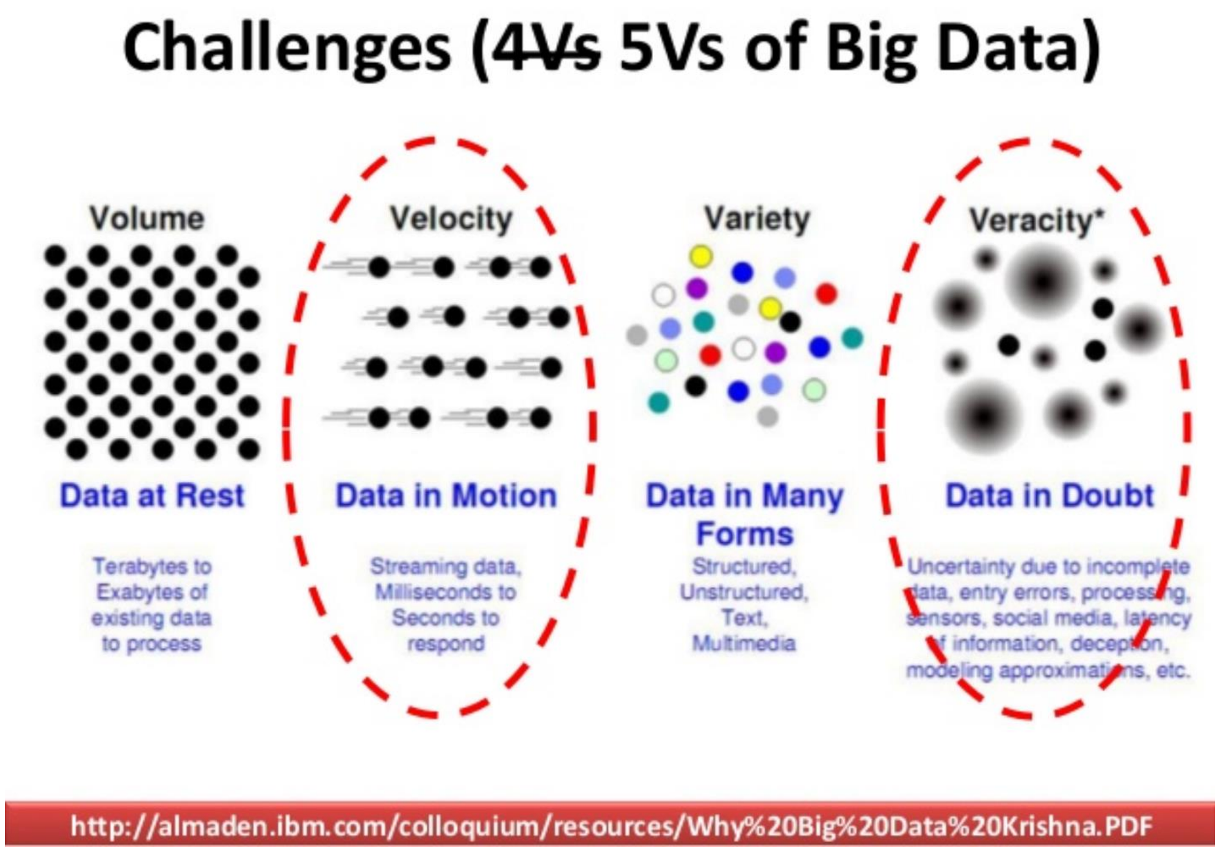
Data Engineering (Use Cases)

Erdi Ölmezoğulları

1-Why/When do we need big data? Explain with example?



Challenges (~~4Vs~~ 5Vs of Big Data)

Volume

Data at Rest

Terabytes to Exabytes of existing data to process

Velocity

Data in Motion

Streaming data, Milliseconds to Seconds to respond

Variety

Data in Many Forms

Structured, Unstructured, Text, Multimedia

Veracity*

Data in Doubt

Uncertainty due to incomplete data, entry errors, processing, sensors, social media, latency of information, deception, modeling approximations, etc.

http://almaden.ibm.com/colloquium/resources/Why%20Big%20Data%20Krishna.PDF

From my Master's thesis presentation, 2013

Big Data is a kind of analytical terms that covers **management** and **processing** of the high volume and speed data by using well-defined distributed systems (Hadoop, Spark, NoSQLs .etc) . It is addressing mainly top four challenging issues, such as Volume, Velocity, Variety, Veracity (Value, Variability)

- **(Big) Data Management:** Design scalable data storages such as Data Lake by using HDFS, NoSQLs.
- **(Big) Data Analytics:** Performing analytical queries to find out a valuable insights to make some improvement on products. MapReduce, Spark, Spark ML, MLlib, Hive, HBase .etc

2a-Hadoop / Hadoop ecosystem

- **Hadoop**: It is distributed big data framework that works on top of distributed commodity storage system to perform high scalable analytics demands with different tools (Hive, MapReduce, HBase, HDFS) togethers.
- **YARN**: It is new 2nd Hadoop version of Hadoop. The old one's JobManager and TaskManager nodes were replaced with new nodes ResourceManager and NodeManager respectively to manage resources (CPU, Memory, Network, etc.) in with respect to a prefered schedule.
- **Datanode**: It consists of different size of redundant blocks (chunks) distributed across HDFS Cluster.
- **Namenode**: It tracks the location of data across HDFS cluster.
- **JobManager**: MapReduce jobs are tracked by JobManager.
- **TaskManager**: It handles tasks (map, reduce) that have already handed over by JobManager.
- **ResourceManager**: It is working as a master in Hadoop cluster that is managing of the resources (CPU, Memory, Network, etc.) to utilize reasonably and effectively on the cluster instead of JobManager.
- **NodeManager**: It is working as a slaver in Hadoop cluster that is managing containers (slots) for incoming task instead of TaskManager in Hadoop 2.

