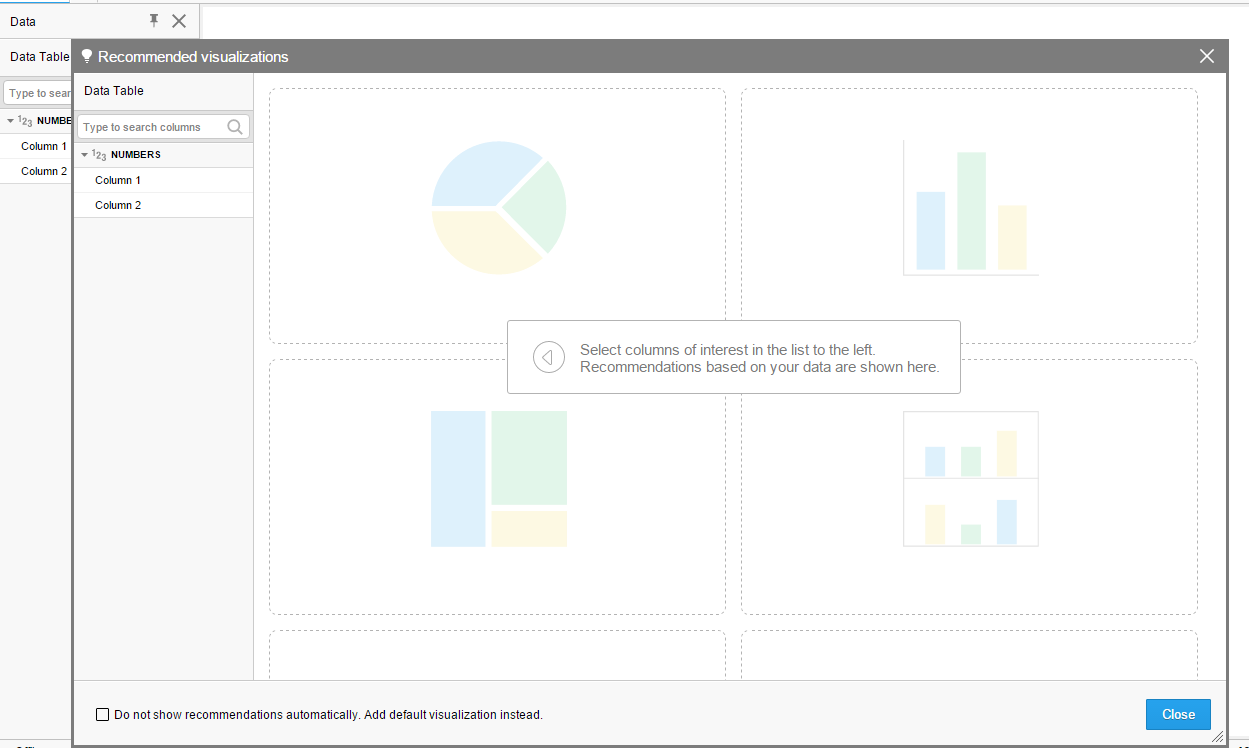
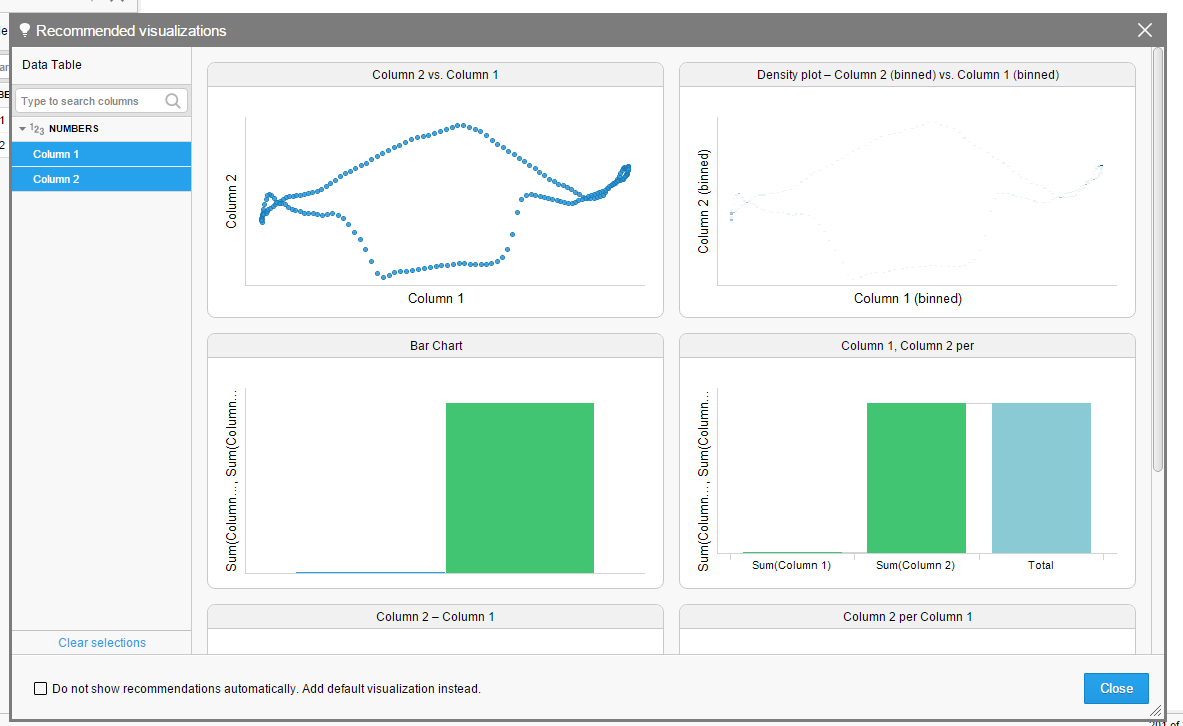
