

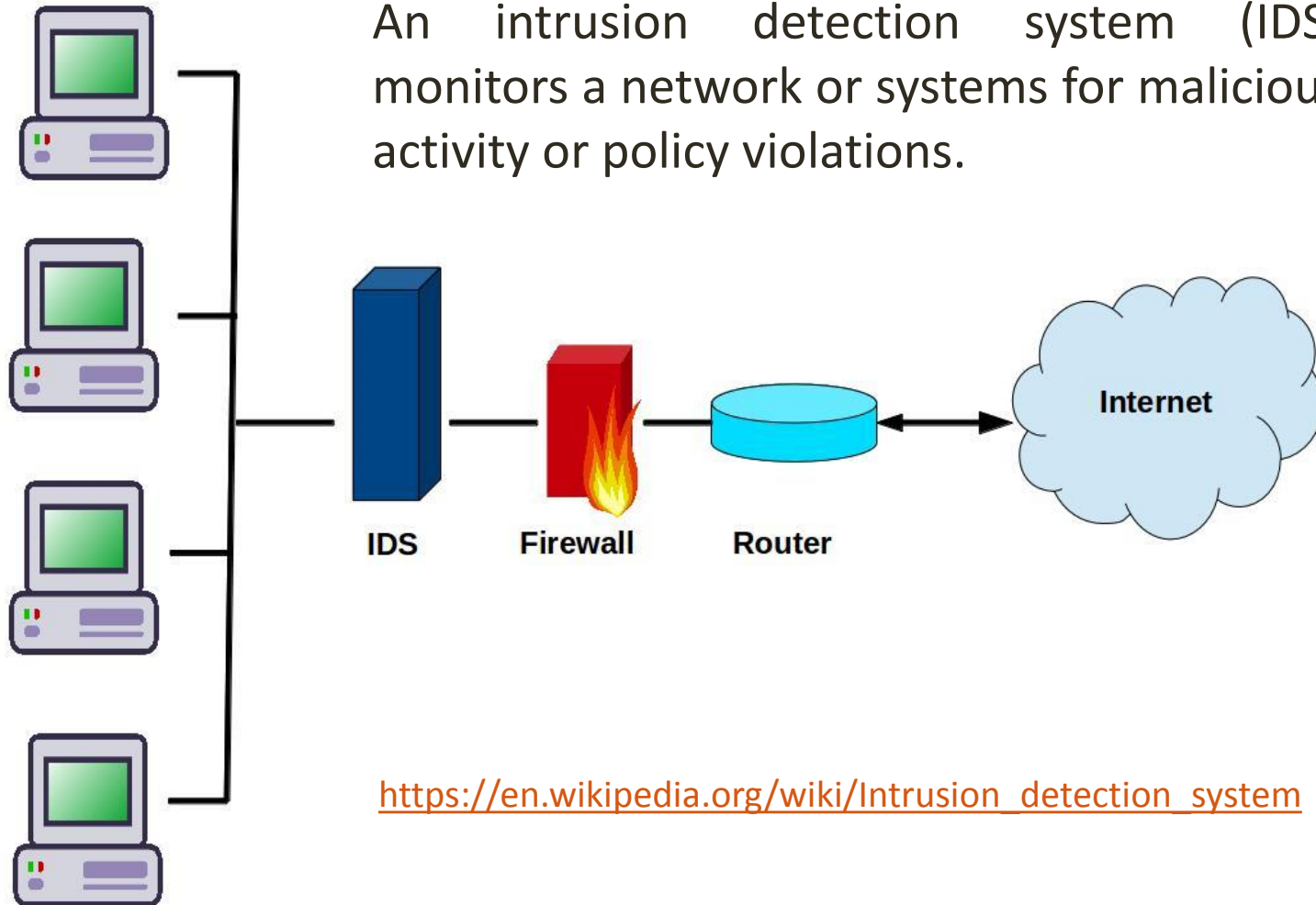
Intrusion Detection Systems (IDS)

Simeon Tsvetanov

simeon.tsvetanov@gmail.com

Definition and applications

An intrusion detection system (IDS) monitors a network or systems for malicious activity or policy violations.



https://en.wikipedia.org/wiki/Intrusion_detection_system

Detection methods

- Signature/Rule bases

Uses certain rules to define what is normal and flags if any of the rules has been triggered.

- Machine Learning

Create a model of trustworthy activity, and then compare new behavior against this model. Used to train classifiers in order to recognize the normality.

Rule bases methods

- Pros:
 - Easy to reason (what causes the anomaly)
 - Simple and understandable
 - Can be dynamic/adaptive
- Cons:
 - Usually not adaptive to traffic changes

Rule bases example

- Similar to firewalls

#	Policy	Protocol	Destination	Port	Comment	Actions
1	Deny	TCP	Any	25	Block SMTP	↕ X
2	Deny	TCP	Any	6881	Block BitTorrent	↕ X
3	Allow	TCP	192.168.1.37/32	Any	Access to student printer	↕ X
	Deny	Any	Local LAN	Any	Wireless clients accessing LAN ⓘ	
	Allow	Any	Any	Any	Default rule	

- Rules are usually created by person and are not adaptable to traffic changes.

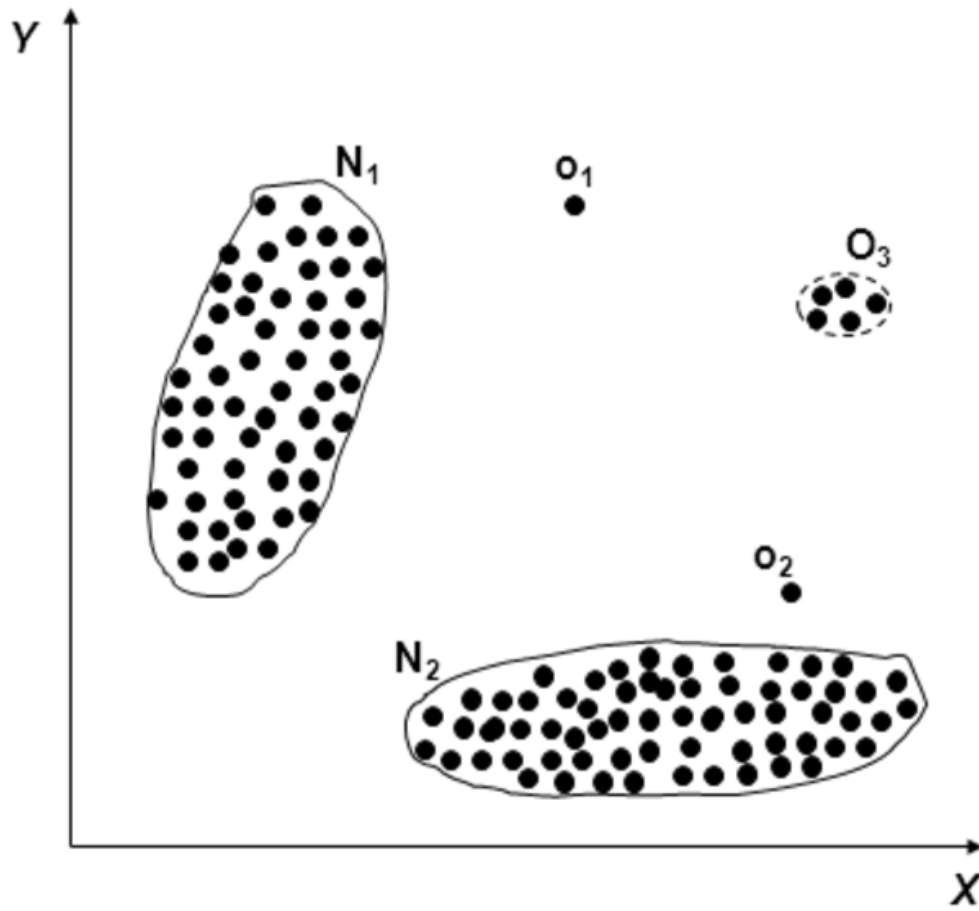
alert tcp !\$PRIVATE_NET any -> 192.168.56.5 80 (msg: "Login attemp")

Machine Learning methods

- Pros:
 - Adaptive to changes
 - Minimal human intervention
- Cons:
 - Semantic gap

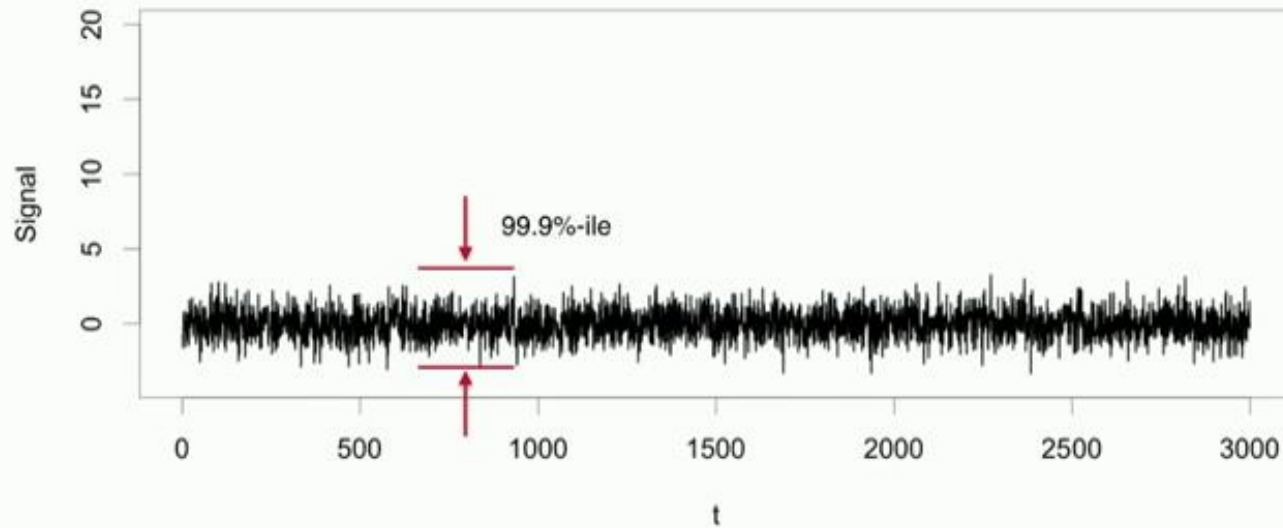
Machine learning example

- Train the model
- Evaluate input data



Other methods

- Spectral analysis (FFT, Wavelets)
- Kalman filter
- etc.



Anomaly Detection

- In data mining, anomaly detection is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.
- Typically the anomalous items will translate to some kind of problem such as network intrusion, bank fraud, etc.

		Predicted label	
		+	-
True label	+	True Positive	False Positive
	-	False Negative	True Negative

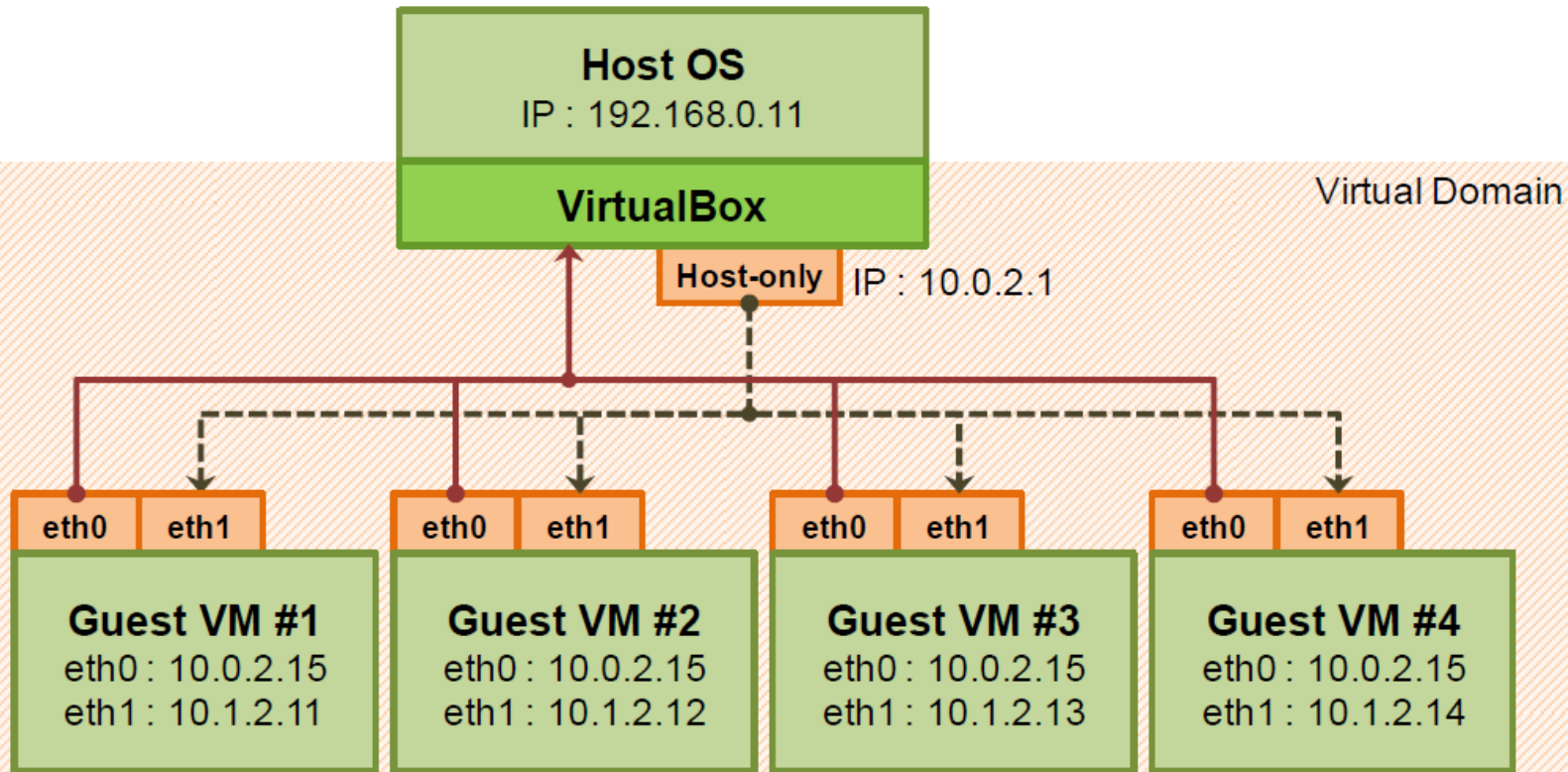
Anomaly types

- Contextual Anomalies
- Collective Anomalies
- Point anomalies & Group Anomalies
- Behavior based Anomaly Detection

Anomaly Detection - Challenges

- Security systems are very intolerant to errors
- Semantic gap
- Lack of training data
- Difficulties in evaluation
- A lot of research, but not many successful systems available

Development Environment

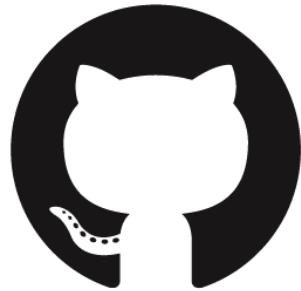


<http://www.slideshare.net/powerhan96/networking-between-host-and-guest-v-vm-in-virtual-box>