2a-Hadoop / YARN

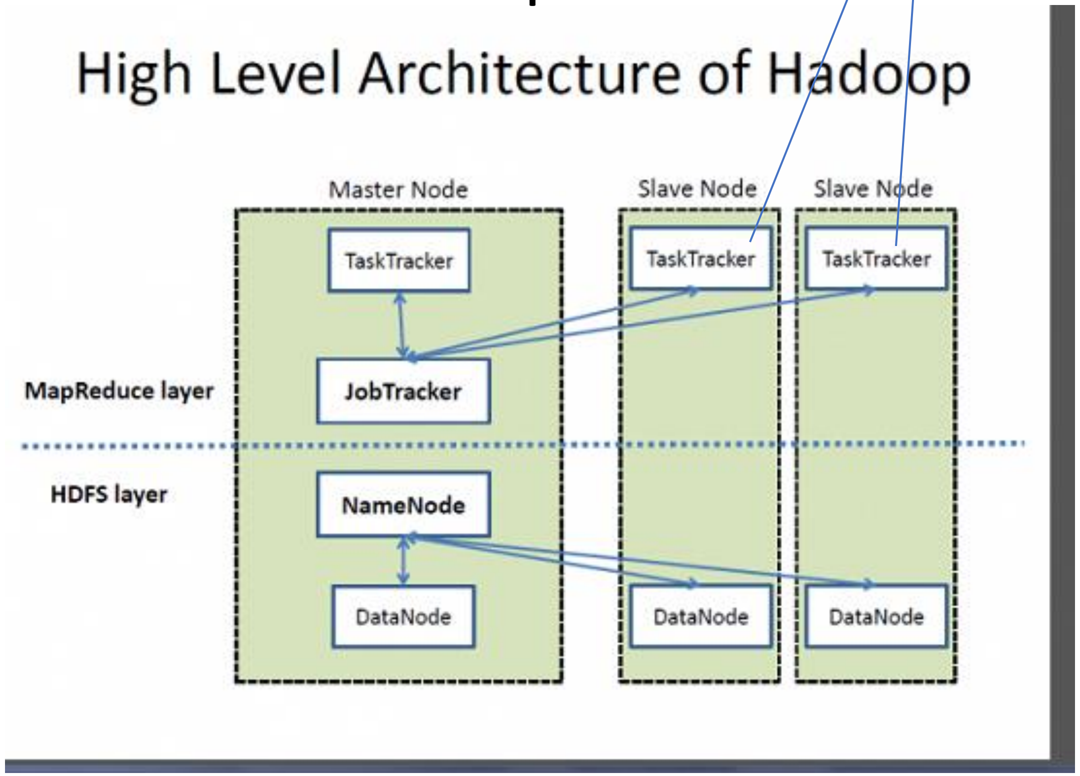
Hadoop1.X vs Hadoop 2.X

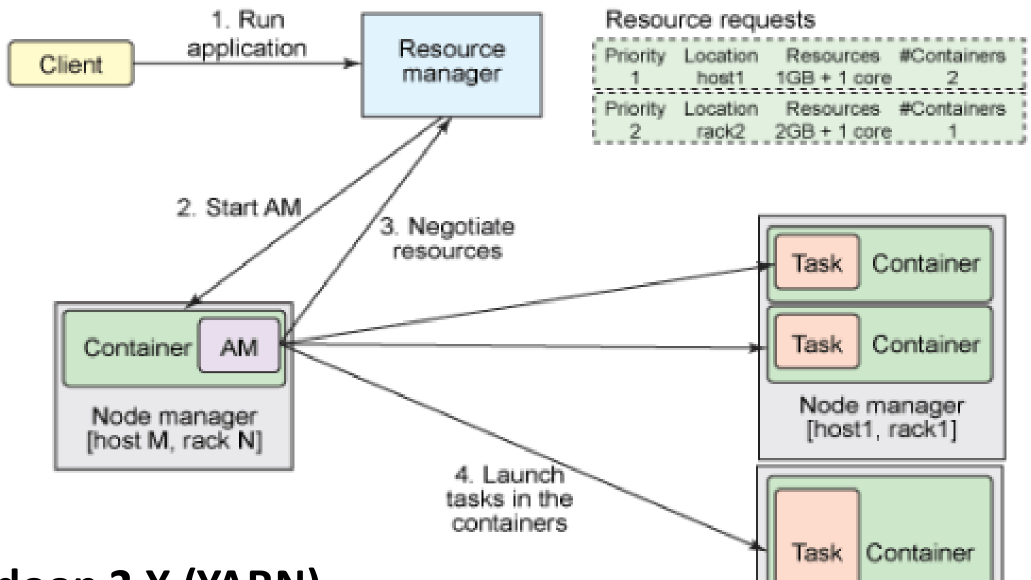
Job Tracker --> Resource Manager

Task Tracker --> Node Manager

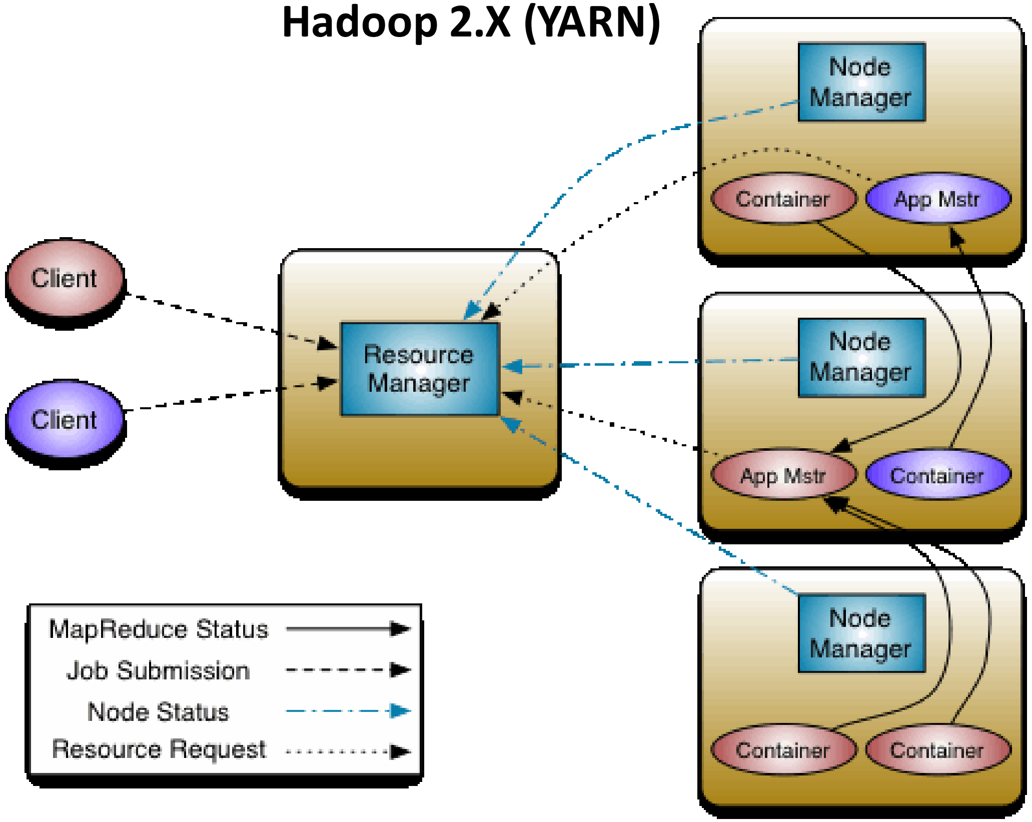
Map/Reduce (Task)

Hadoop 1.X



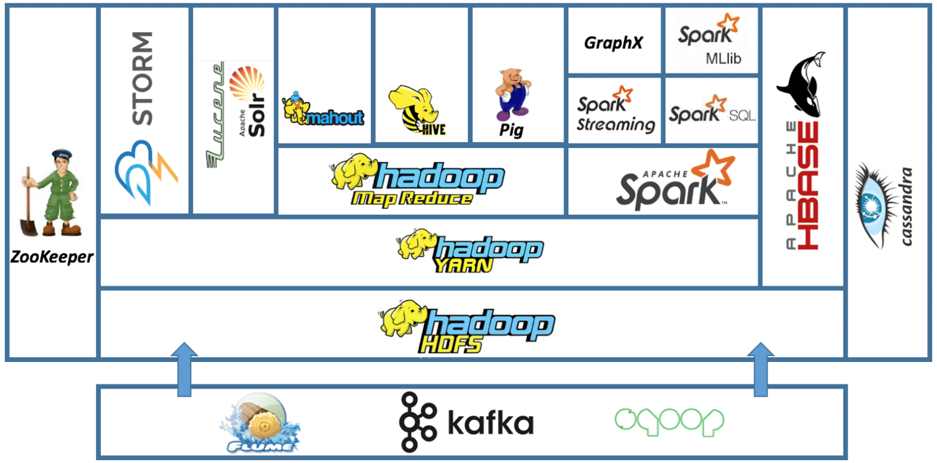

Client
1. Run application
Resource manager
Resource requests
Priority Location Resources #Containers
1 host1 1GB + 1 core 2
Priority Location Resources #Containers
2 rack2 2GB + 1 core 1
2. Start AM
3. Negotiate resources
Container
AM
Node manager [host M, rack N]
Task
Container
Task
Container
Node manager [host1, rack1]
4. Launch tasks in the containers
Task
Container
Node manager [hostA, rack2]

Hadoop 2.X (YARN)

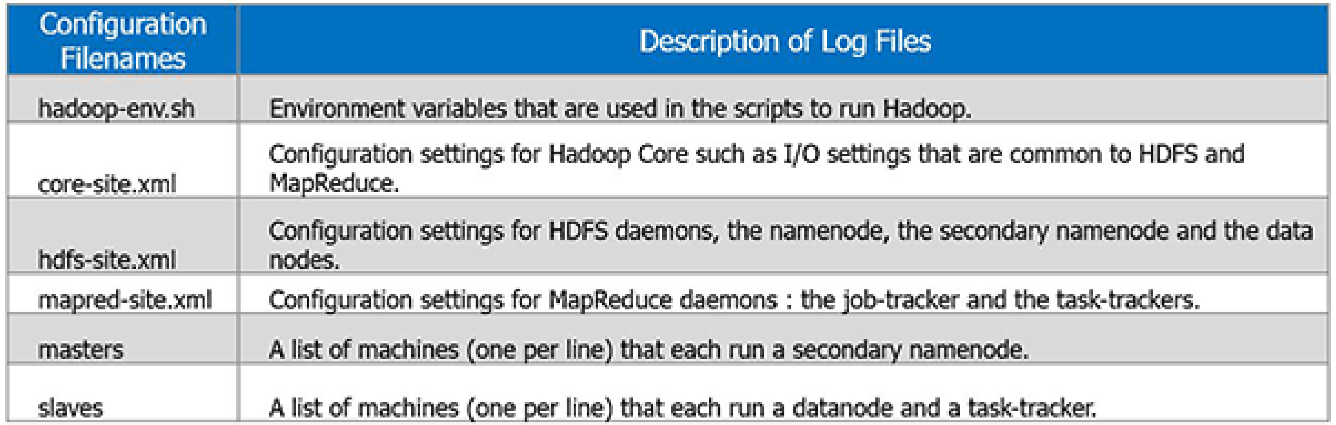


2b-Hadoop / Hadoop ecosystem

- **HDFS** : It is a distributed file system that stores on datanode (data) and namenode(metadata) on Hadoop.
- **Map Reduce** : It is a functional programming paradigm that is being applying in data processing stage to make task parallelize over Hadoop cluster. Map is creating input data. Reduce is gathering the final results over (internal) accumulators.
- **Spark** : It is a distributed in-memory big data and analytics tool that was built upon RDD (Resilient Distributed Datasets) by applying some functional programming approach (lazy evaluation, map reduce). It supports Spark SQL, Spark API, DataSet (DataFrame).
- **Hive** : It is data warehouse that provides a simple abstract like SQL to interact with Hadoop (HDFS) regardless writing any kind of MapReduce job since SQLs are being converted into MapReduce job, underneath.
- **HBase**: It is a columnar based distributed NoSQL database that works on top of HDFS.