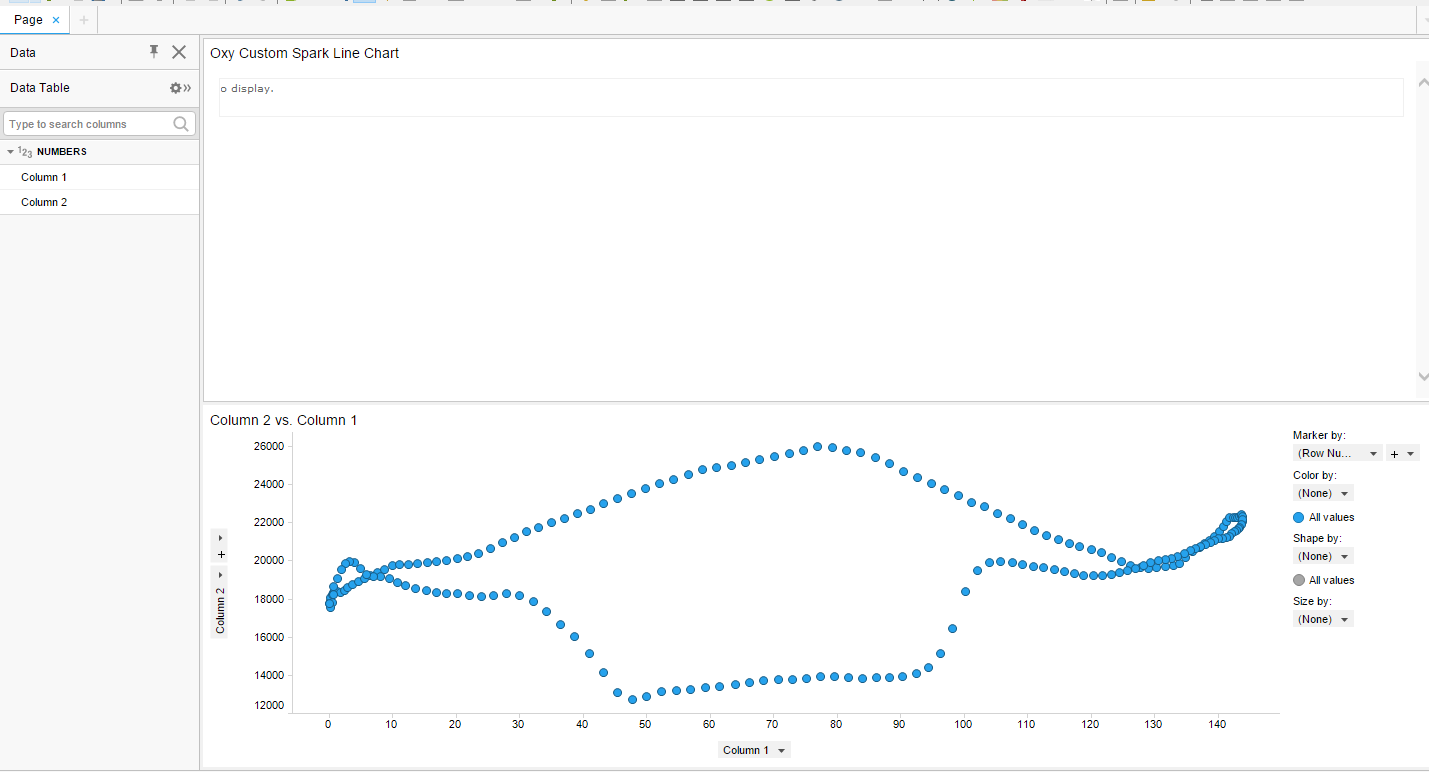
Figure 13.3 – Create Visualizations