Development Environment

- Host – 192.168.56.1
- VM1 -192.168.56.101
- VM1 -192.168.56.102
- VM1 -192.168.56.103

Development environment - Code



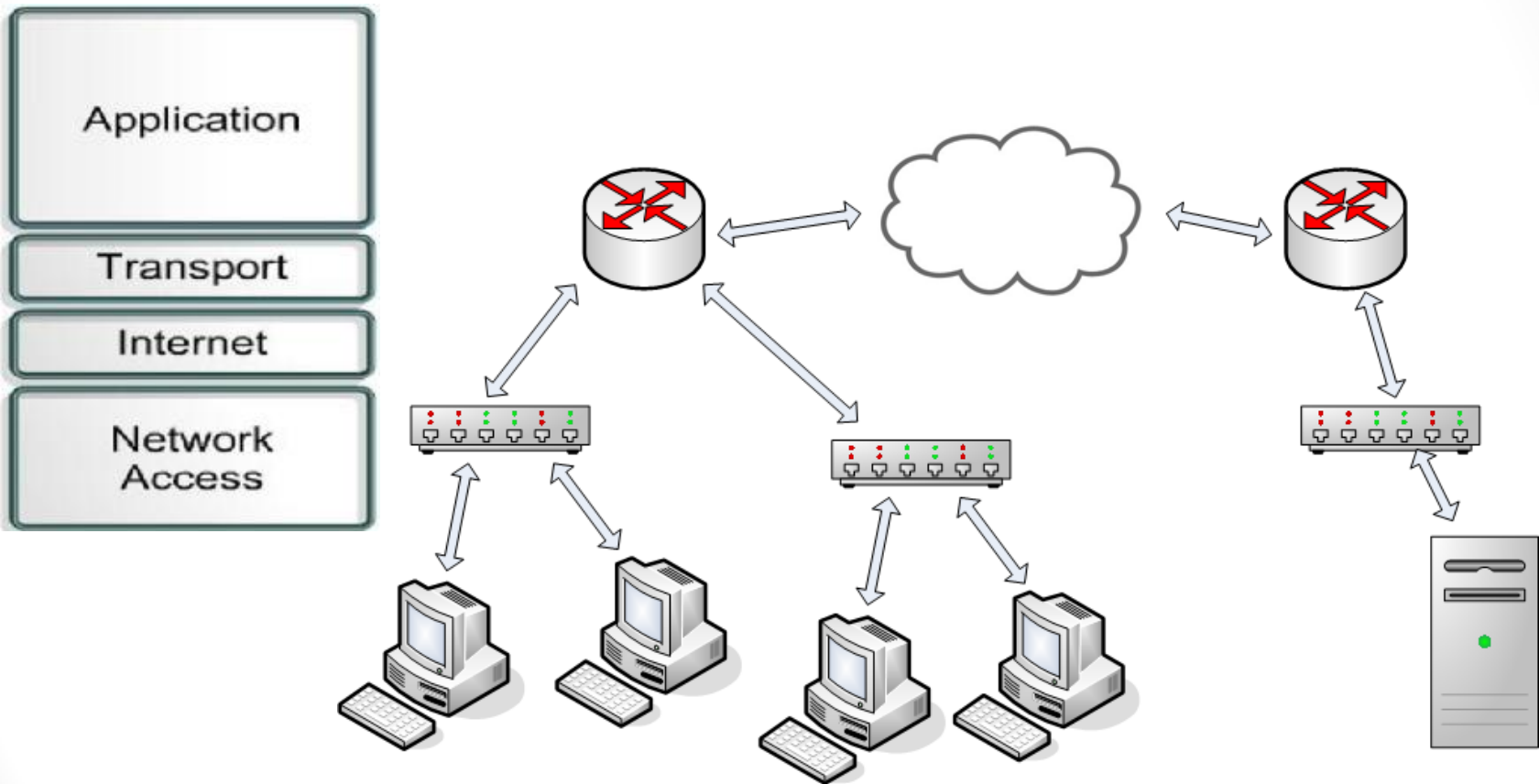
GitHub



GitLab

<https://about.gitlab.com/downloads/#ubuntu1604>

Introduction to computer networks



tcpdump

- **tcpdump** is a common packet analyzer that runs under the command line and allows to save the captured packets for future analysis.

```
sudo tcpdump -i enp0s8 > mylog.txt
```

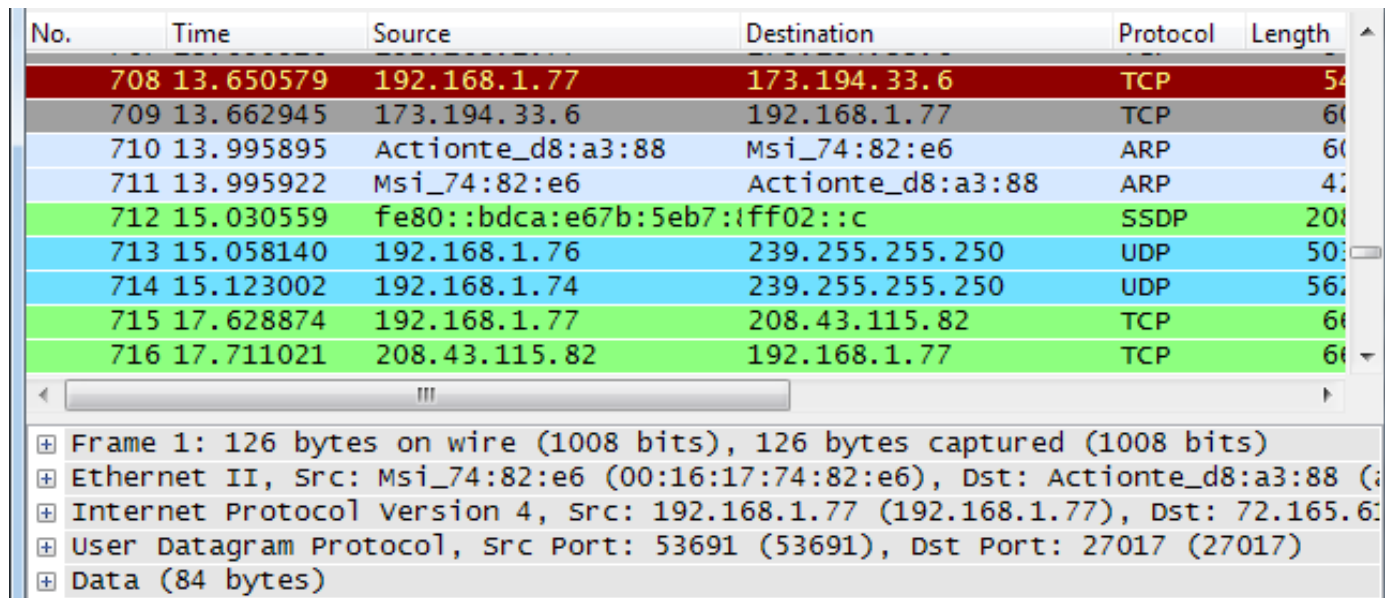
```
susel:~ # tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
20:39:28.014065 IP 192.168.198.1.netbios-ns > 192.168.198.255.netbios-ns: NBT U
DP PACKET(137): QUERY; REQUEST; BROADCAST
20:39:28.014840 IP 192.168.198.128.56851 > 192.168.198.2.domain: 18867+ PTR? 25
5.198.168.192.in-addr.arpa. (46)
20:39:28.027418 IP 192.168.198.1.49733 > 224.0.0.252.llmnr: UDP, length 22
20:39:28.027850 IP 192.168.198.128.50611 > lhr14s24-in-f19.1e100.net.https: P 2
912329209:2912329246(37) ack 1375935787 win 18760
20:39:28.034322 IP lhr14s24-in-f19.1e100.net.https > 192.168.198.128.50611: . a
ck 37 win 64240
20:39:28.037196 IP6 fe80::2cfe:5154:6c0d:fafd.65460 > ff02::1:3.llmnr: UDP, len
gth 22
20:39:28.039057 IP 192.168.198.1.65460 > 224.0.0.252.llmnr: UDP, length 22
20:39:28.051576 IP 192.168.198.2.domain > 192.168.198.128.56851: 18867 NXDomain
0/1/0 (95)
20:39:28.051744 IP 192.168.198.128.35496 > 192.168.198.2.domain: 58919+ PTR? 1
```


Wireshark

- Wireshark is a free and open source packet analyzer

www.wireshark.org

<http://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/>



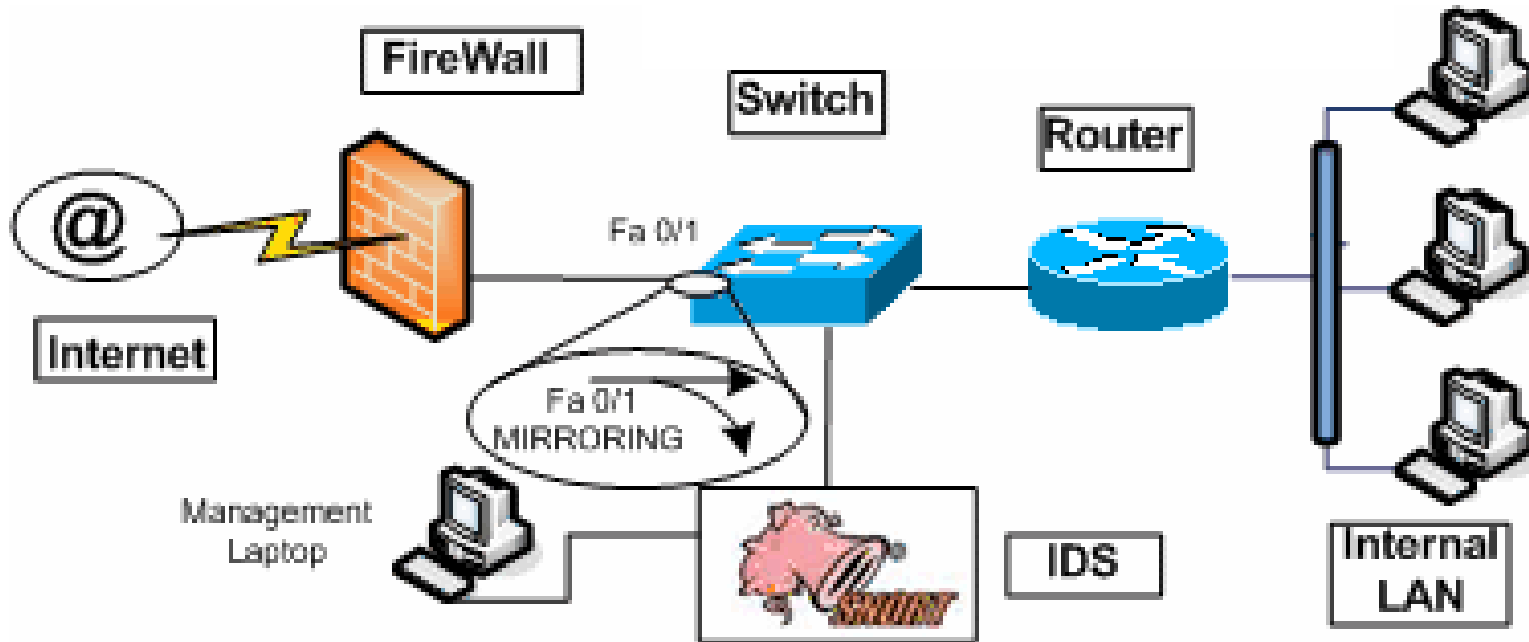
No.	Time	Source	Destination	Protocol	Length
708	13.650579	192.168.1.77	173.194.33.6	TCP	54
709	13.662945	173.194.33.6	192.168.1.77	TCP	60
710	13.995895	Actionte_d8:a3:88	Msi_74:82:e6	ARP	60
711	13.995922	Msi_74:82:e6	Actionte_d8:a3:88	ARP	42
712	15.030559	fe80::bdca:e67b:5eb7:1ff02::c		SSDP	200
713	15.058140	192.168.1.76	239.255.255.250	UDP	50
714	15.123002	192.168.1.74	239.255.255.250	UDP	56
715	17.628874	192.168.1.77	208.43.115.82	TCP	60
716	17.711021	208.43.115.82	192.168.1.77	TCP	60

Frame 1: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits)
Ethernet II, Src: Msi_74:82:e6 (00:16:17:74:82:e6), Dst: Actionte_d8:a3:88 (08:00:27:08:00:27)
Internet Protocol Version 4, Src: 192.168.1.77 (192.168.1.77), Dst: 72.165.67.134 (72.165.67.134)
User Datagram Protocol, Src Port: 53691 (53691), Dst Port: 27017 (27017)
Data (84 bytes)

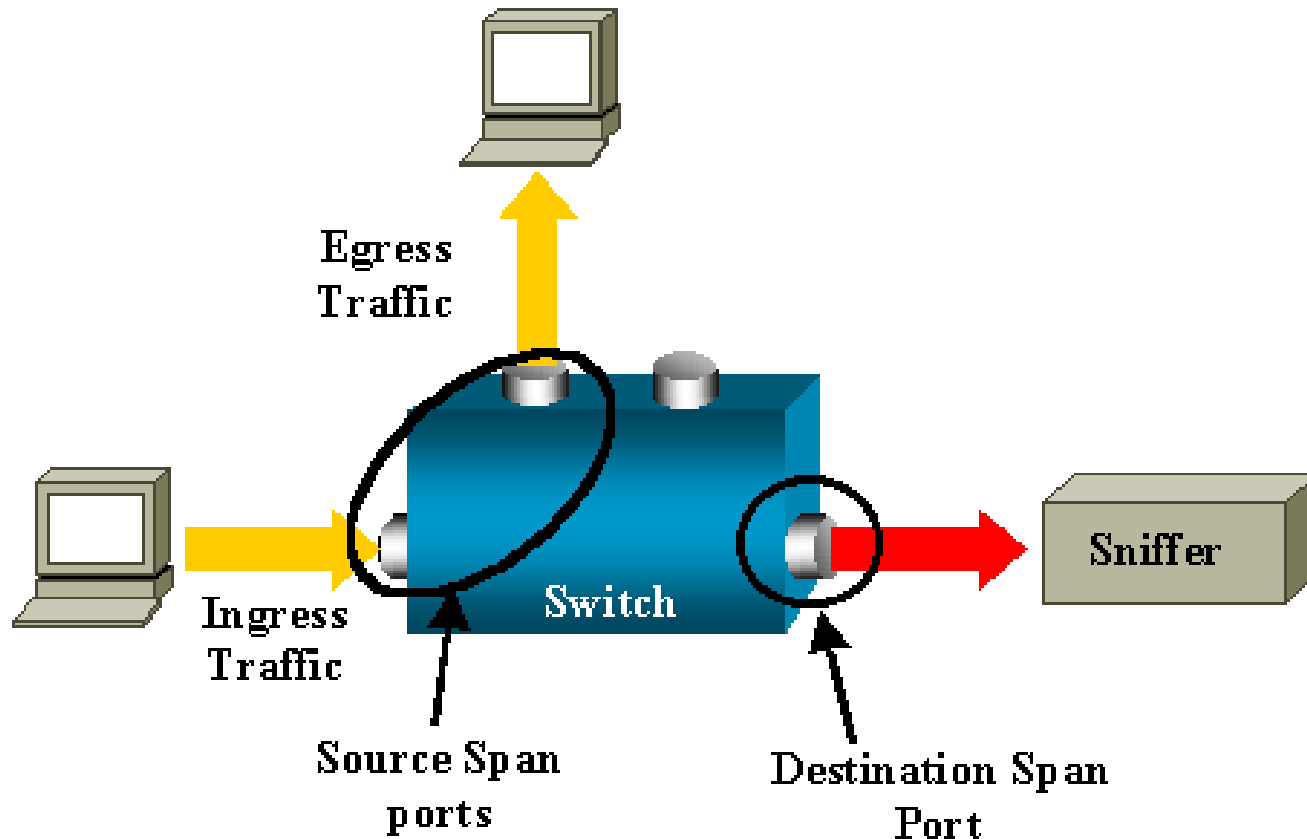
- `sudo tcpdump -i enp0s8 -w mycap.pcap`
- `tshark -r ./mycap.pcap -V`

Port mirroring

- If enabled, the switch sends a copy of all network packets to one port, where the packets can be analyzed.