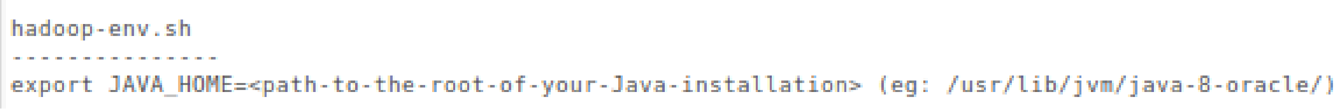


2b.a-Hadoop / HDFS, Configuration

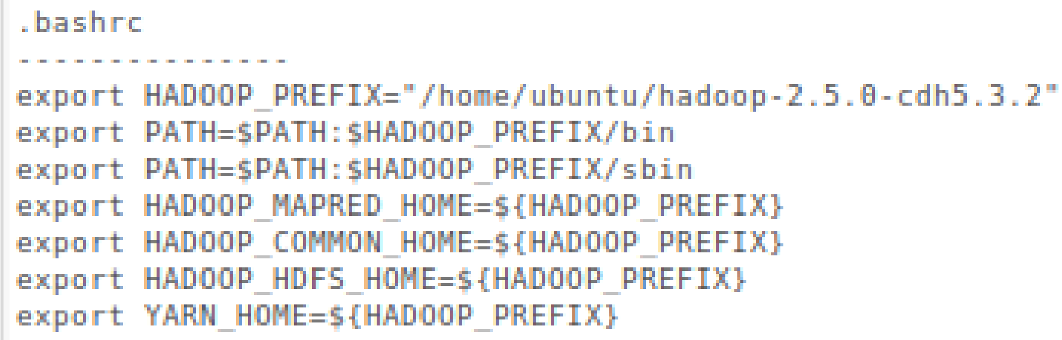


Configuration Filenames	Description of Log Files
hadoop-env.sh	Environment variables that are used in the scripts to run Hadoop.
core-site.xml	Configuration settings for Hadoop Core such as I/O settings that are common to HDFS and MapReduce.
hdfs-site.xml	Configuration settings for HDFS daemons, the namenode, the secondary namenode and the data nodes.
mapred-site.xml	Configuration settings for MapReduce daemons : the job-tracker and the task-trackers.
masters	A list of machines (one per line) that each run a secondary namenode.
slaves	A list of machines (one per line) that each run a datanode and a task-tracker.

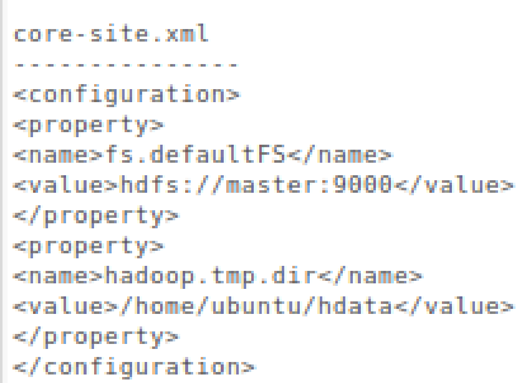
All these files are available under 'conf' directory of Hadoop installation directory.



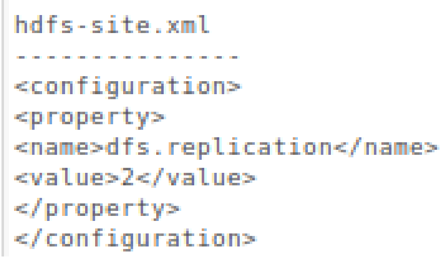
```
hadoop-env.sh
---------------
export JAVA_HOME=<path-to-the-root-of-your-Java-installation> (eg: /usr/lib/jvm/java-8-oracle/)
```



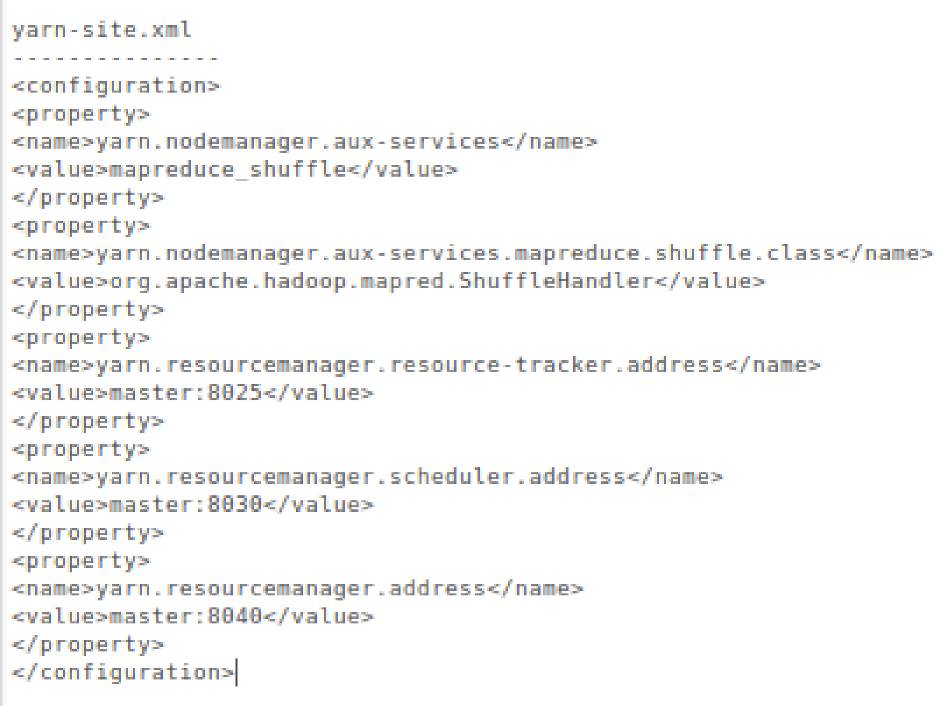
```
.bashrc
---------------
export HADOOP_PREFIX="/home/ubuntu/hadoop-2.5.0-cdh5.3.2"
export PATH=$PATH:$HADOOP_PREFIX/bin
export PATH=$PATH:$HADOOP_PREFIX/sbin
export HADOOP_MAPRED_HOME=${HADOOP_PREFIX}
export HADOOP_COMMON_HOME=${HADOOP_PREFIX}
export HADOOP_HDFS_HOME=${HADOOP_PREFIX}
export YARN_HOME=${HADOOP_PREFIX}
```



```
core-site.xml
---------------
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://master:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/ubuntu/hdata</value>
</property>
</configuration>
```



```
hdfs-site.xml
---------------
<configuration>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
</configuration>
```

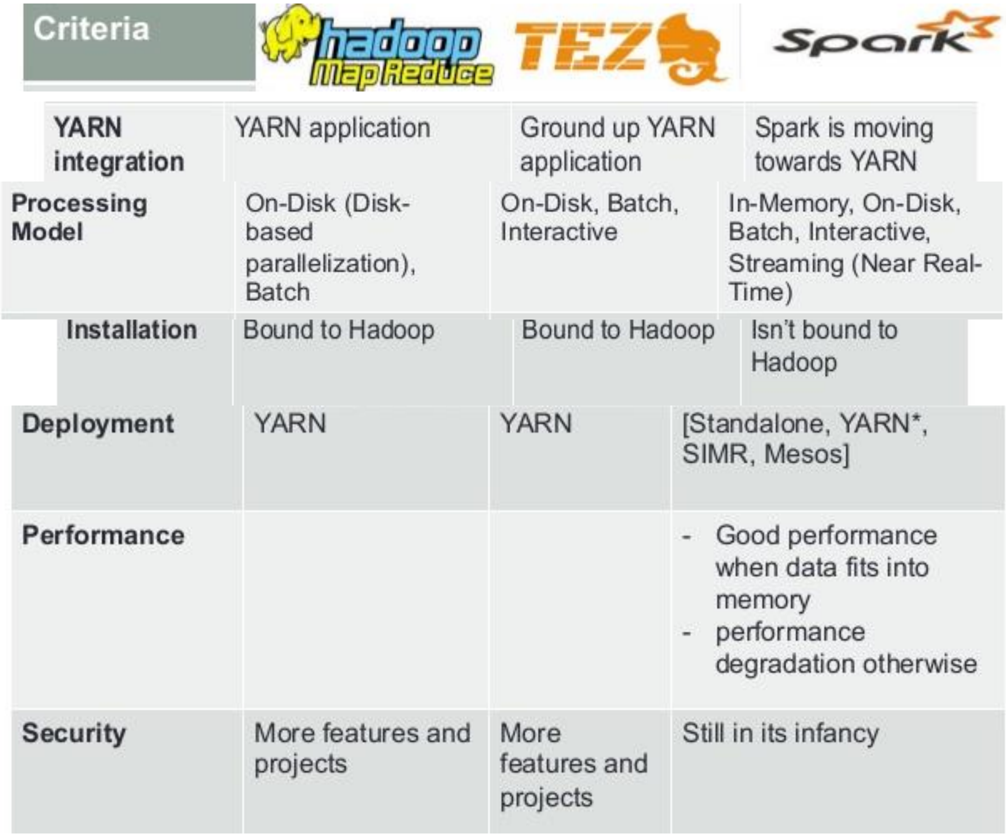


```
yarn-site.xml
--------------
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>master:8025</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>master:8030</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>master:8040</value>
</property>
</configuration>
```