Display by Sector and Drill down in each Sector.





### General Working

* A screenshot of opening screen
* 
* The options to add data are:
  + Open a file
  + Add data table from clipboard
  + Existing databases
  + Etc.
* Add data
* 
* Select data to be visualized
  + 
* Select view from recommended visualizations
* 
* Add recommended visualization and add further requirements
* 

## Python Bokeh

Can python bokeh charts be embedded in word documents?

### Deliver Reports

Best way to release documents to client as of now is to prepare a Jupyter Notebook and issue to client.

https://bokeh.pydata.org/en/latest/

# GIS

https://diego.codes/post/som-tsp/ Self Organizing Map

# Natural Language Processing (NLP)

## Packages

### eNLP by Equinor

<https://github.com/equinor/eNLP/>

# References

http://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/advanced-analytics-nine-insights-from-the-c-suite Mckinsey Data Analytics Article

http://www.cs.cmu.edu/~aarti/Class/10701/slides/ Machine Learning Course

http://www.datasciencecentral.com/profiles/blogs/the-mathematics-of-machine-learning Basics of Machine Learnings.

https://www.linkedin.com/pulse/statistics-versus-machine-learning-angshuman-guha Statistics vs. Machine Learning

- Courses

Courses Finished

Big Data Introduction (2017-03-04)

Business Analytics with Big Data : A Complete Guide

HBase (2017-03-11)

Building Apache HBase Applications

Spark + Python (2017-03-18)

Taming Big Data with Apache Spark and Python - Hands On!

Udemy Course List

Hadoop Administration - Hands on

Hadoop Cluster Administration Course: Guide for Hadoop Admin

Hadoop Administration: Online Hadoop Admin Training

Master Hadoop Cluster Administration

Learning Apache NiFi

The Complete Apache Kafka course for beginners

The Complete Apache Storm Tutorial for Beginners

Building Apache HBase Applications

**Deep Learning: Recurrent Neural Networks in Python**

Deep Learning with TensorFlow

Learn Big Data: The Hadoop Ecosystem Masterclass

**Business Analytics with Big Data : A Complete Guide**

*Worth it for very newbie. A 6 month old big data industry person does not need this*

**Taming Big Data with Apache Spark and Python - Hands On!**

23rd March 2017: New Course List in order of priority

* Apache Spark (1 week)
* Kafka (2 days)
* Java Maven (1 week)
* Springs (1 Week)
* Hive (?)

O&G

* Legacy Data. Industry bringing this cold dead data to life using Analytics.
* Industry currently using newer technologies and predictive analytics
  + Faced with aging assets
  + Retiring workforce
  + Decreasing margins
* Time series data for this industry is the heart beat.
  + Time lost is opportunity lost.
  + Deepscale creates opportunity
  + Frequency mining

Spark

Spark Programming Guide

<http://spark.apache.org/docs/latest/programming-guide.html>

Anadarko Analytics (Sanjay Paranji)

<https://youtu.be/1GKhITOoY-A>

https://www.linkedin.com/pulse/infographic-data-engineering-science-michiel-victor?trk=hp-feed-article-title-like Data Engineer vs. Data Scientist



Production Data:

Geology

Completions

Production Variables (Choke Settings, etc.)

- Spark

Master and Slaves

* Manager is required.
* The manager used depends on the system requirements, API compatibility etc.