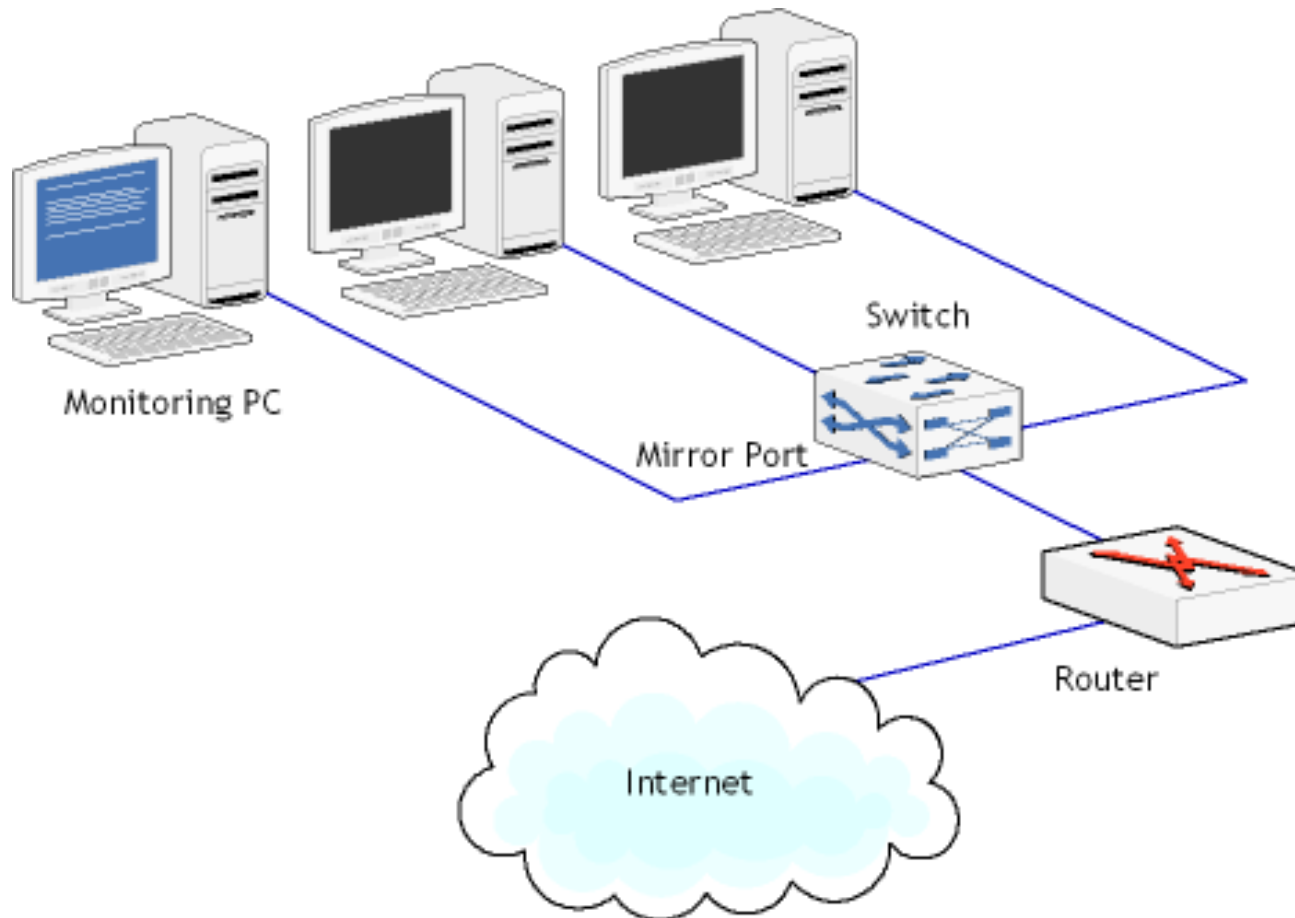


Port mirroring

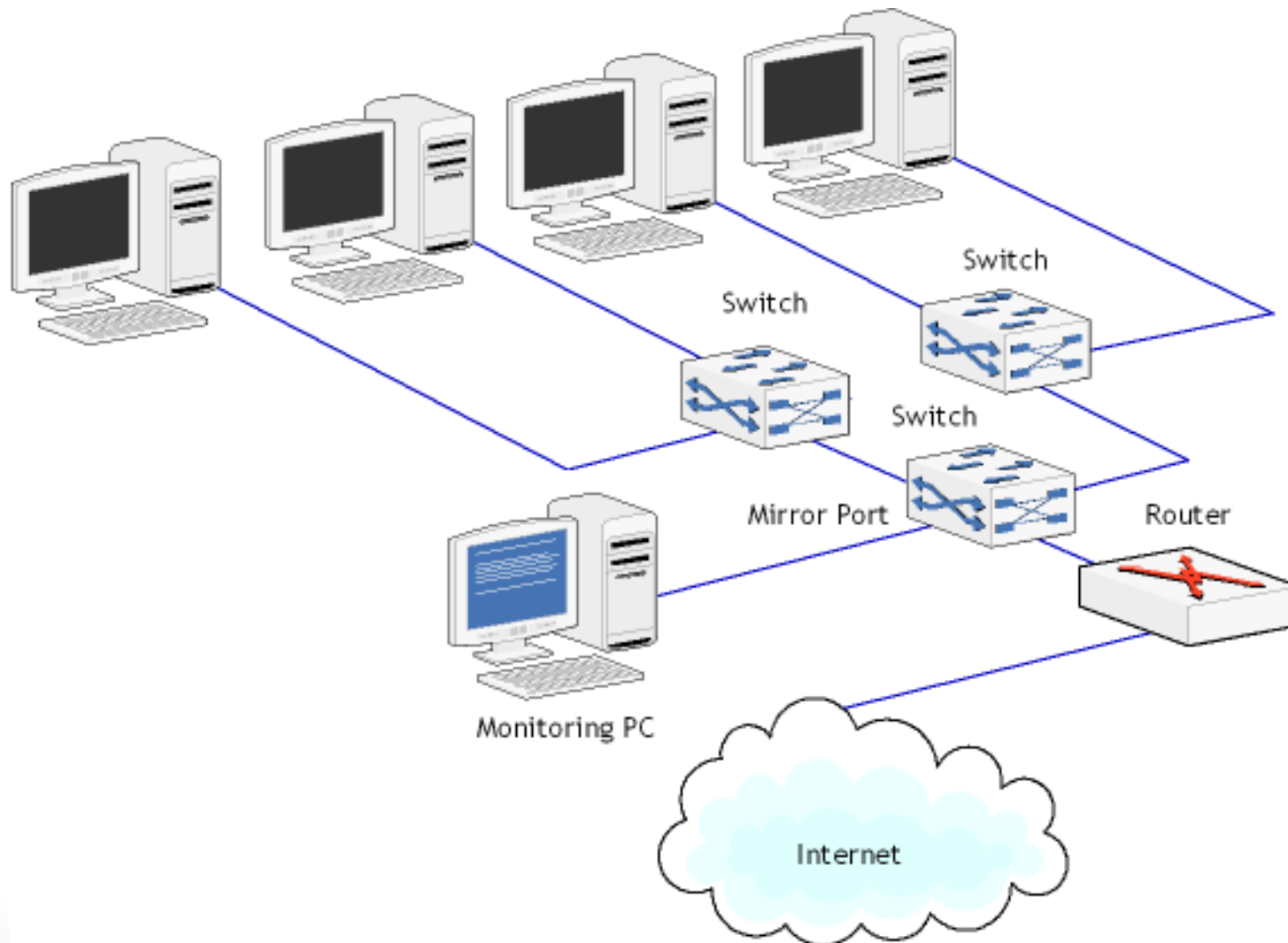


<http://www.cisco.com/c/en/us/support/docs/switches/catalyst-6500-series-switches/10570-41.html>

Port mirroring



Port mirroring



Firewall

ufw is the default firewall configuration tool for Ubuntu.

- To check the ufw status
`sudo ufw status verbose`
- To open a port (ssh in this example):
`sudo ufw allow 22`
- To close a port
`sudo ufw deny 80`
- To add add service by name
`sudo ufw allow ssh`
- to disable/enable ufw
`sudo ufw disable`
`sudo ufw enable`

Snort IDS

- Snort is a free and open source network intrusion detection system (NIDS) <https://www.snort.org/>
- Plugins available <https://www.threatstack.com/> etc.
- <https://www.snort.org/downloads#rules>



Snort IDS

- Snort installation

```
sudo apt-get install snort
```

- Snort rules tutorials

<http://books.gigatux.nl/mirror/snortids/0596006616/snortids-CHP-7-SECT-3.html>

<http://archive.oreilly.com/pub/h/1393>

https://www.snort.org/rules_explanation

- Rules location

```
/etc/snort/rules
```


Create own rules

- /etc/snort/rules/myrules.rules

```
alert icmp any any -> any any (msg: "ICMP Testing Rule";  
sid:1000001; rev:1;)
```

```
alert tcp any any -> any 80 (msg: "TCP Testing Rule"; sid:1000002;  
rev:1;)
```

```
alert udp any any -> any any (msg: "UDP Testing Rule";  
sid:1000003; rev:1;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN  
FIN";flags:SF; reference: arachnids,198; classtype:attempted-recon;  
sid:624; rev:1;)
```

Rule Actions

<http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node29.html>

- alert - generate an alert using the selected alert method, and then log the packet
- log - log the packet
- pass - ignore the packet
- activate - alert and then turn on another dynamic rule
- dynamic - remain idle until activated by an activate rule , then act as a log rule
- drop - block and log the packet
- reject - block the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.
- sdrop - block the packet but do not log it.

Snort Example

- `/etc/snort/snort.conf`

```
include $RULE_PATH/myrules.rules  
output alert_full: alert.full
```

- Run

```
sudo /usr/sbin/snort -d -l /var/log/snort/ -c /etc/snort/snort.conf -i  
enp0s8
```

- read alert file

```
tail -f . /var/log/snort/alert.full
```

Snort on DARPA Dataset

- DARPA Datasets

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

<https://www.ll.mit.edu/ideval/data/1999data.html>

<https://www.ll.mit.edu/ideval/docs/id99-eval-ll.html>

<https://www.ll.mit.edu/ideval/data/1999/testing/week4/index.html>

<https://www.youtube.com/watch?v=OA4hSFxyXXU>

- `gunzip outside.tcpdump.gz`

- Run Snort on DARPA dataset

```
sudo snort -r /home/node1/projects/datasets/darpa.tcpdump -c /etc/snort/snort.conf
```

Snort Service

- `sudo service snort status`
- `sudo service snort start`
- `sudo service snort stop`
- <https://softuni.bg/trainings/1488/linux-system-administration-november-2016>

Commercial Products

- Sourcefire <https://en.wikipedia.org/wiki/Sourcefire>

Cisco FirePOWER appliances, Next-Generation IPS

- IBM [Security Network Intrusion Prevention System](#)
- [McAfee Network Security Platform](#) (NSP)
- Radware [DefensePro](#)
- etc.

Cisco Firepower 9300

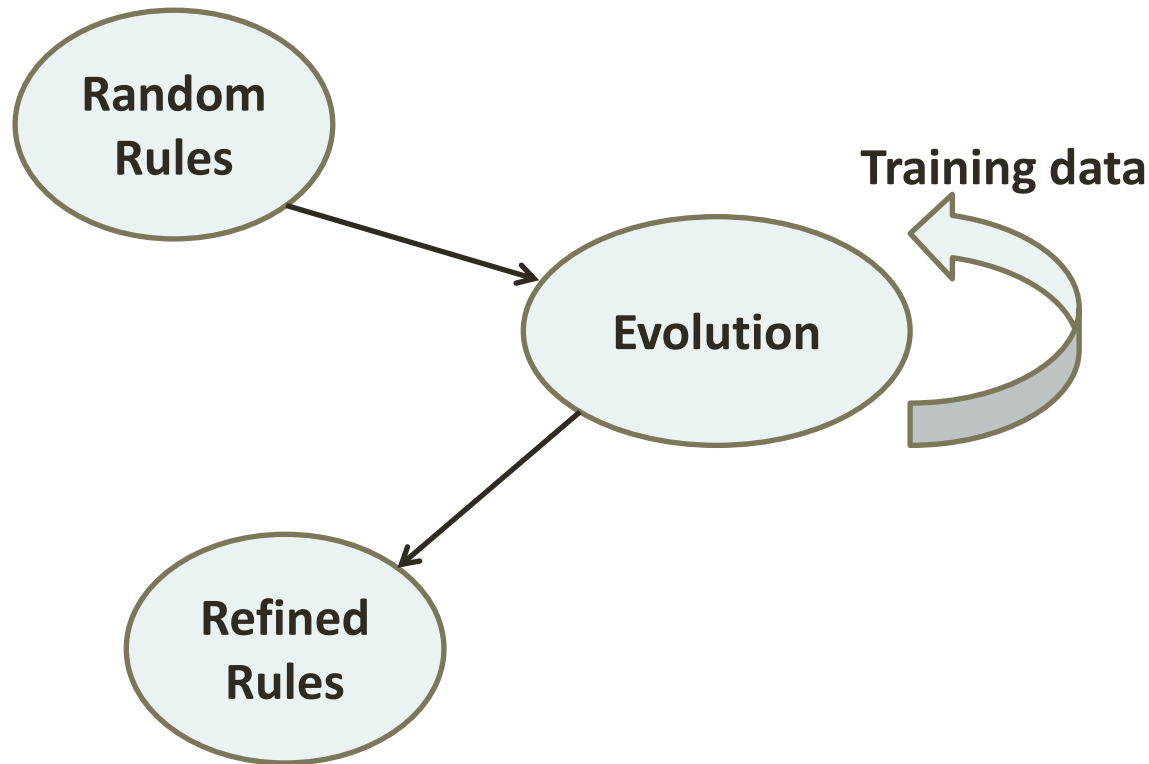


- 10/40/100 Gb Network Interfaces
- 57 million concurrent connections, with application control
- 500,000 new connections per second
- Security services options: AVC, NGIPS, AMP, URL Filtering, DDos Mitigation

Hacking tools and tutorials

- hackthissite.org - Good for practices
- exploit-db.com - Vulnerability database
- kali.org – Penetration tools
- <https://www.facebook.com/ethicalhackingnewsandtutorials/>
- Security Analysis of Estonia's Internet Voting System by Alex Halderman https://www.youtube.com/watch?v=JY_pHvhE4os
- etc.