2b.a-Hadoop / HDFS Limits

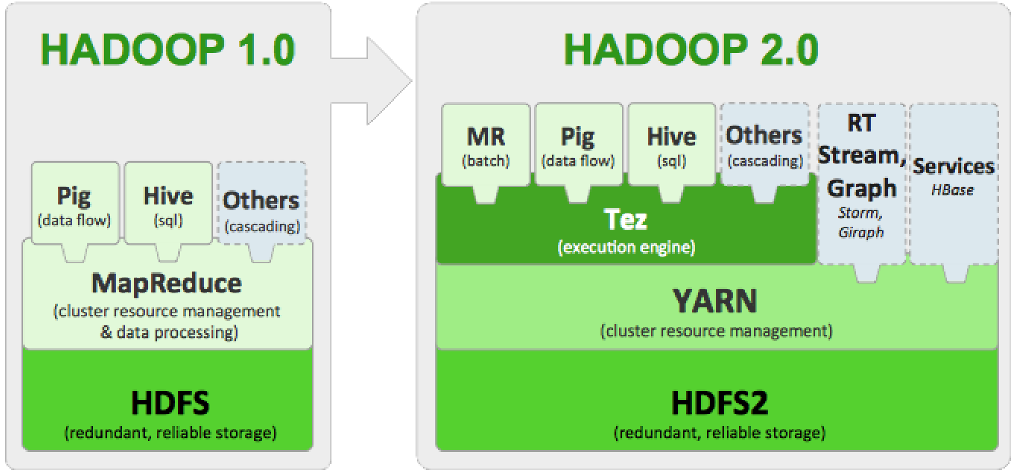
1. Issues with Small Files
2. Slow Processing Speed: Map and Reduce. So, MapReduce requires a lot of time to perform these tasks, thus increasing latency. Hence, reduces processing speed.
3. Support for Batch Processing only
4. No Real-time Processing
5. No Iterative Processing
6. Latency (I/O)
7. Single point of failure before Hadoop 2.x

2.b.b - Hadoop / MapReduce vs Spark vs Tez



Criteria	hadoop MapReduce	TEZ	Spark
YARN integration	YARN application	Ground up YARN application	Spark is moving towards YARN
Processing Model	On-Disk (Disk-based parallelization), Batch	On-Disk, Batch, Interactive	In-Memory, On-Disk, Batch, Interactive, Streaming (Near Real-Time)
Installation	Bound to Hadoop	Bound to Hadoop	Isn't bound to Hadoop
Deployment	YARN	YARN	[Standalone, YARN*, SIMR, Mesos]
Performance			- Good performance when data fits into memory - performance degradation otherwise
Security	More features and projects	More features and projects	Still in its infancy

Data	HDFS	HDFS	HDFS/RDD


HADOOP 1.0
Pig (data flow)
Hive (sql)
Others (cascading)
MapReduce (cluster resource management & data processing)
HDFS (redundant, reliable storage)
HADOOP 2.0
MR (batch)
Pig (data flow)
Hive (sql)
Others (cascading)
RT Stream, Graph Storm, Giraph
Services HBase
Tez (execution engine)
YARN (cluster resource management)
HDFS2 (redundant, reliable storage)

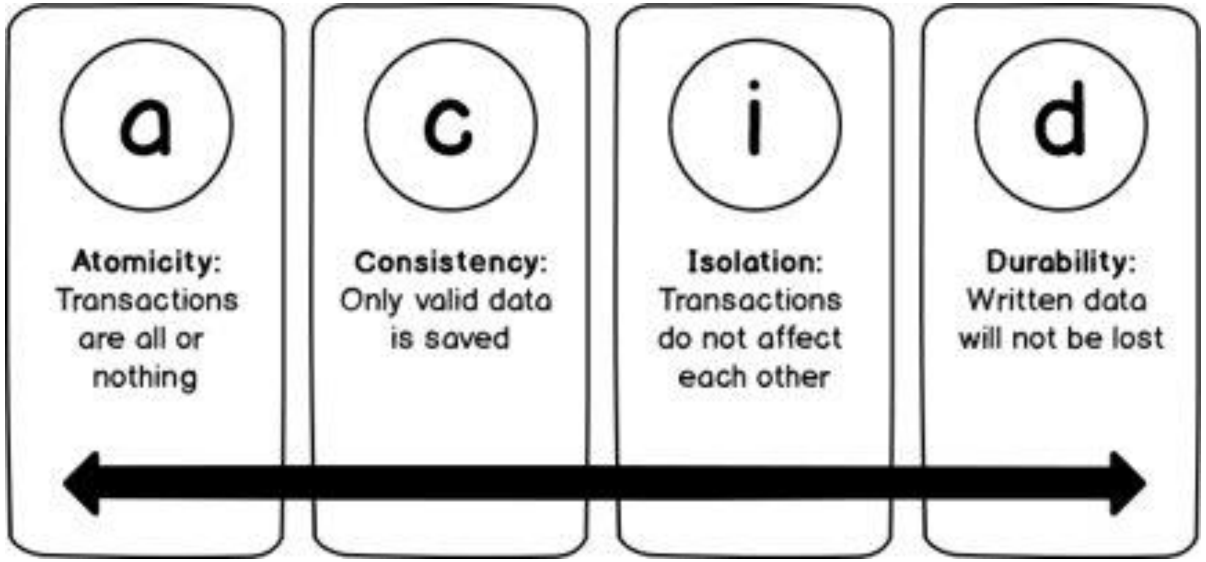
Monolithic

~ Microkernel

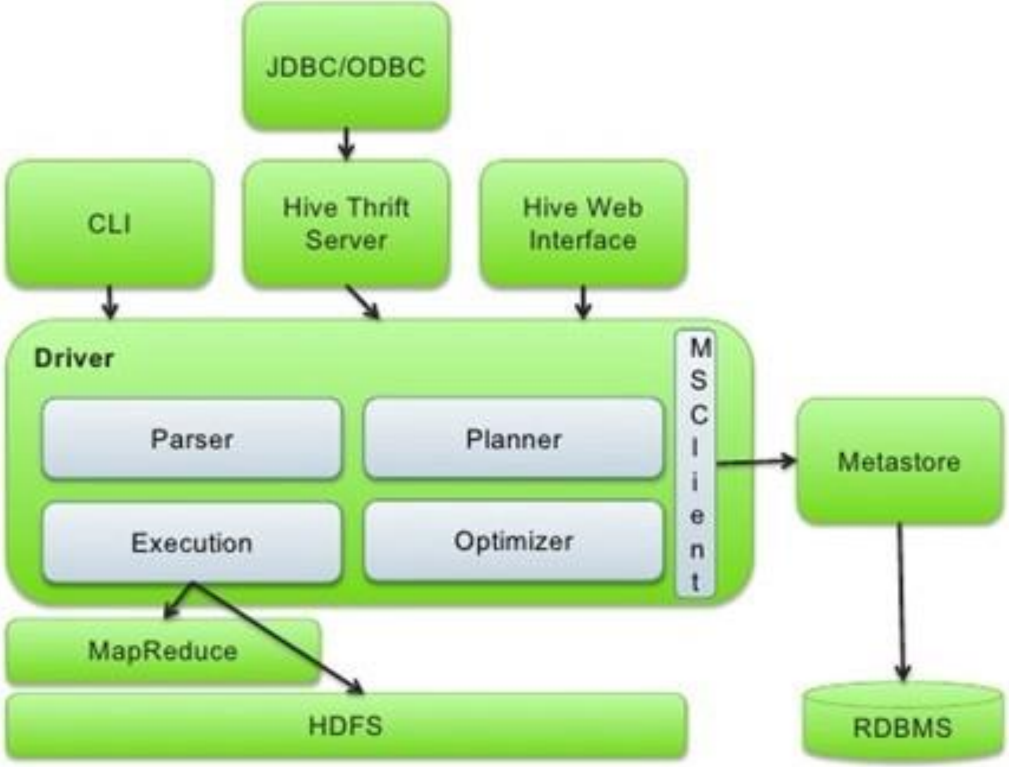
2.b.b - Hadoop / MapReduce vs Spark vs Tez

- We need an execution engine to outperform the executions of requests which is coming from the rest of Hadoop's components (MR, Hive, Pig) well, concurrently like working multiple engines (Hadoop 2.X) on single dataset rather than single engine (Hadoop 1.X)
- So, Tez is important role to execute a native YARN application over YARN that bridges interactive and batch workloads by creating a DAG of the application like Spark, underneath. However, Spark is in-memory processing engine that processes data as streaming additionally.
- All of them can be operated by YARN.