Typical Data Process

Data Ingestion:

Structured/Semi-structred/unstructured data to

Hadoop

Maven Builds are performed in Eclipse

Spring Framework -

Time series analysis: Prediction tool

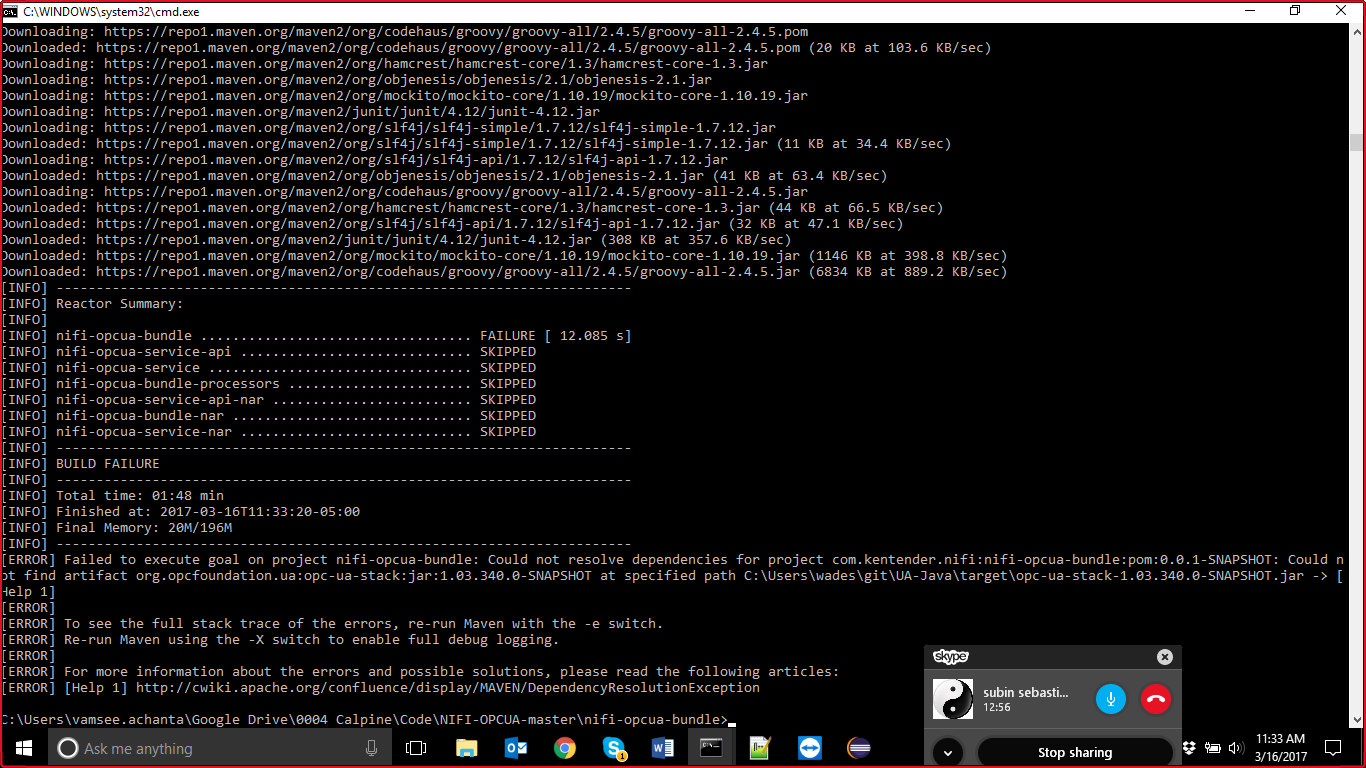
Data Cleansing

Data Process

Visualization

– Maven

* Download Apache Maven
* Add path (environment variables) for Maven program
* Build or compile package
* Missing package as shown below
  + C:\Users\wades\git\UA-Java\target\opc-ua-stack-1.03.340.0-SNAPSHOT.jar
* Troubleshooting now



Also please comment out test classes if you are getting errors there

Amazon S3

<https://aws.amazon.com/blogs/big-data/tips-for-migrating-to-apache-hbase-on-amazon-s3-from-hdfs/>

Latency

https://aws.amazon.com/blogs/big-data/low-latency-access-on-trillions-of-records-finras-architecture-using-apache-hbase-on-amazon-emr-with-amazon-s3/

- OTC 2017 Big Data Lessons

Structured vs. Unstructured data

O&G needs unstructured data ingestion solutions to help transition into Big Data.