Generate Rules



<http://www.brie.com/brian/netga/>

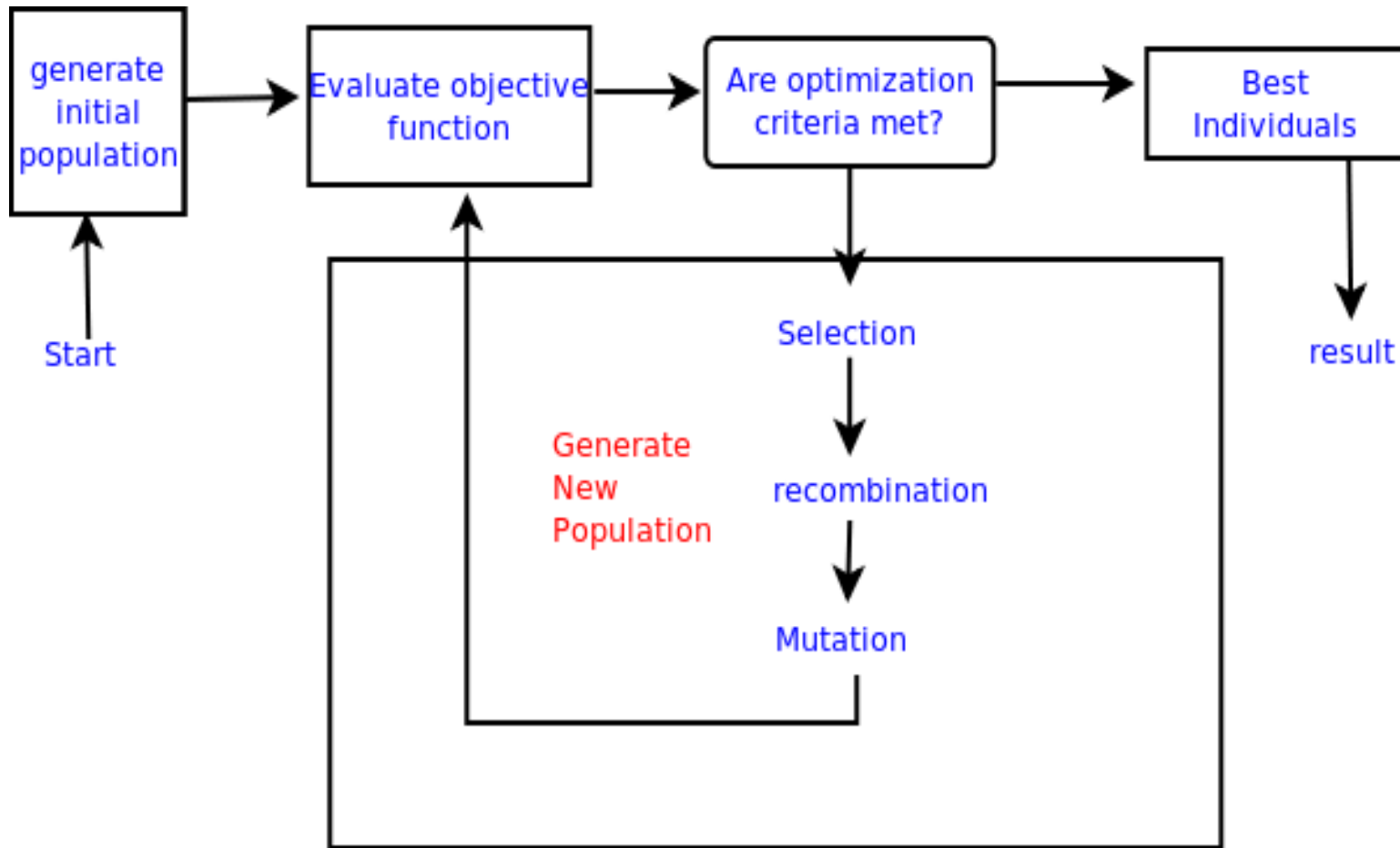
<https://www.youtube.com/watch?v=KjowmnXHse4>

Ren Hui Gong, Mohammad Zulkernine, Purang Abolmaesumi, "A Software Implementation of a Genetic Algorithm Based Approach to Network Intrusion Detection"

DARPA Data format

```
55 01/23/1998 16:58:40 00:00:03 smtp 1832 25 192.168.1.30 192.168.0.20 0 -
57 01/23/1998 16:58:39 00:00:02 finger 1834 79 192.168.1.30 192.168.0.20 0 -
65 01/23/1998 16:58:48 00:00:01 finger 1841 79 192.168.1.30 192.168.0.20 0 -
69 01/23/1998 16:58:55 00:00:04 finger 1847 79 192.168.1.30 192.168.0.20 0 -
73 01/23/1998 16:58:58 00:00:18 ftp 1850 21 192.168.1.30 192.168.0.20 0 -
77 01/23/1998 16:59:05 00:00:01 finger 1855 79 192.168.1.30 192.168.0.20 0 -
90 01/23/1998 16:59:22 00:00:22 telnet 1867 23 192.168.1.30 192.168.0.20 1 guess
99 01/23/1998 16:58:58 00:00:03 smtp 43533 25 192.168.0.40 192.168.0.20 0 -
101 01/23/1998 16:59:37 00:00:44 telnet 1876 23 192.168.1.30 192.168.0.20 0 -
106 01/23/1998 16:59:14 00:00:00 smtp 43538 25 192.168.0.40 192.168.0.20 0 -
110 01/23/1998 17:00:00 00:00:23 telnet 1884 23 192.168.1.30 192.168.0.20 1 guess
114 01/23/1998 16:59:30 00:00:01 smtp 43541 25 192.168.0.40 192.168.0.20 0 -
116 01/23/1998 17:00:09 00:01:40 telnet 1890 23 192.168.1.30 192.168.0.20 0 -
118 01/23/1998 17:00:13 00:00:11 ftp 1892 21 192.168.1.30 192.168.0.20 0 -
122 01/23/1998 17:00:31 00:00:00 smtp 1900 25 192.168.1.30 192.168.0.20 0 -
125 01/23/1998 17:00:38 00:00:02 rsh 1023 1021 192.168.1.30 192.168.0.20 1 rcp
126 01/23/1998 17:00:39 00:00:23 telnet 1906 23 192.168.1.30 192.168.0.20 1 guess
128 01/23/1998 17:00:41 00:00:14 rlogin 1022 513 192.168.1.30 192.168.0.20 1 rlogin
136 01/23/1998 17:00:57 00:00:02 rsh 1022 1021 192.168.1.30 192.168.0.20 1 rsh
```

Genetic algorithm overview



Fitness calculation

	Duration			Protocol	SRC PORT	DST PRT	SRC IP				DST IP				Attack Type
	H	M	S				0	1	2	3	0	1	2	3	
1	0	0	11	ftp	1892	21	192	168	1	30	192	168	0	20	-
2	0	0	0	smtp	1900	25	192	168	1	30	192	168	0	20	-
3	0	0	2	rsh	1023	1021	192	168	1	30	192	168	0	20	rcp
4	0	0	23	telnet	1906	23	192	168	1	30	192	168	0	20	guess
5	0	0	14	rlogin	1022	513	192	168	1	30	192	168	0	20	rlogin
6	0	0	2	rsh	1022	1021	192	168	1	30	192	168	0	20	rsh
7	0	0	15	ftp	43549	21	192	168	0	40	192	168	0	20	-
8	0	0	40	telnet	1914	23	192	168	1	30	192	168	0	20	guess
9	0	1	24	telnet	43560	23	192	168	0	40	192	168	0	20	-
10	0	0	13	ftp	43566	21	192	168	0	40	192	168	0	20	-

Chromosome for Individual (-1 is wildcard)

-1	0	-1	rsh	-1	1021	192	168	-1	-1	192	168	0	-1	rsh
----	---	----	-----	----	------	-----	-----	----	----	-----	-----	---	----	-----

Fitness calculation

$N = 10$ connections.

$$|A| = 2$$

$$|A \text{ and } B| = 1$$

$$w1 = 0.2$$

$$w2 = 0.8$$

$$\text{fitness} = w1 * \text{support} + w2 * \text{confidence}$$

$$\text{support} = |A \text{ and } B| / N = 1 / 10 = 0.1$$

$$\text{confidence} = |A \text{ and } B| / |A| = 1 / 2 = 0.5$$

$$\text{fitness} = 0.2 * 0.1 + 0.5 * 0.8 = \mathbf{0.42}$$

Rules Generation Experiments

- Install DEAP <https://github.com/DEAP>

```
sudo apt-get install python3-pip
```

```
sudo pip3 install deap
```

- Generate rules

```
python acega.py
```

- Evaluate input data

```
python testerMod.py
```

Tanapuch Wanwarang, Machigar Ongtang, "Elitism Enhancements for Genetic Algorithm Based Network Intrusion Detection System" (AceGA)

<https://github.com/nixor/GANIDS>

Rules Generation Experiments

Attack Type: ipsweep

Test Data records: 1542781

Total Number of Attacks in Test Records: 3101.0

All alerts: 4186.0

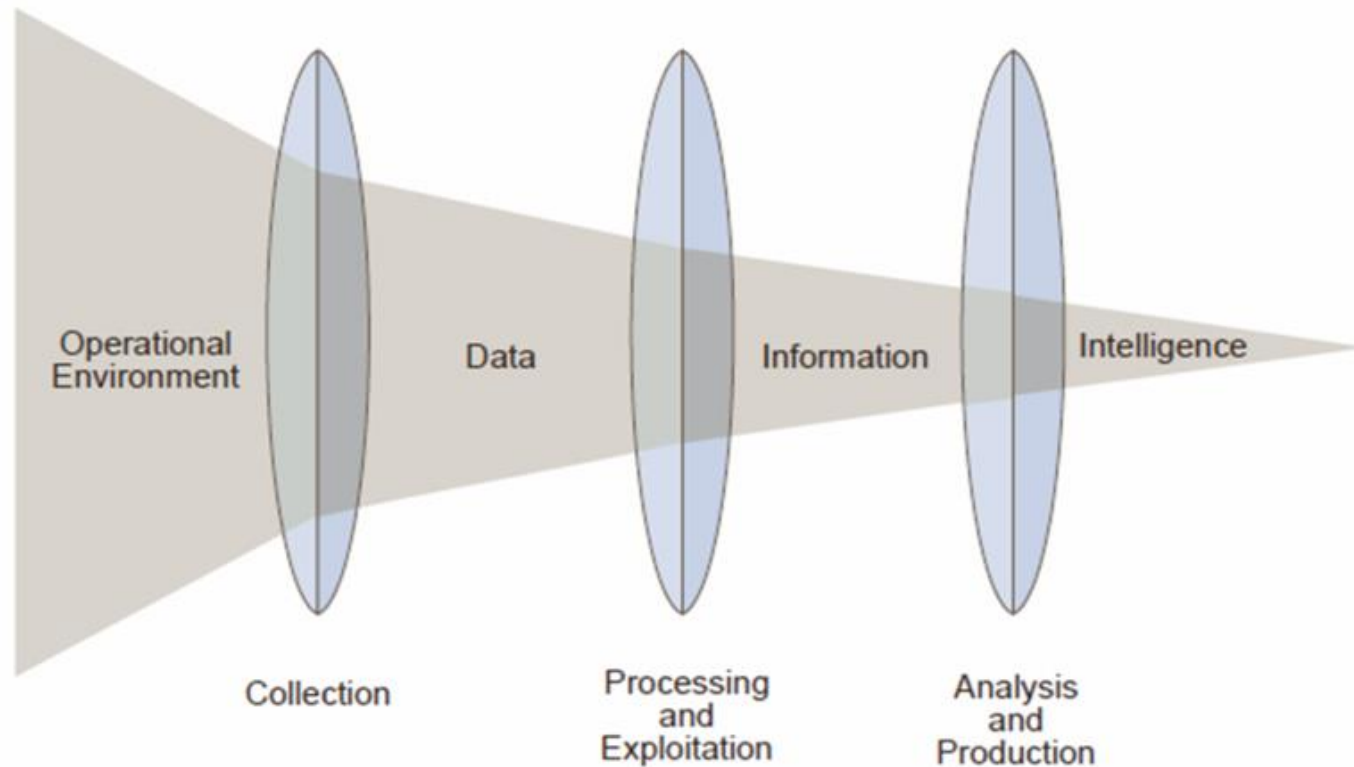
L False Positive/False Alerts: 1144.0, 0.0742%

L False Negative/Undetected Attacks: 59.0, 1.9026%

H True Positive/Detected Attacks: 3042.0, 98.0974%

H True Negative/Normal conn correctly identified: 1538536.0,
99.9257%

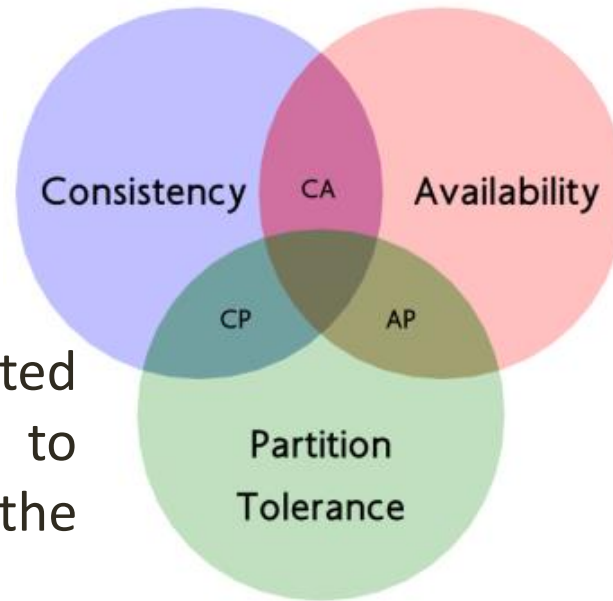
Big Data Analysis



<http://www-01.ibm.com/software/data/infosphere/hadoop/what-is-big-data-analytics.html>

<http://www.sciencedirect.com/science/article/pii/S0268401214001066>

CAP theorem



- It is impossible for a distributed computer system to simultaneously provide all of the three guarantees.
- Hadoop (Have) is optimized for throughput/consistency and works with static immutable data.

<https://dzone.com/articles/better-explaining-cap-theorem>

<http://robertgreiner.com/2014/08/cap-theorem-revisited/>

Big Data Analysis - Map-Reduce

Running parallel programs on distributed systems is complex and challenging process that requires a solid background in computing.