2.b.c - Hadoop / Hive vs RDBMS



a
c
i
d
Atomicity: Transactions are all or nothing
Consistency: Only valid data is saved
Isolation: Transactions do not affect each other
Durability: Written data will not be lost



2.b.c - Hadoop / Hive vs RDBMS

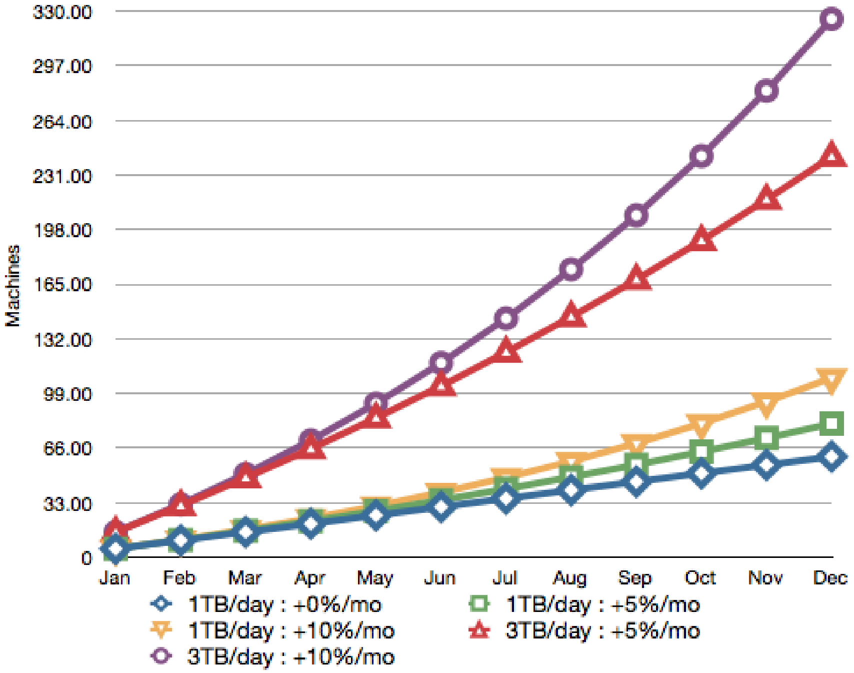
OLTP. (Online
Transaction Processing), s
pport OLAP(Online
Analytical Processing)

Hive	RDBMS
Fulfilling partially ACID until 0.13. Later ACID	Full ACID
Schema on read. i.e. schema doesn't not verify loading data	Schema on write. i.e. schema verify loading data, else rejected.
Write once, Read many times.	Read and Write many times.
Hive data size is Petabytes	Maximum data size is Terabytes
No support OLTP. However, support OLAP.	It supports only OLTP.
Static data analysis (non-real time data) example text file.	Dynamic data analysis (real time data) example data from the sensors and web feeds.
No supporting record updates in Hive After 0.14 delete, create, update	CRUD and transactions are supported.
Hive is very easily scalable at low cost	RDBMS is not scalable to low cost.
Supporting SQL but it is not a database.	It is a database.
No support indexes because data is always scanned.	Supports indexes.
Focus on only analytics.	Focus on analytics or online(device connected to network).
Distributed processing done via map/reduce	Distributed processing varies by company or person.
Higher Scales up (hundreds of nodes)	Scales to beyond 20 nodes.

3-Hadoop / Mid-size Hadoop Clusters

- How to plan my storage?
- How to plan my CPU?
- How to plan my memory?
- How to plan the network bandwidth?

1. Daily data input
2. HDFS replication factor
3. Monthly growth
4. Size of a hard drive disk



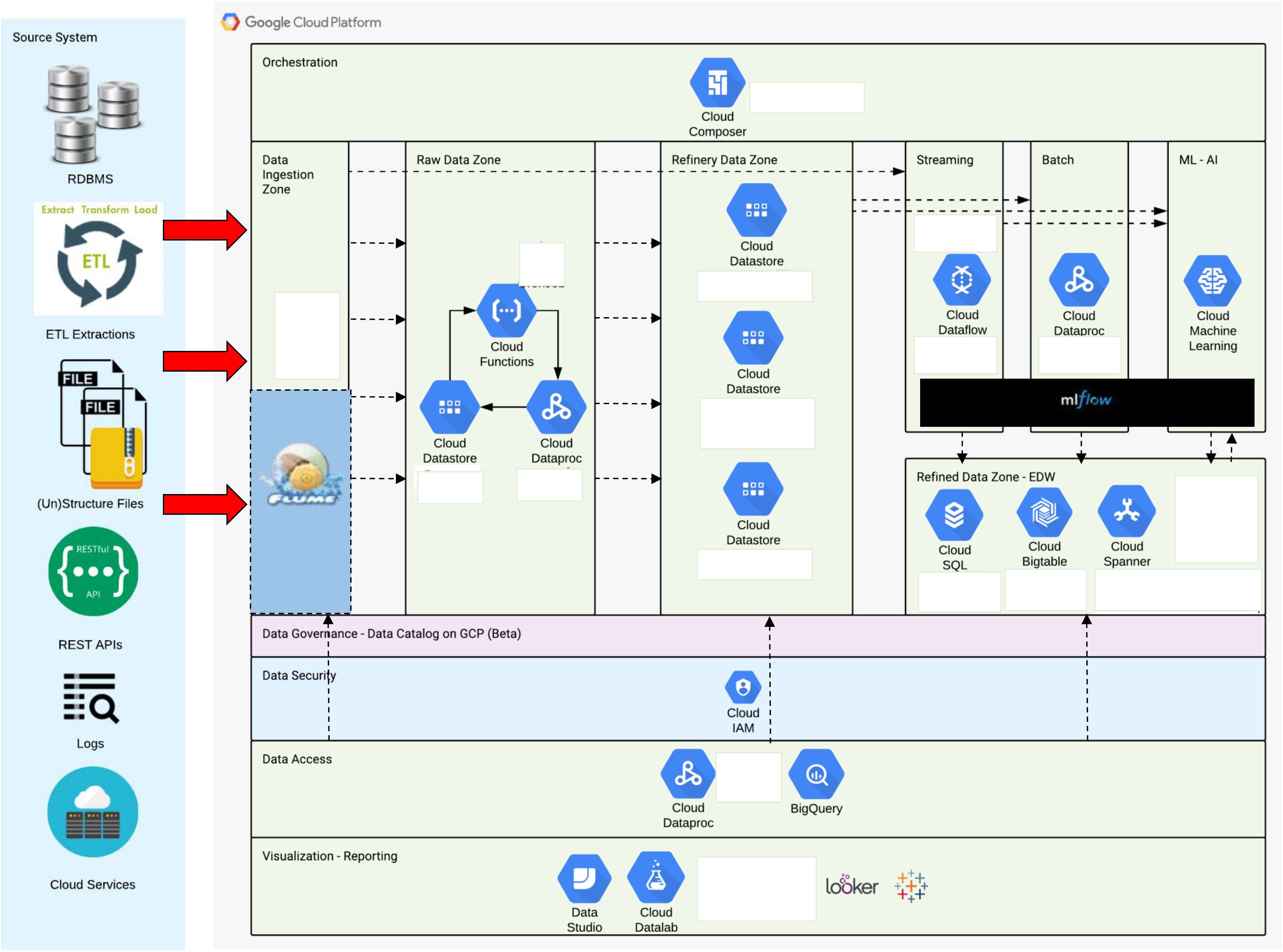
1. Click to NameNode and DataNode memory
2. Secondary NameNode memory
3. DataNode TaskTracker memory
4. OS memory
5. HDFS memory
6. DataNode CPU % utilization
7. Cluster nodes
8. Replication factor