Maintenance of digital twins for upstream assets.

Engineering Algorithms

Drilling technology

* Weight on BIT
* FIT

Reservoir Data

Incorporation of analytical models for structural integrity

* Fatigue
* Shaft
* Bearings
* Other structures
* Fitness for service per API-579 and ASME-FFS1

Metocean Data

* Forecast data
* Historical or hindsight data to prepare prediction models

Standardization Solutions

Structural engineering

Aspect ratios

Analytical models

BOEM Data

NOAA (NDBC) data

Asset Framework

Asset frameworks using PI system tags

https://www.linkedin.com/pulse/building-af-models-enterprise-hoon-chew-toh?trk=hp-feed-article-title-like O&G Data Analysis

Competition

<https://youtu.be/h-MpdIvLv-M> GE Digital Marine Solutions

https://www.qubole.com/blog/product/big-data-analytics-at-the-tip-of-your-tongue/ Data Analytics with speech recognition

http://hortonworks.com/blog/modern-oil-gas-architectures-built-hadoop/ Hadoop in Oil and Gas

Type of Analytics:

Descriptive

Predictive

Prescriptive

<https://www.linkedin.com/pulse/big-data-analytics-descriptive-vs-predictive-naveen-joshi?trk=v-feed&lipi=urn%3Ali%3Apage%3Ad_flagship3_feed%3BLhsknN6KWoZOHFN03a%2FJ8A%3D%3D>

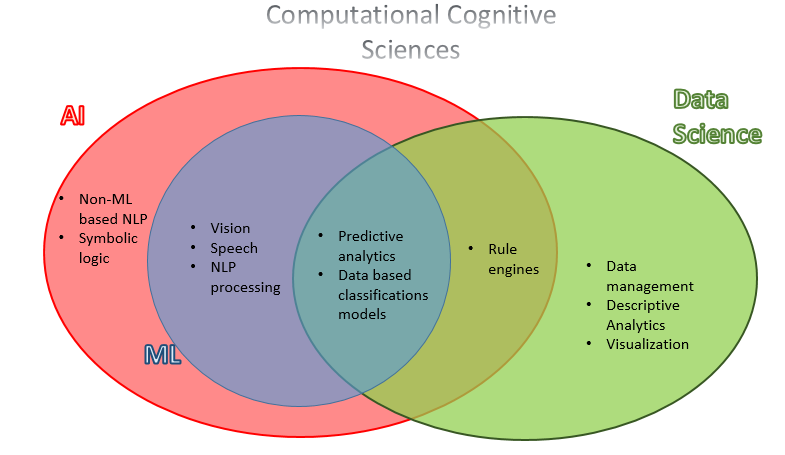
Spark

1. Understand ETL logic from Informatica mappings and rebuild logic in Spark

2. Develop Python code in Spark to execute complex transformations on Amazon EMR

3. Develop end to end flow from source systems to Redshift EDW and optimize where necessary.

<http://www.drillingcontractor.org/selective-standardization-approach-could-help-industry-sidestep-market-barriers-achieve-full-automation-42592> Selective Standardization



Common Machine Learning Algorithms (in python and spark.)

* k-mean clustering
* k-nearest neighbors
* support vector machines
* Gaussian Naive Bayes

<https://en.wikipedia.org/wiki/K-means_clustering>

<https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>

1. Petroleum Data Analytics

2. SPE 24282 (Object Oriented Data Management)

3. SPE 24442 (Data Management: A Case Study).

– Data Science

Data science key learnings are:

– Data structures (local and distributed)

– Data indexing

– Data privacy and anonymization

– Data lifecycle management

– Data transformation (deduplication, handling outliers and missing values, dimensionality reduction)

– Data analysis (experiment design, classification, regression, unsupervised methods)

– Machine learning methods (feature engineering, regularization, hyperparameter tuning, ensemble methods and neural networks)

– Computer and database programming, numerical optimization

– Distributed data processing

– Real-time and high-frequency data processing

– Linux (my personal bias)

A data scientist also has to be a good popularizer of complex ideas.

https://www.linkedin.com/pulse/how-get-firm-foundation-data-science-jt-kostman-phd/

* python - what to learn
* pandas manipulating data, in dataframes, similar to R
* numpy math, multidimensional arrays , matrices, statistical analysis,
* SciPy matrices, optimization, fast computing
* matplotlib graphs
* scikitlearn good documentation
* ipython, jupiter notebook interactive computing, sharing