Jeffrey Dean and Sanjay Ghemawat - **MapReduce: Simplified Data Processing on Large Clusters**

MapReduce provides an abstraction and significantly simplifies the process, but it's still complicated.

<http://dl.acm.org/citation.cfm?id=1327492>

<http://static.googleusercontent.com/media/research.google.com/bg//archive/mapreduce-osdi04.pdf>

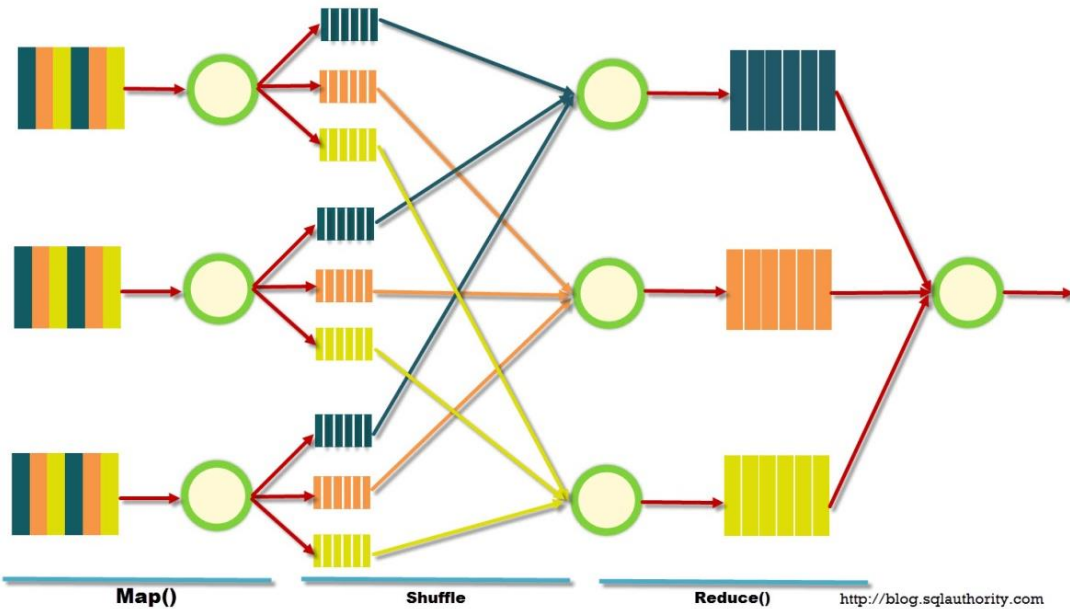
Map-Reduce

$\text{map}(k1, v1) \rightarrow \text{list}(k2, v2)$
 $\text{reduce}(k2, \text{list}(v2)) \rightarrow \text{list}(v2)$

```
map(String key, String value):  
// key: document name  
// value: document contents  
for each word w in value:  
EmitIntermediate(w, "1");
```

```
reduce(String key, Iterator  
values):  
// key: a word  
// values: a list of counts  
int result = 0;
```

```
for each v in values:  
result += ParseInt(v);  
Emit(AsString(result));
```



Counting the number of occurrences of each word in a large collection of documents

Hadoop

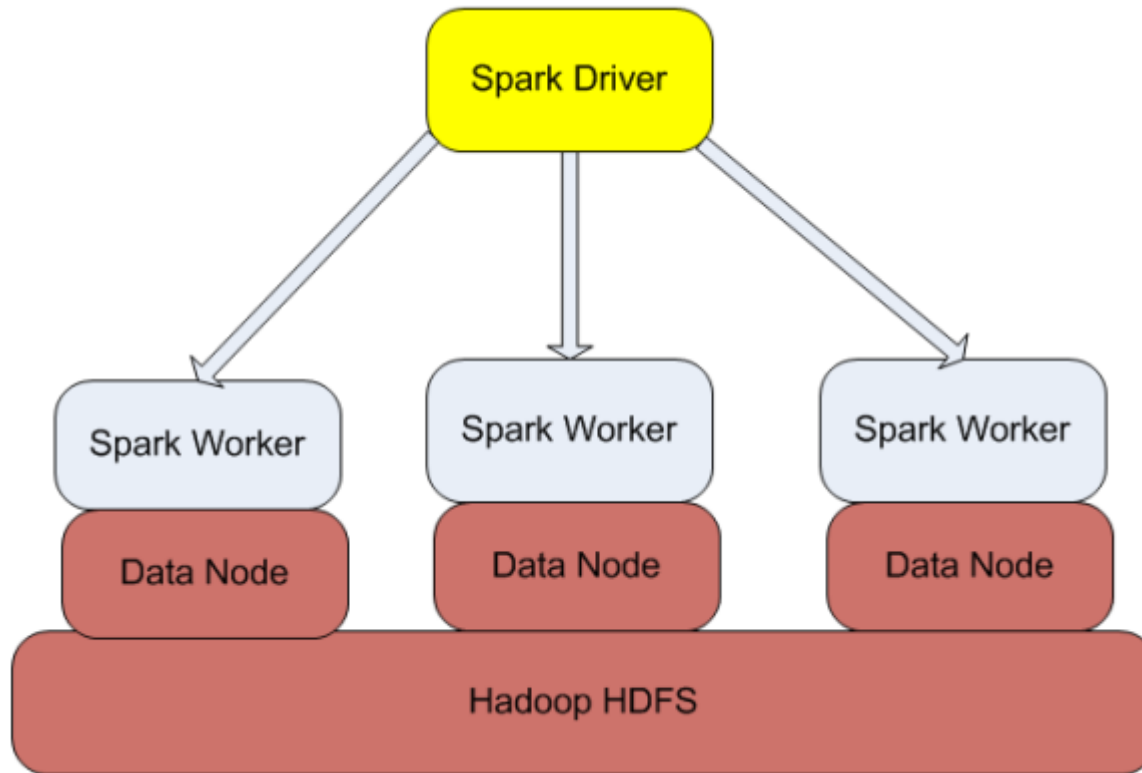
Open source implementation of the Map-Reduce idea initially created by Doug Cutting.

- Storage part (Hadoop Distributed File System (HDFS))
- Processing part (MapReduce) - a programming model for large scale data processing
- *Hadoop YARN* – a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users applications
- Hive – Uses higher (Hive QL) language similar to SQL to simplify and abstract from Map-Reduce. Might say that Hive QL is “compiled” to Hadoop.

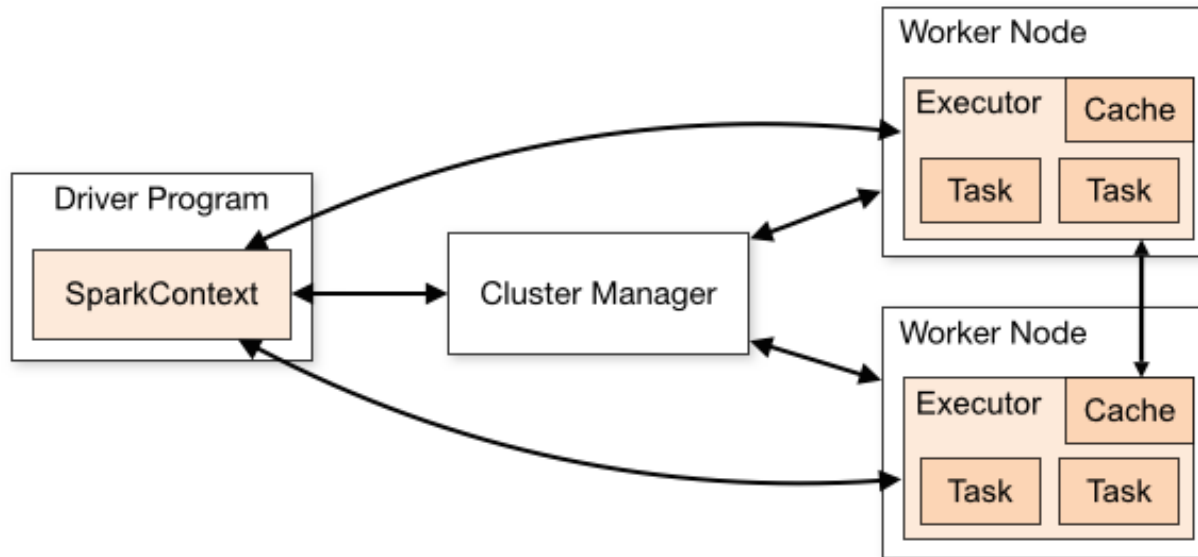
Apache Spark

- Apache Spark is a fast, in-memory data processing engine with expressive development APIs in Scala, Java, and Python that allow data workers to efficiently execute machine learning algorithms that require fast iterative access to datasets.
- For cluster management, Spark supports standalone (native Spark cluster), Hadoop YARN, or Apache Mesos.
- For distributed storage, Spark can interface with a wide variety, including Hadoop Distributed File System (HDFS), Cassandra, OpenStack Swift, Amazon S3, Kudu.

Apache Spark



Spark cluster



- A Spark cluster is composed of one Driver JVM and one or many Executor JVMs.

<http://spark.apache.org/docs/latest/cluster-overview.html>

Spark Cluster - Examples

- An RDD is a collection of items distributed across many nodes that can be manipulated in parallel.
- Run the master
`./sbin/start-master.sh`
- Run the slave
`./sbin/start-slave.sh <master-spark-URL>`
- Open the master web GUI
`master-IP:8080`

<http://blog.insightdatalabs.com/spark-cluster-step-by-step/>

<http://blog.abhinav.ca/blog/2014/04/13/setup-a-spark-cluster-in-5-minutes/>

Spark Cluster - Examples



Spark Master at spark://ip-172-31-38-214:7077

URL: spark://ip-172-31-38-214:7077

REST URL: spark://ip-172-31-38-214:6066 (*cluster mode*)

Workers: 3

Cores: 18 Total, 0 Used

Memory: 8.6 GB Total, 0.0 B Used

Applications: 0 Running, 0 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers

Worker Id	Address	State	Cores	Memory
worker-20150929091656-172.31.38.215-44824	172.31.38.215:44824	ALIVE	6 (0 Used)	2.9 GB (0.0)
worker-20150929091656-172.31.38.216-47400	172.31.38.216:47400	ALIVE	6 (0 Used)	2.9 GB (0.0)
worker-20150929091656-172.31.38.217-39731	172.31.38.217:39731	ALIVE	6 (0 Used)	2.9 GB (0.0)

Running Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Du
----------------	------	-------	-----------------	----------------	------	-------	----

Resilient Distributed Datasets RDDs

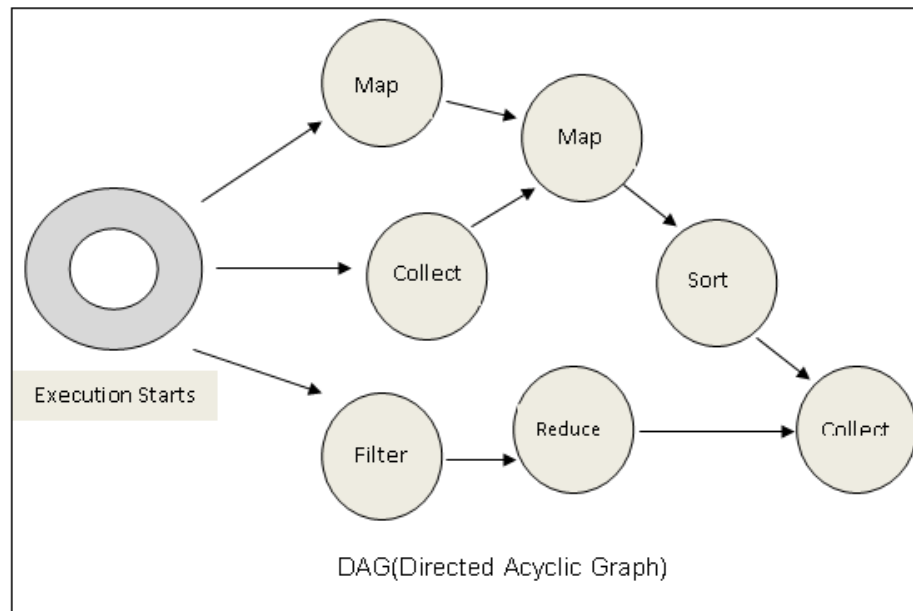
- The RDD abstraction is exposed through a language-integrated API in Java, Python, Scala, and R similar to local, in-process collections.
- This simplifies programming complexity because the way applications manipulate RDDs is similar to manipulating local collections of data.
- An RDD is a collection of items distributed across many nodes that can be manipulated in parallel.
- RDDs offer two types of operations after creation: **transformations** and **actions**. Transformations construct a new RDD. Actions, on the other hand, compute a result based on an RDD, and either return it to the driver program or save it to an external storage system.

RDDs in Action

- RDDs support a number of operations that do useful data manipulation, but they always yield a new RDD instance. Once created, they never change, thus the adjective **immutable**.
- RDDs are **resilient** because of the Spark's built-in fault recovery mechanics. Spark is capable of healing RDDs in case of node failure.

<http://freecontent.manning.com/spark-in-action-the-notion-of-resilient-distributed-dataset-rdd/>

Directed Acyclic Graph DAG



- The DAG engine helps to eliminate the MapReduce multi-stage execution model and offers significant performance improvements.