1. Maximum mapper's slot numbers on
2. one node in a large data context
3. number of slots for the cluster:

4-Hadoop / Data Lake

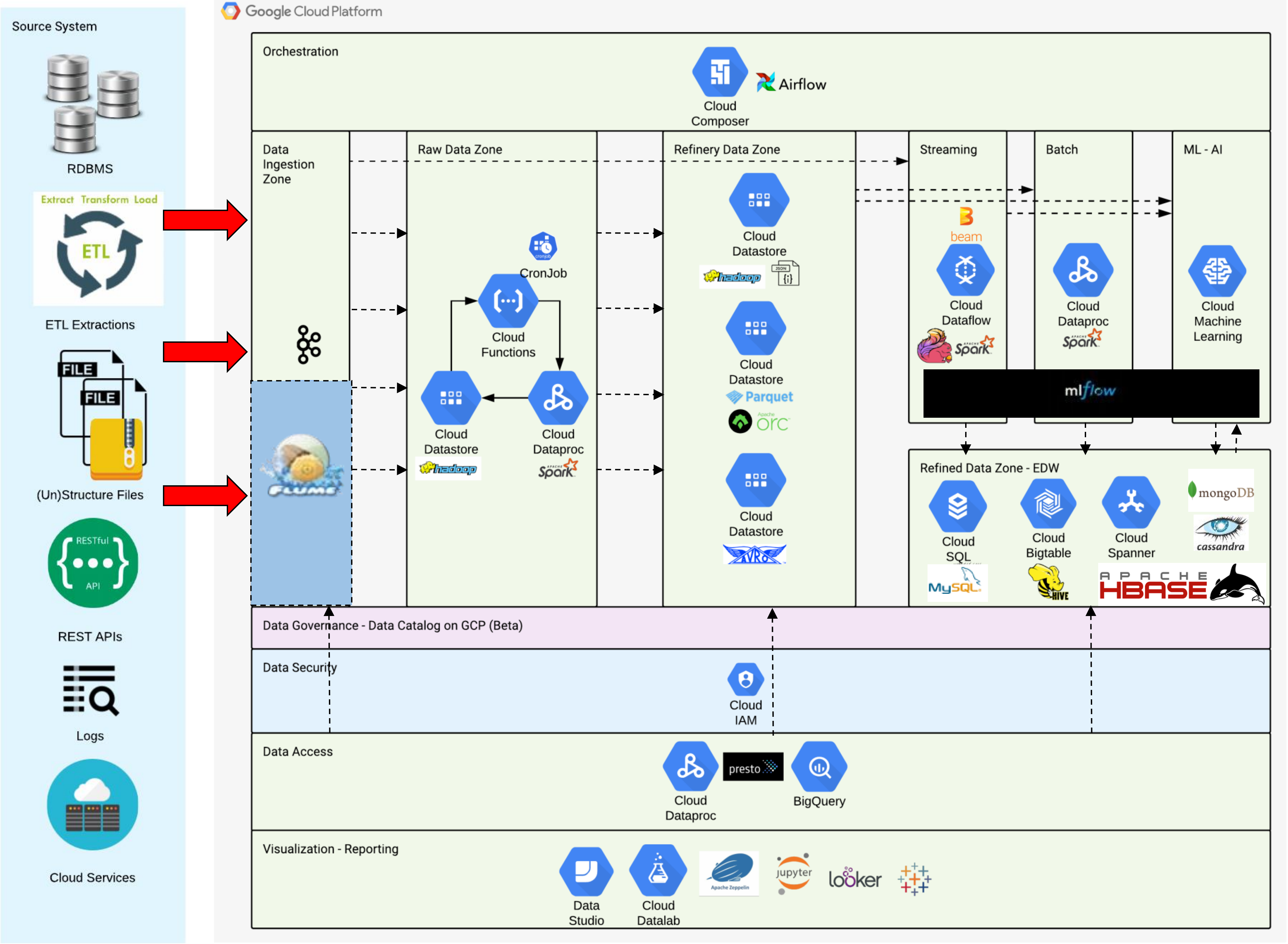
- Yes, we can develop a data lake by using the frameworks in Hadoop's Ecosystems.

Data Lake



Please, note that this design covers both approaches (open sources, cloud services) to expose the difference of the aspect of views. Any different kind of version of design might be curated by using different frameworks.

Data Lake



Source System

RDBMS

Extract Transform Load

ETL

ETL Extractions

FILE

FILE

(Un)Structure Files

RESTful

API

REST APIs

Logs

Cloud Services

Google Cloud Platform

Orchestration

Cloud Composer

Airflow

Data Ingestion Zone

FLUME

Raw Data Zone

CronJob

Cloud Functions

Cloud Datastore

hadoop

Cloud Dataproc

Spark

Refinery Data Zone

Cloud Datastore

hadoop

JSON

Cloud Datastore

Parquet

orc

Cloud Datastore

AVRO

Streaming

beam

Cloud Dataflow

Spark

Batch

Cloud Dataproc

Spark

ML - AI

Cloud Machine Learning

mlflow

Refined Data Zone - EDW

Cloud SQL

MySQL

Cloud Bigtable

HIVE

Cloud Spanner

mongoDB

cassandra

APACHE HBASE

Data Governance - Data Catalog on GCP (Beta)

Data Security

Cloud IAM

Data Access

Cloud Dataproc

presto

BigQuery

Visualization - Reporting

Data Studio

Cloud Datalab

Apache Zeppelin

jupyter

looker

Please, note that this design covers both approaches (open sources, cloud services) to expose the difference of the aspect of views. Any different kind of version of design might be curated by using different frameworks.

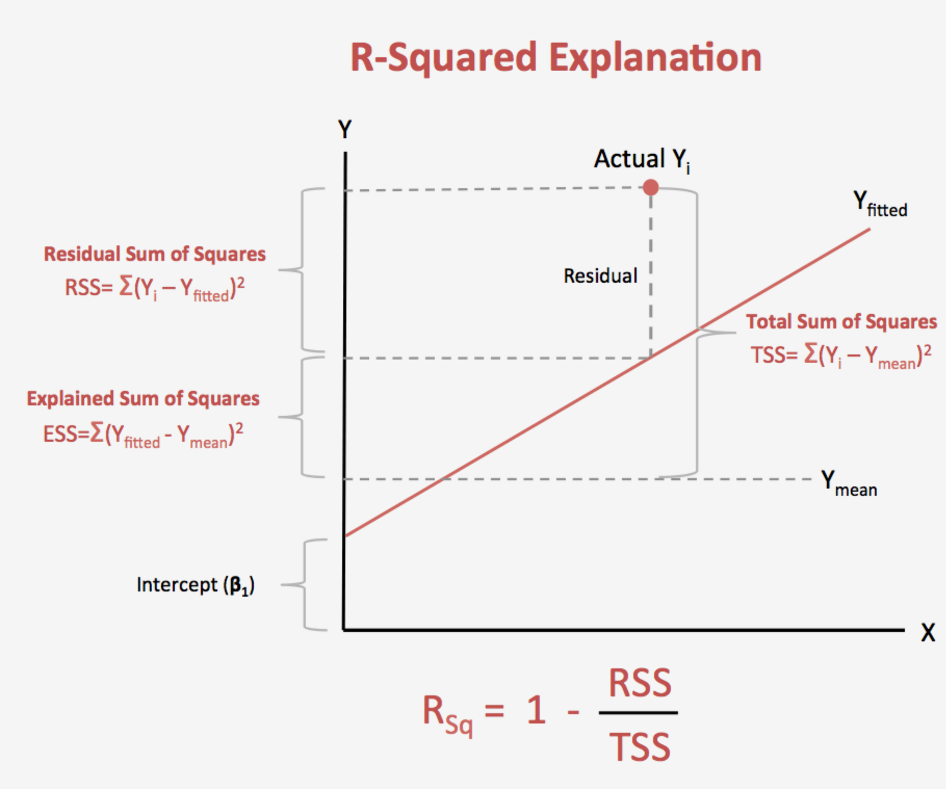
Data Scientist (Use Cases)

Erdi Ölmezoğulları

1.How would you prepare your data for analysis?

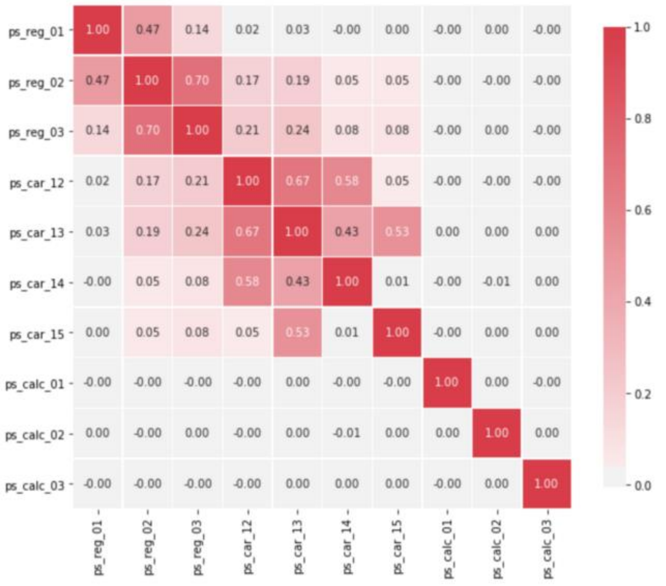
- Data Cleaning
  - Impossible or otherwise incorrect values for specific variables
  - Cases in the data who met exclusion criteria and shouldn’t be in the study
  - Duplicate cases
  - Missing data and outliers
  - Skip-pattern or logic breakdowns
- Encoding categorical data
- Feature Scaling
- Creating New Variables
- Formatting the Variables

2.How would you find correlated features in your data?