- Prognostics

Sample Data

<https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/>

- NewField

NewField Exploration Co.

objects as Middleware

ODBC

JDBC

Spark -> is tuned for digital world. Not so good time series algorithms

Data Lake (Building)

- All data should be here

How to bring the siesmic data

Analytics will be where Siesmic data can be read live and exploited (Store & use)

- Work project specific

- Work play specific

- Patterns/trends

- Normalize to understand the trends

Time-depth conversion issues -> use interplotation

Mapping: Fuzzy Mapping

Newer network of relationships

E&P company - 1 B tags per day.

– Deep Learning Course

Deep Learning is useful for making predictions.

Course Projects

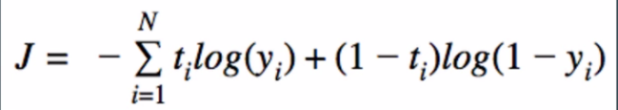
Binary classification to Multi-class Classification

Bionomial/Binary Logistic Regression (also known as Nueron)

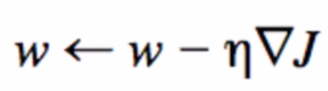
P(y = 1 | x ) = σ (wT x)

Cross-Entropy Cost Function for Binary classification. It is also the negative log-likelihood of the model outputs

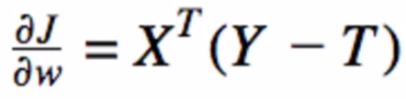
J = -



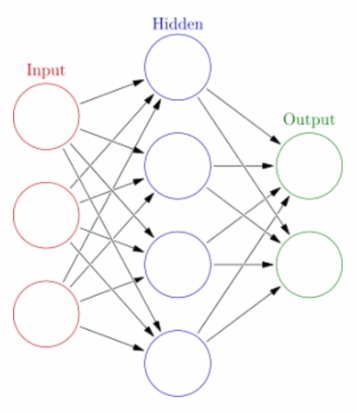
Gradient Decent used to train the logistic regression model



Gradient of the cost function under the logistic regression model.

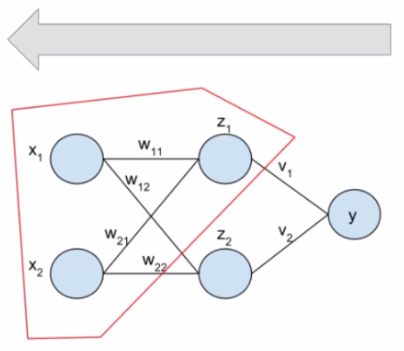


Supervised Learning

* All supervised learning models have 2 basic functions:
  + train() - learn model params from the data (X, Y)
  + predict() - make accurate predictions using the params learnt during training (Predict Y using X)
* The unsupervised models are:
  + Logistic regression
  + K-nearest neighbour
  + Naïve Bayes
  + SVM
  + Decision tree
  + Neural networks
* A neural network example is shown below
  + 

Back Propagation

Back propagation learning is that the error gets propagated backwards

* ssss
  + adsfa
  + 
  + V depends on the error at Y
  + W depneds on error at Z
  + So on…
* The weights gets updated based on the propagated error

Build Neural Network

TensorFlow

Course Project

Engineer at a Inventory company

Facial Expression Recognition

- Digital Twins

https://www.linkedin.com/pulse/licengineering-akselos-first-market-build-digital-twins-leurent Digital Twins for O&G

http://www.mcdermott-investors.com/phoenix.zhtml?c=96360&p=irol-newsArticle&ID=2256535 McDermott Digital Twin

http://www.mckinsey.com/industries/oil-and-gas/our-insights/five-strategies-to-transform-the-oil-and-gas-supply-chain?cid=other-soc-lkn-mip-mck-oth-1607&kui=mioU\_LV6v5Nq2MrUiRjjKA O&G Strategies

http://fortune.com/2017/05/12/warwick-energy-kate-richard/ shale and the associated economics

https://phys.org/news/2017-05-analysis-trigger-shale-gas-revolution.html Data Ananlytics

Interview Questions:

<https://www.dezyre.com/article/100-data-science-interview-questions-and-answers-general-for-2017/184>

DYSYS

Rob Mushank

Doug Johnson

Shawn Mathews

Using Kongsberg technology

Riser Data

Fatigue Data

Standard analysis work

Engineering

Data science and data analytics

Need Project manager and documentation (Requirements)

VIV

Potential meeting soon.