<https://databricks.com/blog/2015/06/22/understanding-your-spark-application-through-visualization.html>

<https://www.sigmoid.com/apache-spark-internals/>

DAG Example

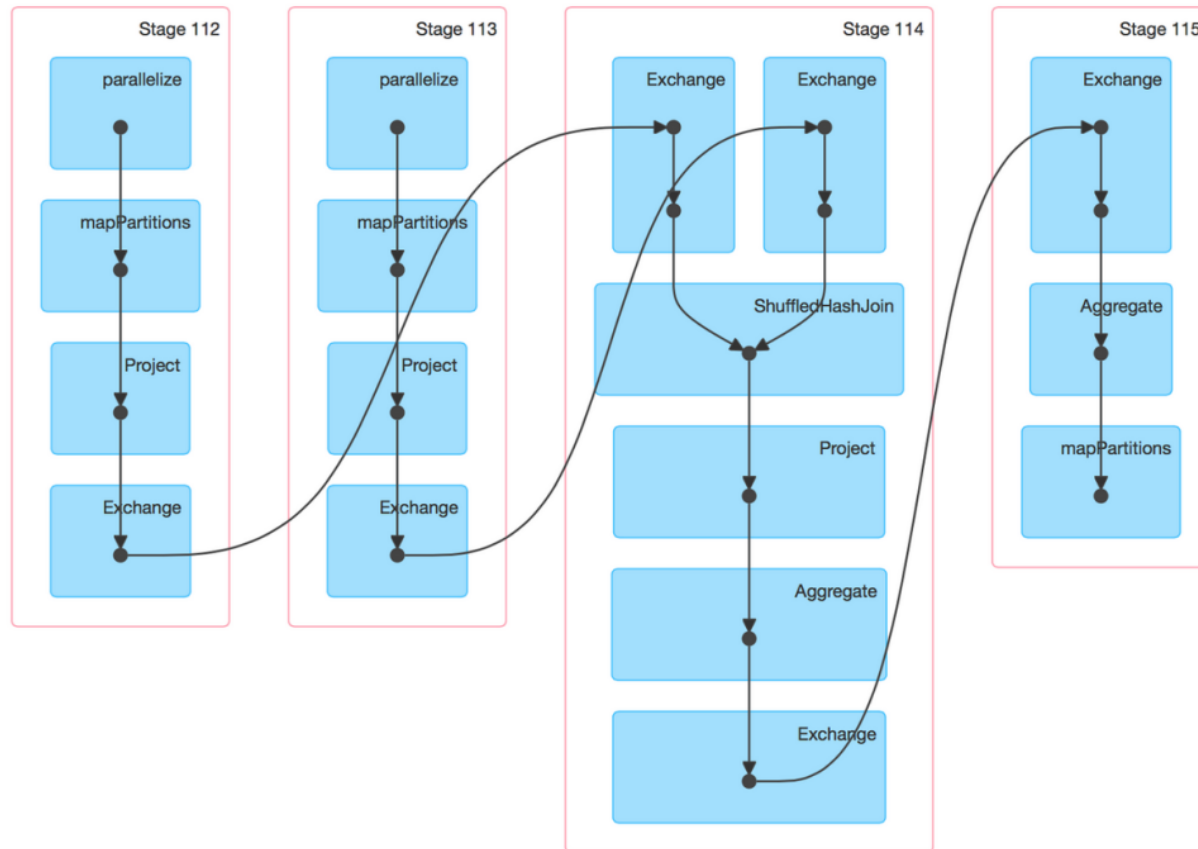
Details for Job 8

Status: SUCCEEDED

Completed Stages: 4

▶ Event Timeline

▼ DAG Visualization



<https://databricks.com/blog/2015/06/22/understanding-your-spark-application-through-visualization.html>

Spark 5 minutes example

```
wget http://en.wikipedia.org/wiki/Hortonworks
```

```
su hdfs
```

```
hadoop fs -chmod -R 777 /user/guest
```

```
exit
```

```
hadoop fs -put ~/Hortonworks /user/guest/Hortonworks
```

Pyspark

```
myLines = sc.textFile('/user/guest/Hortonworks')
```

```
myLines_filtered = myLines.filter( lambda x: len(x) > 0 )
```

```
myLines_filtered.count()
```

<http://hortonworks.com/hadoop-tutorial/hands-on-tour-of-apache-spark-in-5-minutes/>

Run Python GANIDS code on Spark

- Convert to python3

```
2to3 -w example.py
```

- WARNING: Running python applications through 'pyspark' is deprecated as of Spark 1.0.

```
spark-submit <python file>
```

Spark Useful Resources

Good presentation + eBook available

https://www.youtube.com/watch?v=mL5dQ_1gkiA

James A. Scott – “Getting Started With Apache Spark”

<https://www.mapr.com/ebooks/spark/>

“Advanced Analytics with Spark”

Includes examples from different areas

<http://shop.oreilly.com/product/0636920035091.do>

<https://github.com/sryza/aas>

Machine Learning Pipeline

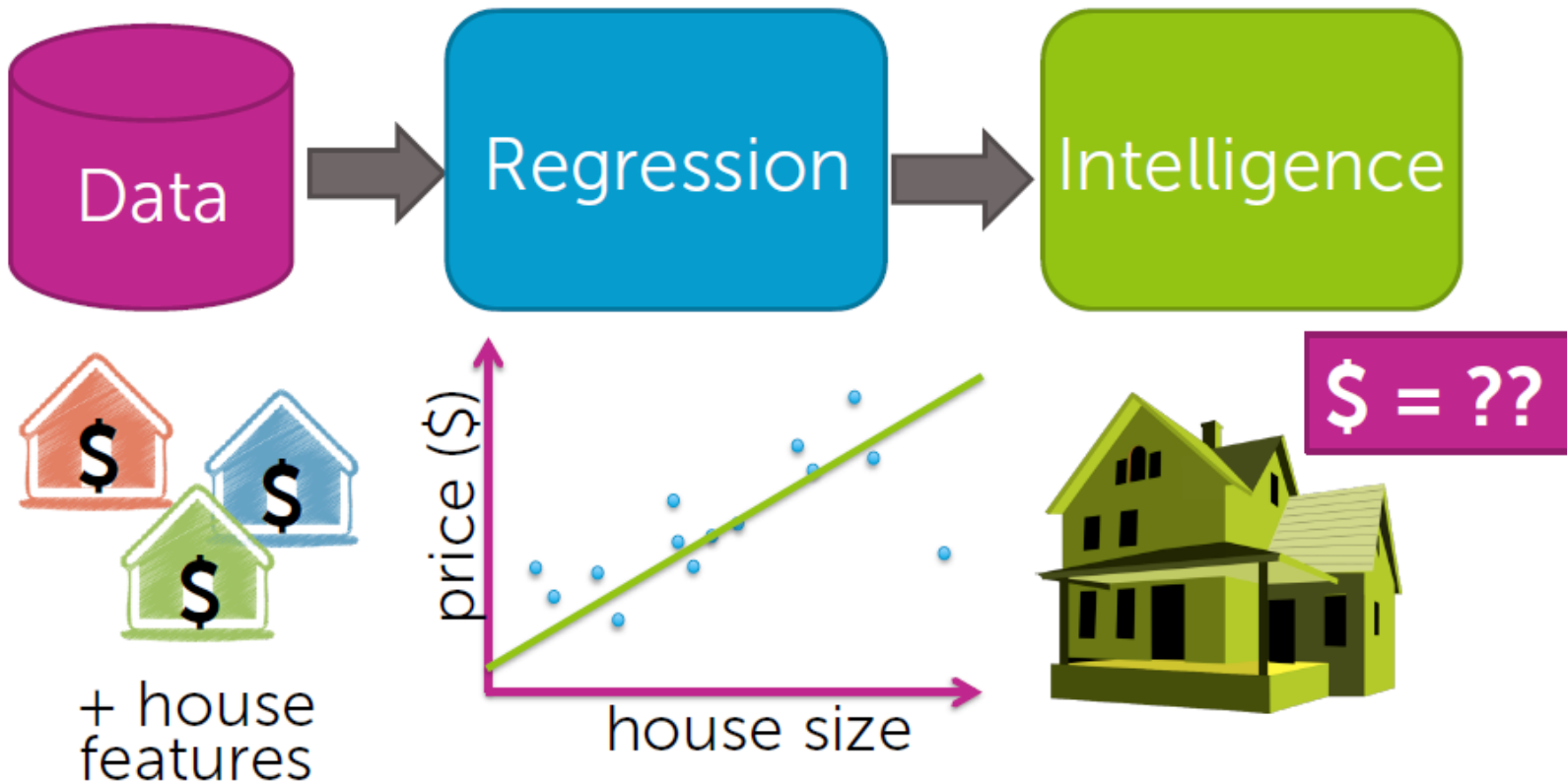


<https://www.coursera.org/learn/ml-foundations>

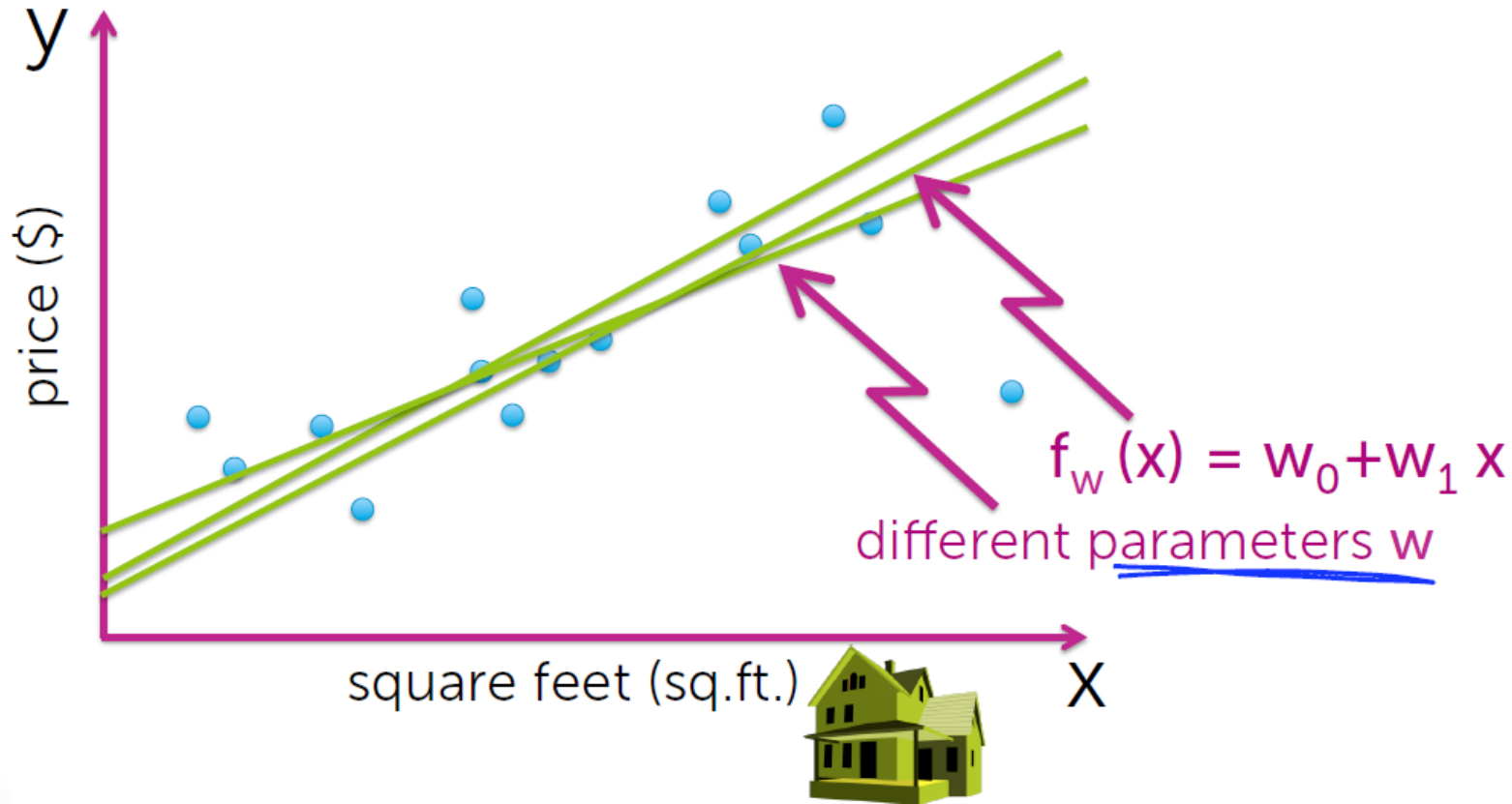
<https://www.coursera.org/learn/ml-regression>