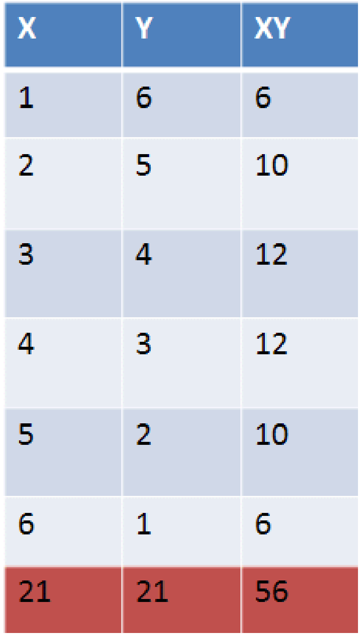
R^2

Correlation Matrix

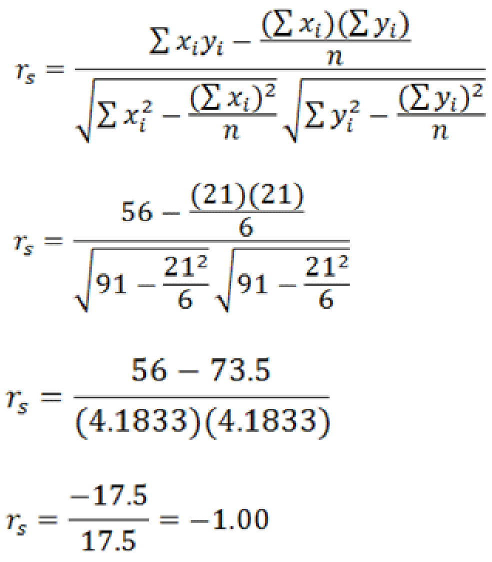


	ps_reg_01	ps_reg_02	ps_reg_03	ps_car_12	ps_car_13	ps_car_14	ps_car_15	ps_calc_01	ps_calc_02	ps_calc_03
ps_reg_01	1.00	0.47	0.14	0.02	0.03	-0.00	0.00	-0.00	0.00	-0.00
ps_reg_02	0.47	1.00	0.70	0.17	0.19	0.05	0.05	-0.00	-0.00	-0.00
ps_reg_03	0.14	0.70	1.00	0.21	0.24	0.08	0.08	-0.00	0.00	-0.00
ps_car_12	0.02	0.17	0.21	1.00	0.67	0.58	0.05	-0.00	-0.00	-0.00
ps_car_13	0.03	0.19	0.24	0.67	1.00	0.43	0.53	0.00	0.00	0.00
ps_car_14	-0.00	0.05	0.08	0.58	0.43	1.00	0.01	-0.00	-0.01	0.00
ps_car_15	0.00	0.05	0.08	0.05	0.53	0.01	1.00	-0.00	0.00	0.00
ps_calc_01	-0.00	-0.00	-0.00	-0.00	0.00	-0.00	-0.00	1.00	0.00	-0.00
ps_calc_02	0.00	-0.00	0.00	-0.00	0.00	-0.01	0.00	0.00	1.00	0.00
ps_calc_03	-0.00	-0.00	-0.00	-0.00	0.00	0.00	0.00	-0.00	0.00	1.00

1.0
0.8
0.6
0.4
0.2
0.0



X	Y	XY
1	6	6
2	5	10
3	4	12
4	3	12
5	2	10
6	1	6
21	21	56



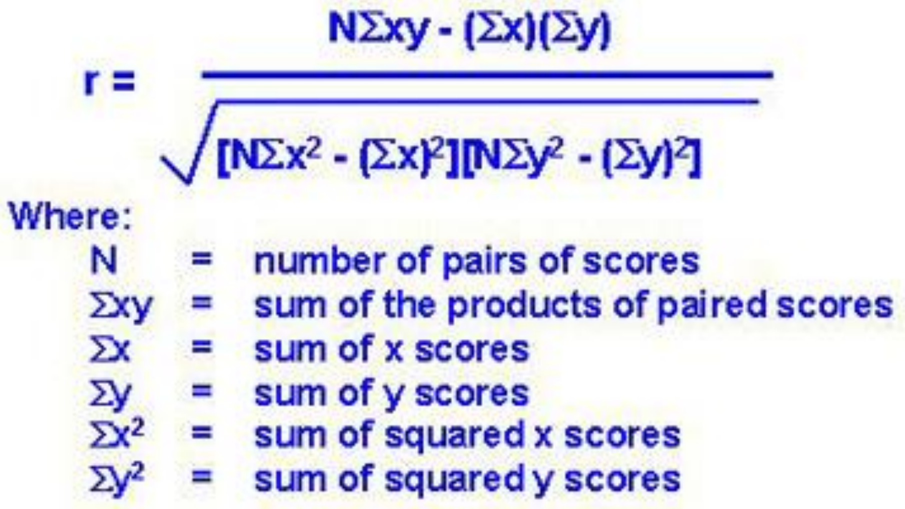
$$r_s = \frac{\sum x_i y_i - \frac{(\sum x_i)(\sum y_i)}{n}}{\sqrt{\sum x_i^2 - \frac{(\sum x_i)^2}{n}}\sqrt{\sum y_i^2 - \frac{(\sum y_i)^2}{n}}}$$

$$r_s = \frac{56 - \frac{(21)(21)}{6}}{\sqrt{91 - \frac{21^2}{6}}\sqrt{91 - \frac{21^2}{6}}}$$

$$r_s = \frac{56 - 73.5}{(4.1833)(4.1833)}$$

$$r_s = \frac{-17.5}{17.5} = -1.00$$

Spearman Correlation



$$r = \frac{N\Sigma xy - (\Sigma x)(\Sigma y)}{\sqrt{[N\Sigma x^2 - (\Sigma x)^2][N\Sigma y^2 - (\Sigma y)^2]}}$$

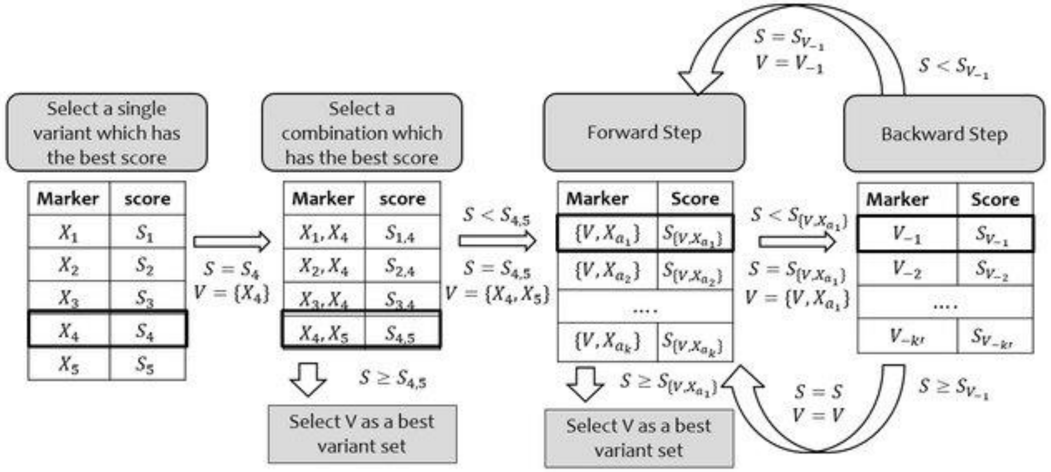
Where:

N	=	number of pairs of scores
Σxy	=	sum of the products of paired scores
Σx	=	sum of x scores
Σy	=	sum of y scores
Σx^2	=	sum of squared x scores
Σy^2	=	sum of squared y scores

Pearson Correlation

3.While working on a data set, how do you select important variables? Explain your methods.

- Remove the correlated variables prior to selecting important variables
- Use linear regression and select variables based on p values
- Use Forward Selection, Backward Selection, Stepwise Selection
- Use Random Forest, Xgboost and plot variable importance chart
- Use Lasso Regression
- Measure information gain for the available set of features and select top n features accordingly.