Contract arrangement is directly with AceEngineer inc.

- R

If #rstats programming gets you bored, you can easily switch to do Latex stuff, in fact very complex Latex stuff, equations, graphics, documents, that later can be included within your #rstats publication. Publication is another of #rstats strongholds. Notebooks is just a start. You can publish a web version of your package, write papers, write books, publish your book as a web-book, or publish your application with Shiny, with no html or JavaScript at all.

R Field Guide

<http://fg2re.sellorm.com/about.html>

- Finance Data

<https://lectures.quantecon.org/py/> A good python finance course

Some examples which can help you define your data:

https://www.kaggle.com/szrlee/stock-time-series-20050101-to-20171231

https://www.kaggle.com/dgawlik/nyse

- Analysis (Technical, Historical, Artificial Intelligence etc.)

- Outputs : Storing outputs in database? or otherwise

- Visualizations : python modules of Bokeh or Plotly are web compatible as well. Good starting modules.

https://www.robinhood.com/ Free online trading account

https://github.com/QuantEcon/QuantEcon.py

- Chemistry

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

<https://biopython.org/>

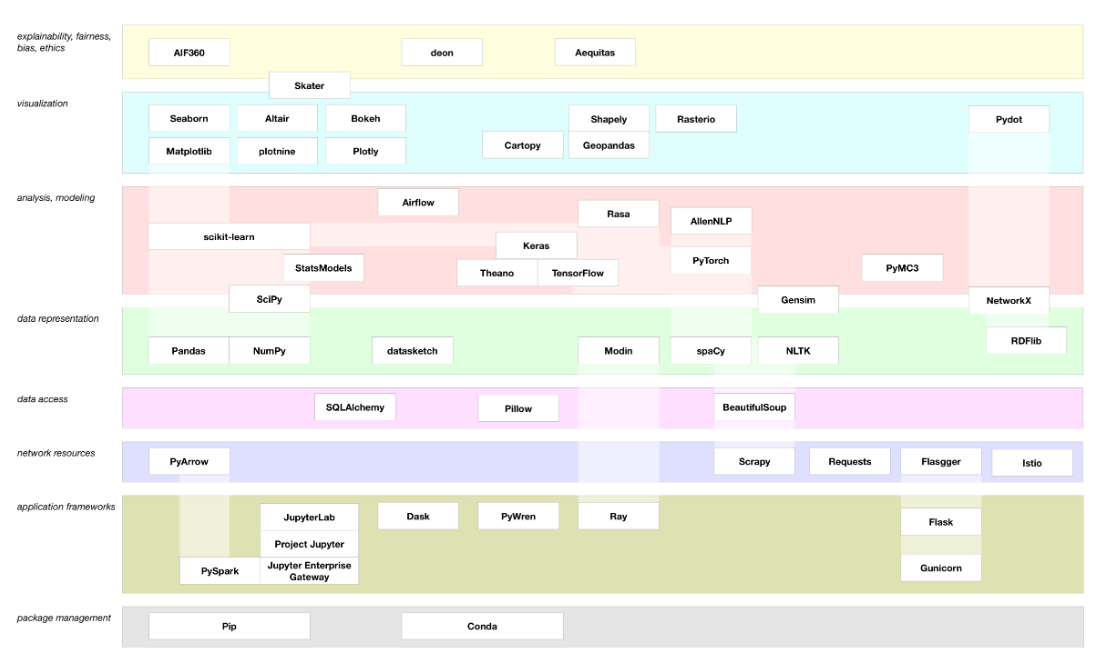
– Sports Case Studies

<https://fivethirtyeight.com/features/how-mapping-shots-in-the-nba-changed-it-forever/>

<https://github.com/abresler/nbastatR>

<https://indianexpress.com/article/sports/cricket-world-cup/rocket-science-behind-jasprit-bumrah-art-5731805/>

- Python for Data Science



Python and Open Server

<https://www.linkedin.com/pulse/guide-using-python-petroleum-experts-openserver-sam-cotterill/>

R Studio

Deploying R Studio applications may be best in Dockers

<https://medium.com/@OilGains/granular-control-of-r-shiny-apps-with-docker-containers-and-shinyproxy-6f2df572b42b>