<https://www.coursera.org/learn/ml-clustering-and-retrieval>

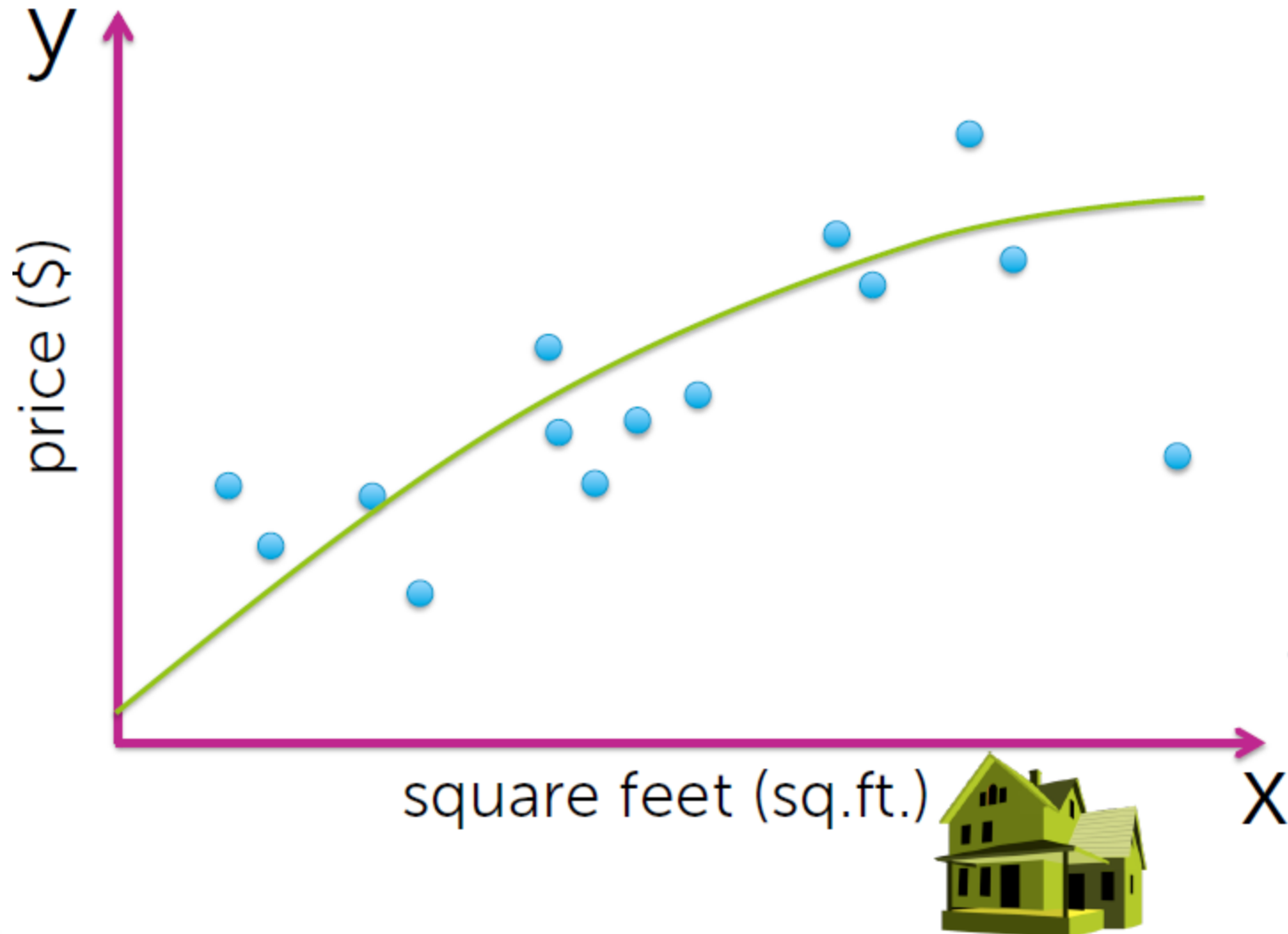
Regression



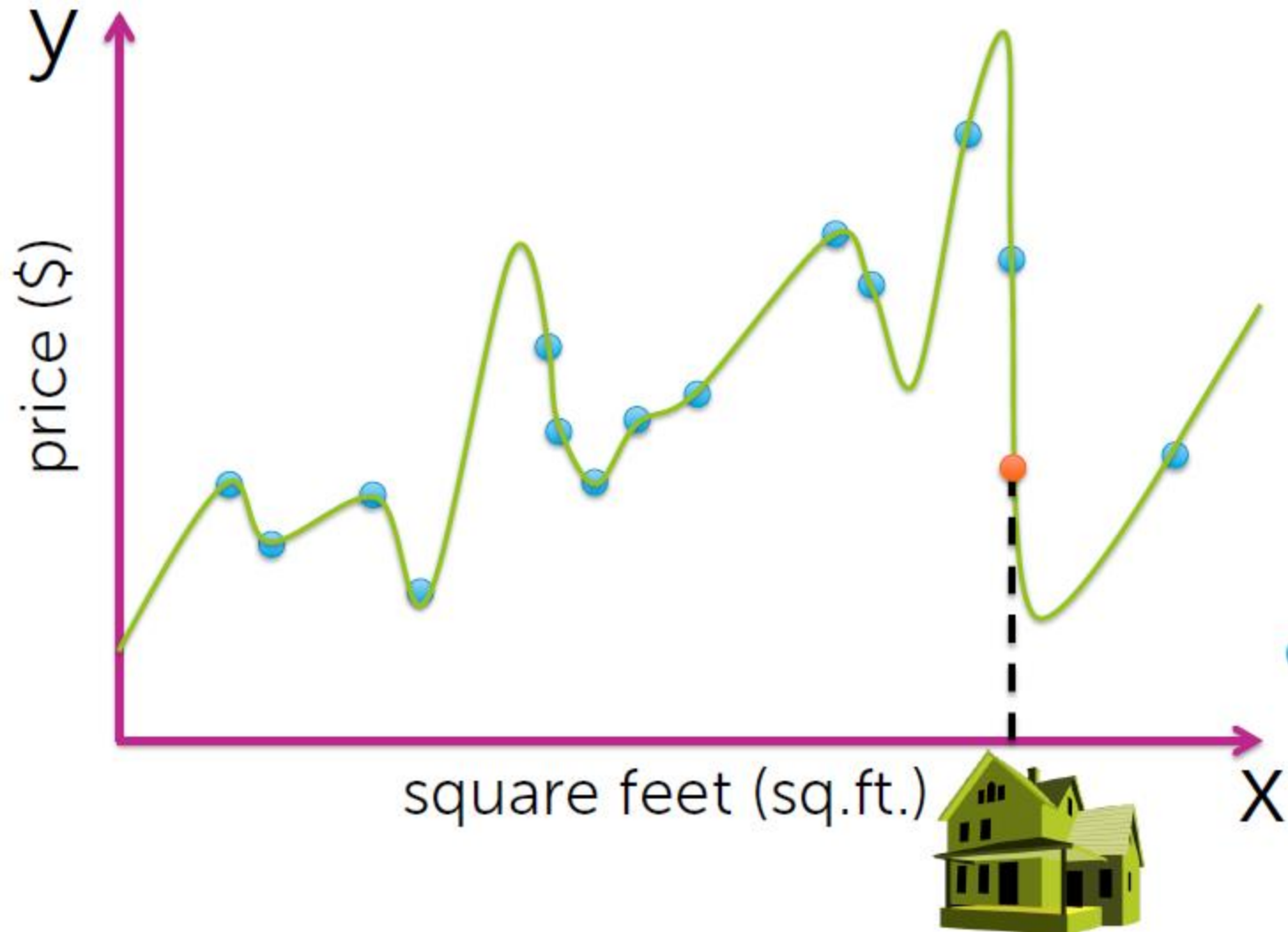
Linear Regression Model



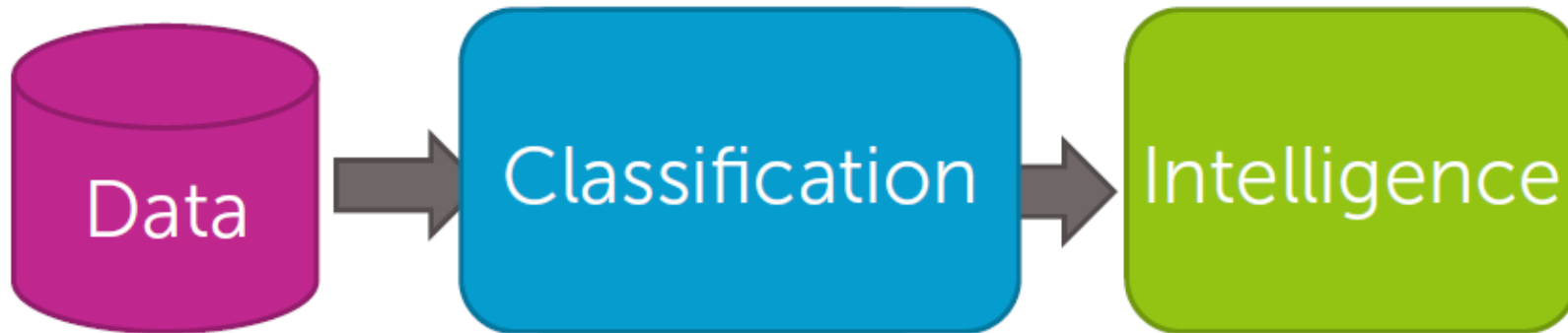
Quadratic function model



Quadratic function model



Classification



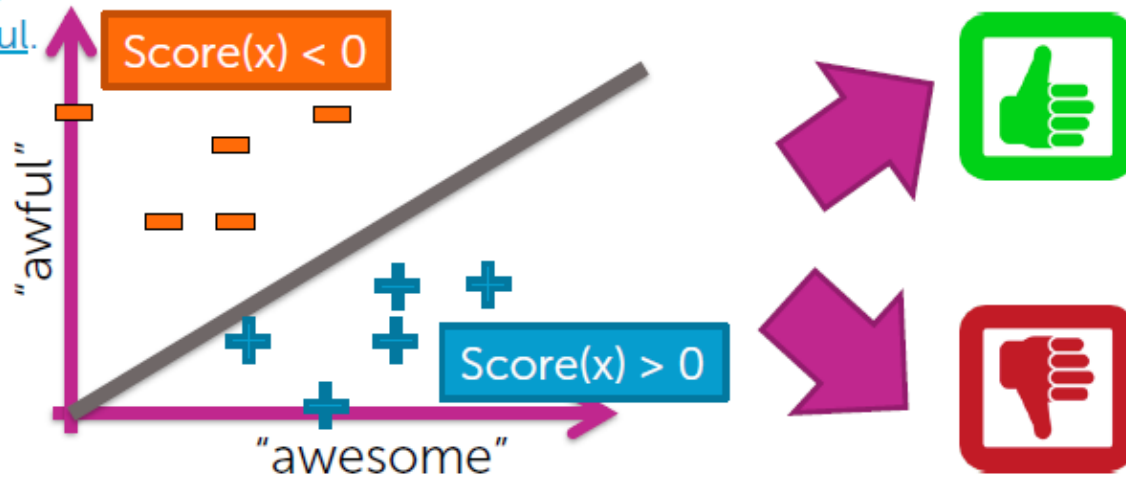
Sushi was awesome,
the food was awesome,
but the service was awful.

All reviews:

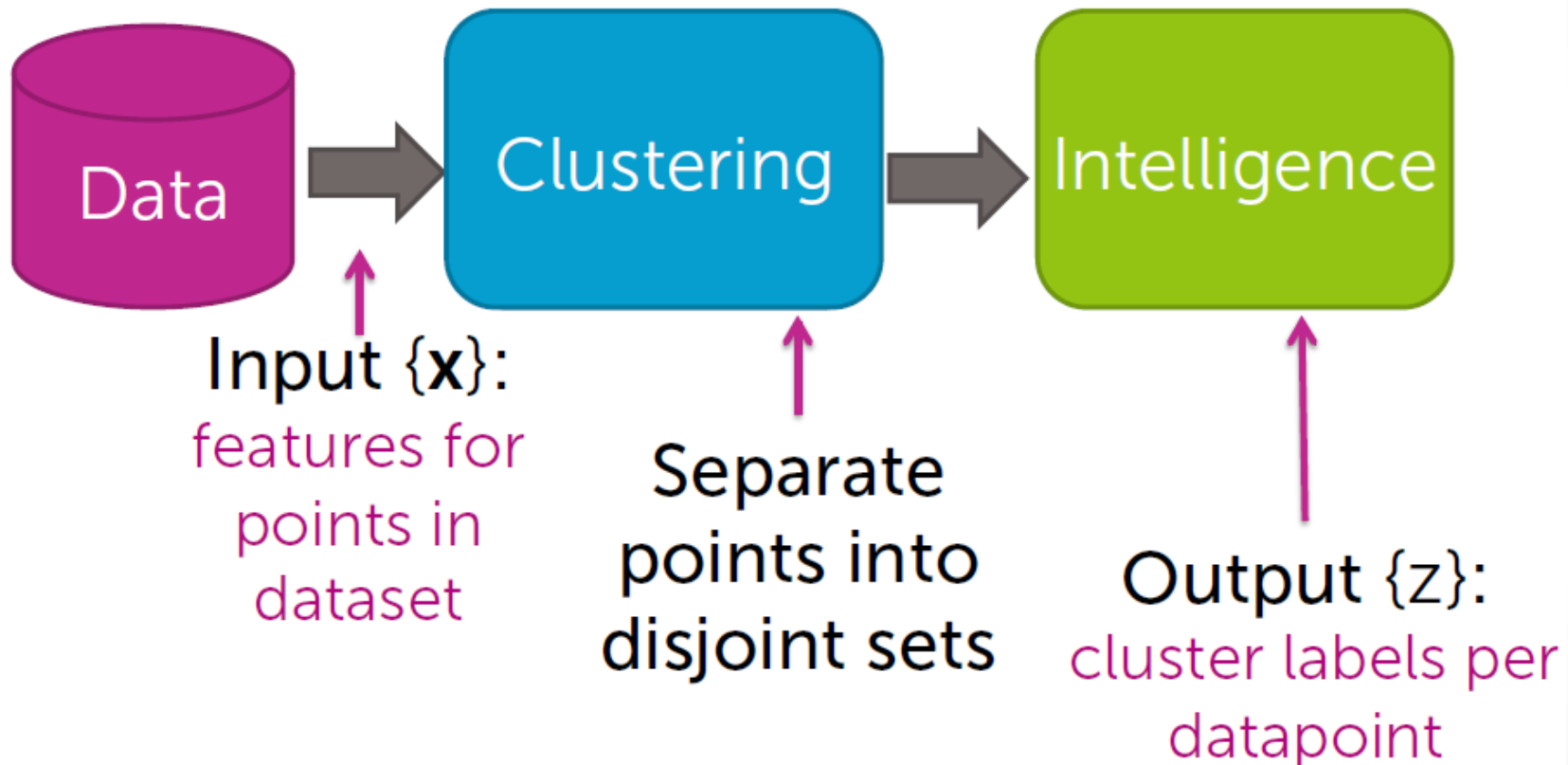
★★★★★ 7/21/2015
This is probably my favorite place to eat Japanese in Seattle. My boyfriend and I ordered nigiri of scallop, Japanese snapper (seasonal), and the agedashi tofu and 2 special rolls. I would skip the special rolls, because the nigiri and sashimi cuts is where this place excels. The tofu, as recommended by other Yelpers was amazing. It's more chewy and the sauce/gravy is the perfect amount of flavor for the delicate tofu.

★★★★★ 6/11/2015
Dining here at the sushi bar made me feel like sitting front row to an amazing performance. We didn't have rosos, banged down to the ID after work, got here breathlessly at 5:10pm, and got the last two seats in the place.

★★★★★ 8/9/2015
I came here having high expectations due to the reviews of this place, but I was bit disappointed. The restaurant is small so do make reservations when you come here. Dishes cost from \$4-25 each and dishes are small.



What is clustering



Document retrieval



SPORTS



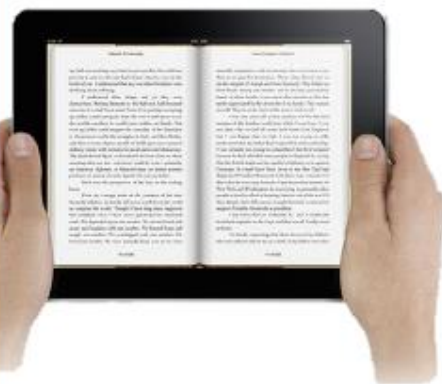
WORLD NEWS



ENTERTAINMENT



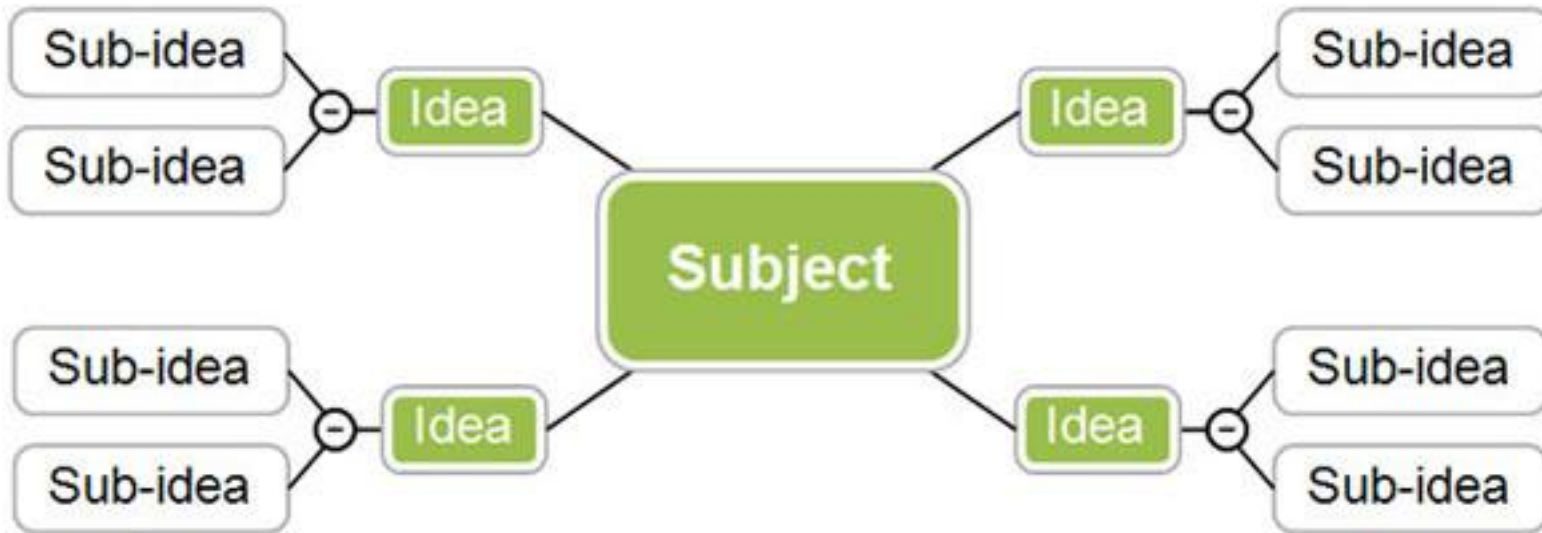
SCIENCE



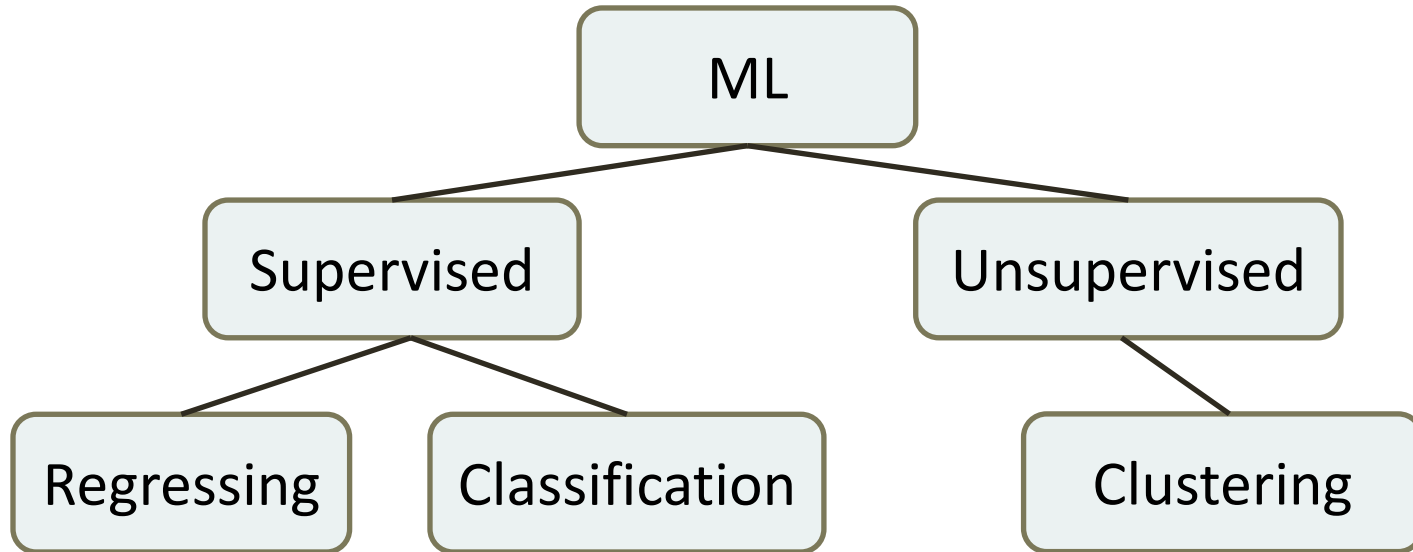
Feature selection

- Principal Component Analysis (PCA) is an essential technique in data compression and feature extraction.
- PCA reduces the amount of dimensions required to classify new data and produces a set of principal components, which are orthonormal eigenvalue/eigenvector pairs.
- It reduces the dimensionality of data by restricting attention to those directions in the feature space in which the variance is greatest

Mind Maps



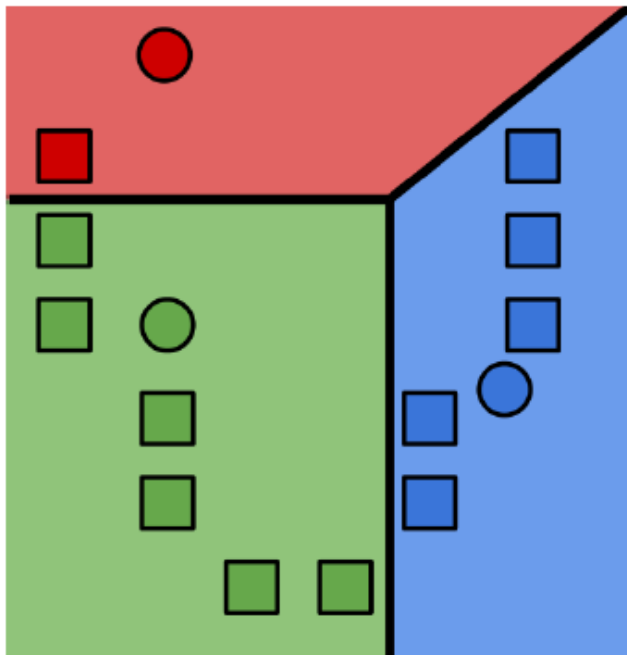
ML Methods



- In supervised learning, there is a training set used to build the model. Then the input data is tested against the model in order to get prediction.
- Supervised learning is divided in two classes: Regressing and Classification. Regression produces/predict single value outcome, while Classification - class of values.

K-Means

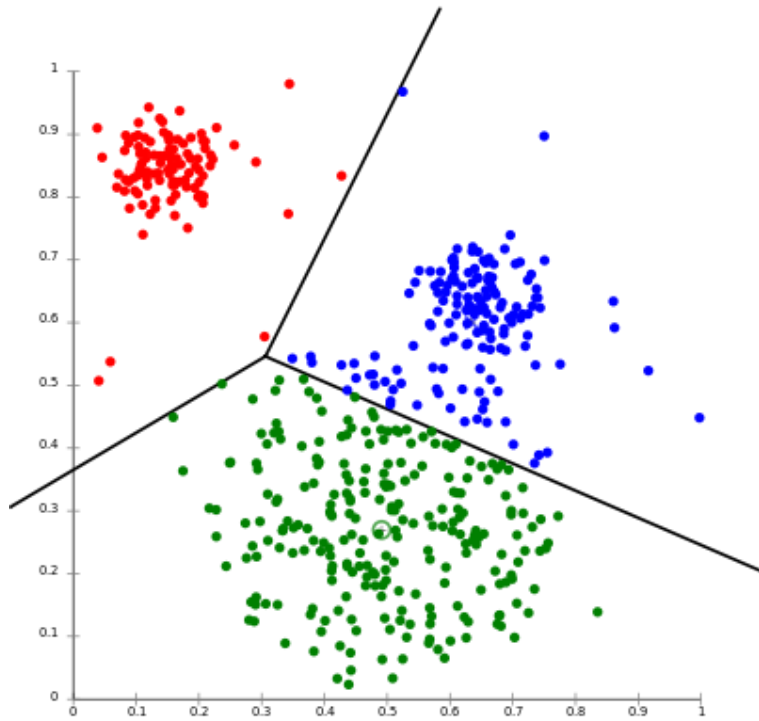
k-means aims to minimize sum of square distances to cluster centers



Makes **hard assignments** of data points to clusters

Unsupervised
learning task

K-Means



$$J(V) = \sum_{i=1}^C \sum_{j=1}^{c_i} (\|x_i - v_i\|)^2$$

Numerical Libraries

- 1. NumPy (494214 downloads in August 2015)
- 2. Pandas (224120 downloads in August 2015)
- 3. Scipy (169860 downloads in August 2015)
- 4. matplotlib (145772 downloads in August 2015):
- 5. Patsy (47625 downloads in August 2015)
- 6. Sympy (18086 downloads in August 2015)
- 7. Plotly (18044 downloads in August 2015)
- 8. statsmodels (17834 downloads in August 2015)
- 9. ADiPy (511 downloads in August 2015)
- 10. matalg27 (136 downloads in August 2015)

<http://www.palrad.com/top-python-math-statistics-libraries-w-12007/>

<http://www.datasciencecentral.com/profiles/blogs/9-python-analytics-libraries-1>

https://www.kevinsheppard.com/images/0/09/Python_introduction.pdf

MLlib

- MLlib is Apache Spark's scalable machine learning library
- MLlib fits into Spark's APIs and interoperates with NumPy in Python. You can use any Hadoop data source (e.g. HDFS, HBase, or local files), making it easy to plug into Hadoop workflows.

<http://spark.apache.org/mllib/>

MLlib contains the following algorithms and utilities:

- logistic regression and linear support vector machine (SVM)
- classification and regression tree
- random forest and gradient-boosted trees
- recommendation via alternating least squares (ALS)
- clustering via k-means, Gaussian mixtures (GMM), and power iteration clustering
- topic modeling via latent Dirichlet allocation (LDA)
- singular value decomposition (SVD) and QR decomposition
- principal component analysis (PCA)

MLlib contains the following algorithms and utilities:

- linear regression with L_1 , L_2 , and elastic-net regularization
- isotonic regression
- multinomial/binomial naive Bayes
- frequent itemset mining via FP-growth and association rules
- sequential pattern mining via PrefixSpan
- summary statistics and hypothesis testing
- feature transformations
- model evaluation and hyper-parameter tuning

prediction.io

- An **open-source** machine learning server for developers and data scientists to create predictive engines for production environments, with zero downtime training and deployment.

<https://prediction.io/>

- **PredictionIO template gallery** offers a wide range of predictive engine templates for download, developers can customize them easily.

<http://templates.prediction.io/>

- Single command install

```
$ bash -c "$(curl -s https://install.prediction.io/install.sh)"
```

<https://docs.prediction.io/install/>

prediction.io Examples

- PredictionIO's Classification Engine Template has integrated Apache Spark MLlib's Naive Bayes algorithm by default.

To predict the label of a user with attr0=2, attr1=0 and attr2=0, you send this JSON { "attr0":2, "attr1":0, "attr2":0 } to the deployed engine and it will return a JSON of the predicted plan.

input

```
{ "attr0":2, "attr1":0, "attr2":0 }
```

output

```
{"label":0.0}
```

<https://docs.prediction.io/templates/classification/quickstart/>

Anomaly Detection, K-Means Method

- <https://www.youtube.com/watch?v=TC5cKYBZAel>
- <http://www.slideshare.net/cloudera/anomaly-detection-with-apache-spark-2>
- <https://github.com/keiraqz/anomaly-detection>
- This model is using KMeans(Spark MLlib K-means) approach and it is trained on "normal" dataset only. After the model is trained, the centroid of the "normal" dataset will be returned as well as a threshold.
- During the validation stage, any data points that are further than the threshold from the centroid are considered as "anomalies".

Dataset

The dataset is downloaded from KDD Cup 1999 Data for Anomaly Detection.

<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>

- Training Set: The training set is separated from the whole dataset with the data points that are labeled as "normal" only.
- Validation Set: The validation set is using the whole dataset. All data points that are NOT labeled as "normal" are considered as "anomalies".

Dataset

0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,
0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.
00,normal.

0,tcp,http,SF,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0
.00,0.00,1.00,0.00,0.00,19,19,1.00,0.00,0.05,0.00,0.00,0.00,0.00,0
.00,normal.

0,tcp,http,SF,235,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,
0.00,0.00,1.00,0.00,0.00,29,29,1.00,0.00,0.03,0.00,0.00,0.00,0.00,
0.00,normal.

0,tcp,http,SF,219,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,6,6,0.00,0.00,
0.00,0.00,1.00,0.00,0.00,39,39,1.00,0.00,0.03,0.00,0.00,0.00,0.00,
0.00,normal.

Run scala code file

- `$ spark-shell -i AnomalyDetection-shell.scala`
- there is two data inputs inside scala file

.....

```
val rawData = sc.textFile("dataset/ad.train.csv", 120)
```

.....

```
val rawTestdata = sc.textFile("dataset/ad.all.csv", 120)
```

.....

Anomalies count and print

.....

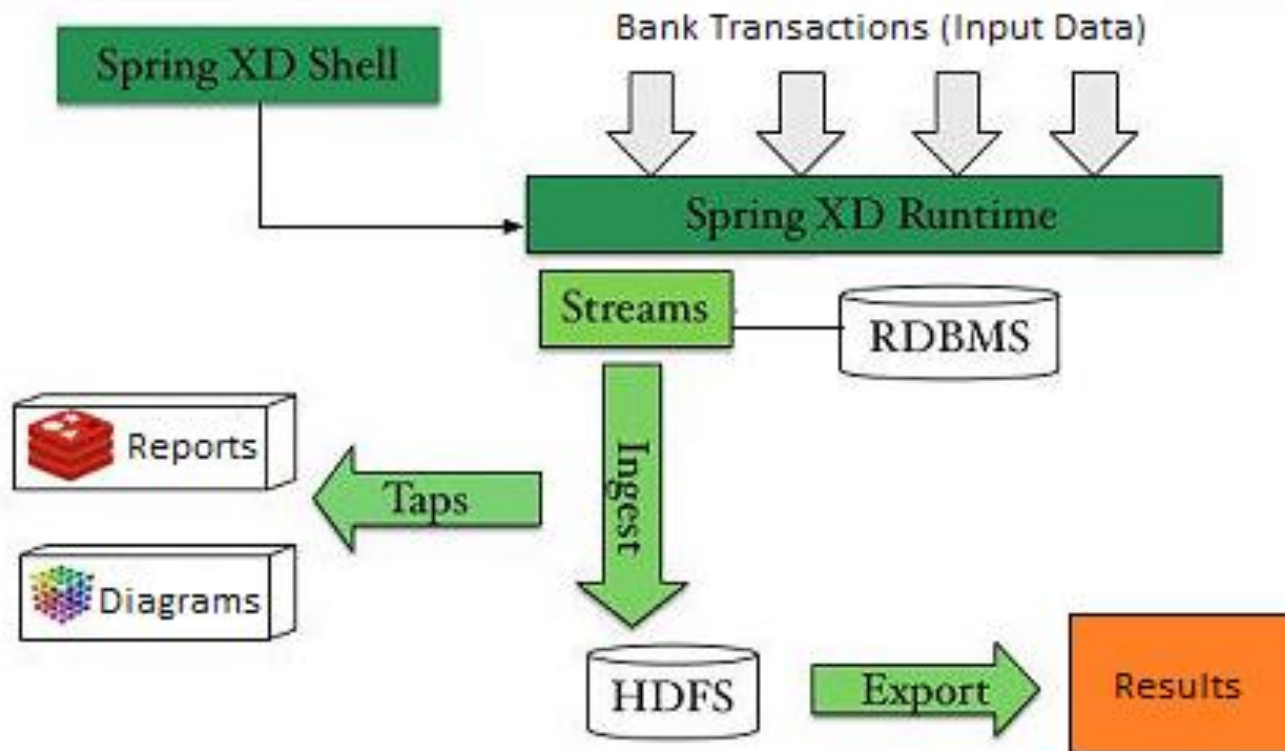
```
// get the anomalies
```

```
val anomalies = normalizedTestDataAndLabel.filter(  
    d => distToCentroid(d._1, model) > threshold  
)
```

```
anomalies.count()
```

```
anomalies.filter(x => x._2 != "normal.").count // count true  
"anomalies"
```


Spring XD



- Spring XD is a unified, distributed, and extensible system for data ingestion, real time analytics, batch processing, and data export.

<https://github.com/spring-projects/spring-xd/wiki/About-Spring-XD>

Spark Streaming



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

Spark Streaming

- Need to detect the anomalies closer to their appearance
- DStream is represented as a sequence of RDDs



Boiling Frog



Conclusion

- Rule based methods
- GA algorithm for generating rules
- ML methods to create model of normality
- Combine machine learning and rule-based techniques
- Big data technologies for implementation
- Using real network data

Thank you for your
attention!