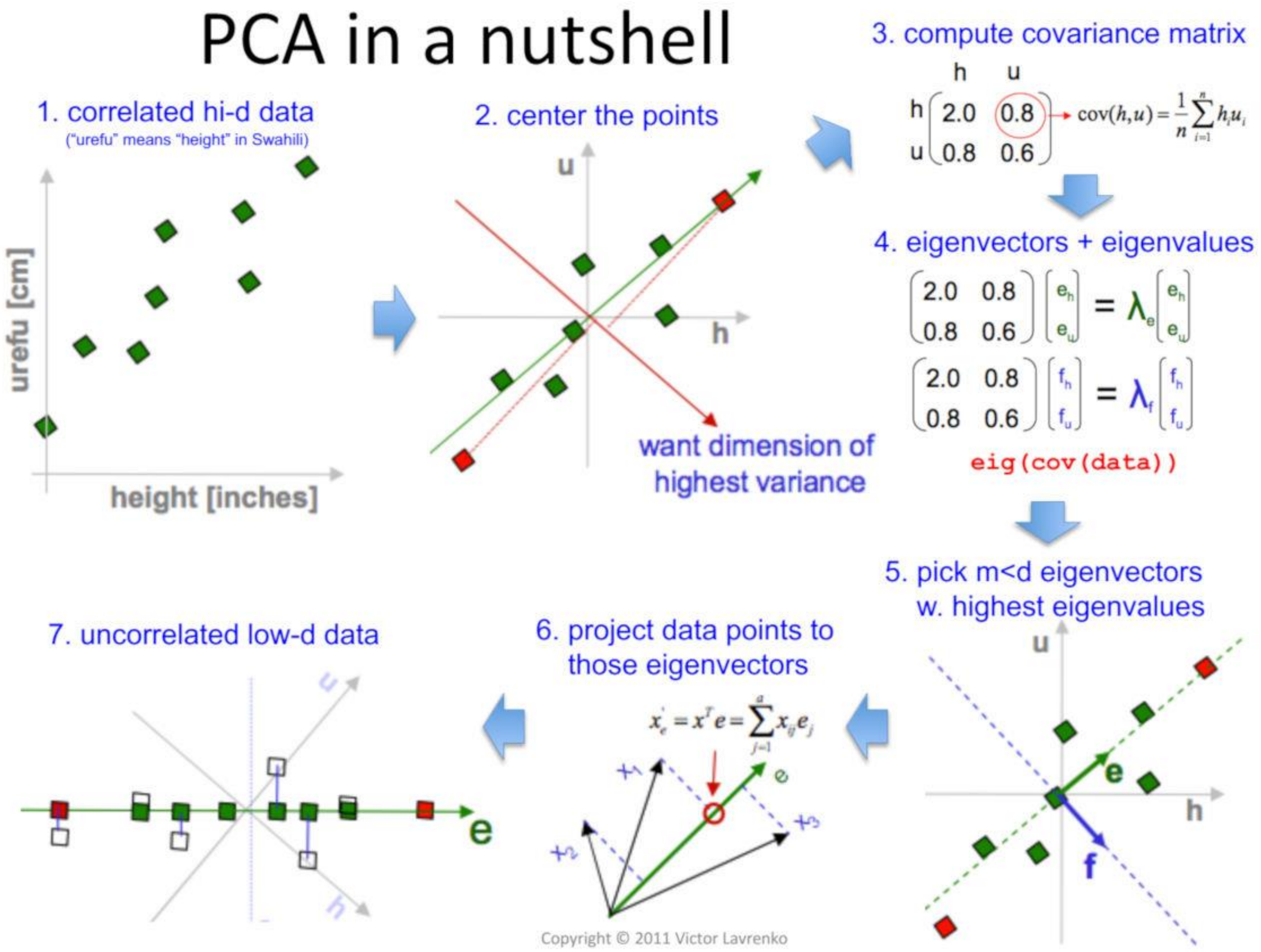
Marker	score
X_1	S_1
X_2	S_2
X_3	S_3
X_4	S_4
X_5	S_5

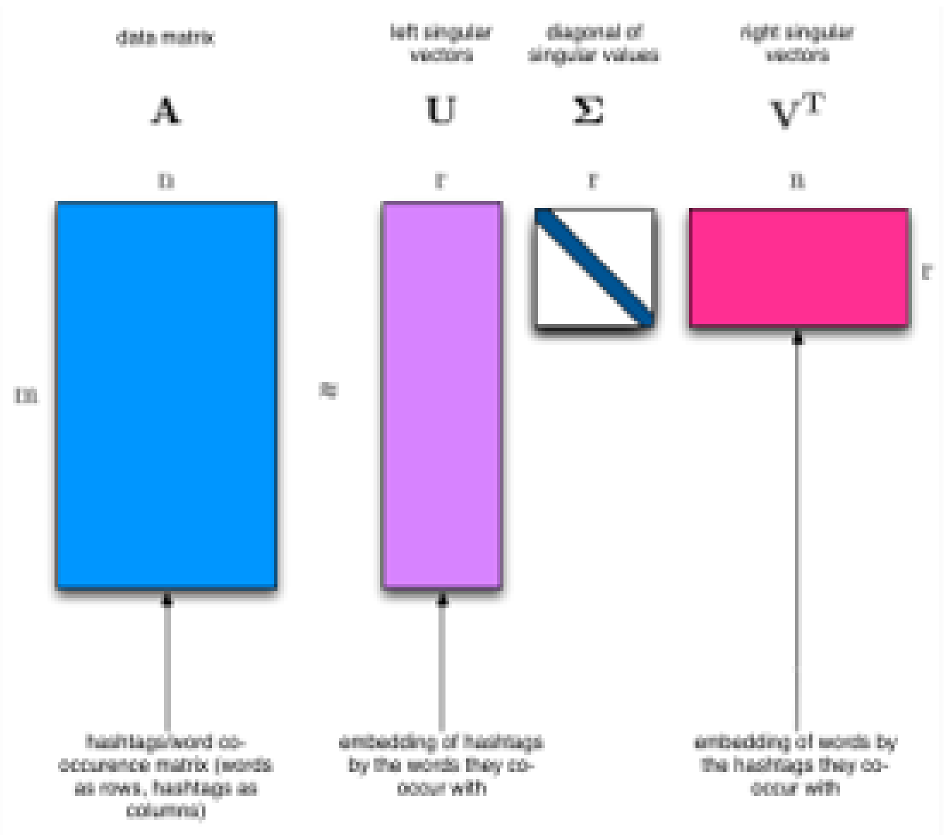
Marker	score
X_1, X_4	$S_{1,4}$
X_2, X_4	$S_{2,4}$
X_3, X_4	$S_{3,4}$
X_4, X_5	$S_{4,5}$

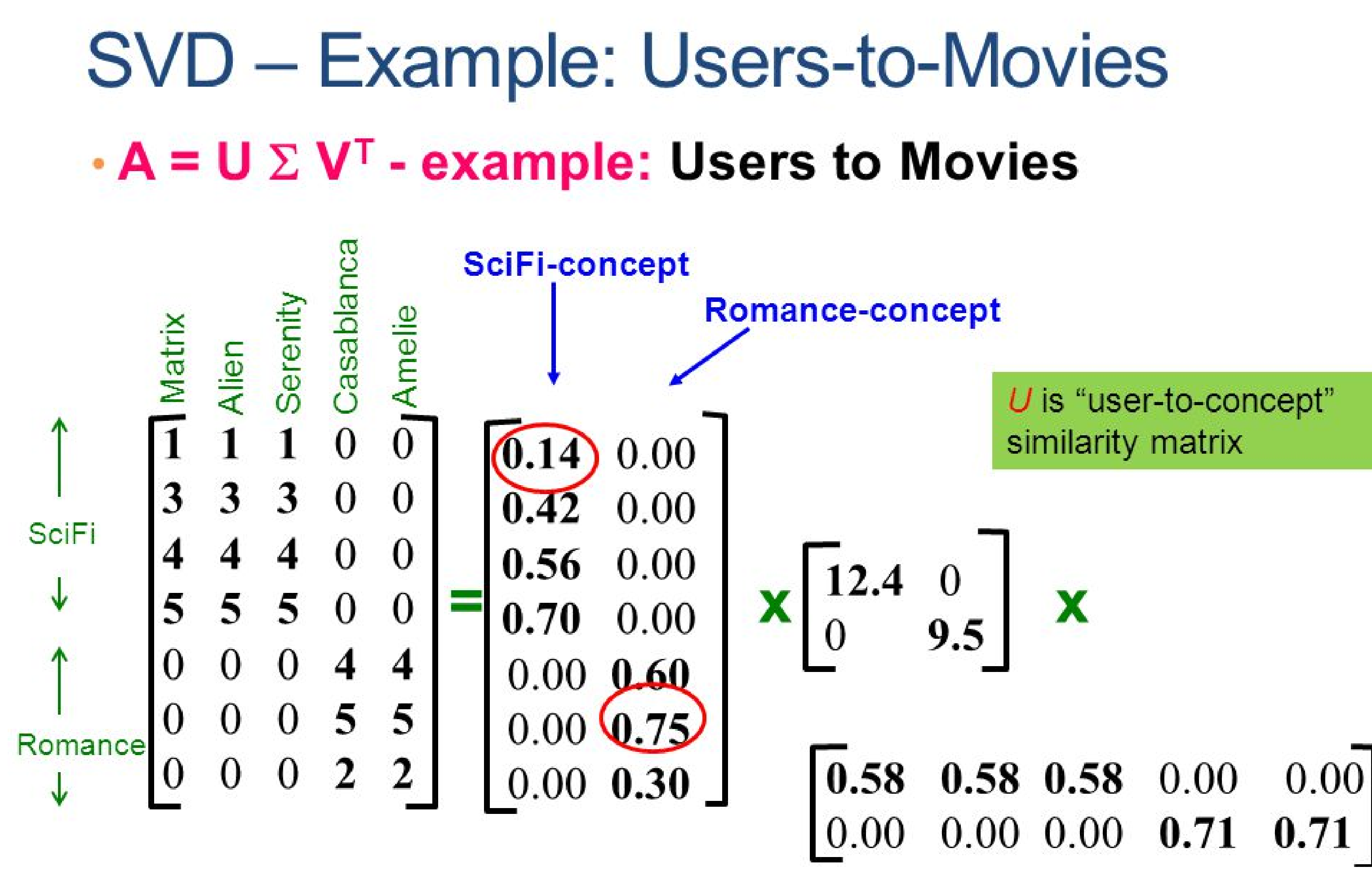
Marker	Score
$\{V, X_{a_1}\}$	$S_{\{V,X_{a_1}\}}$
$\{V, X_{a_2}\}$	$S_{\{V,X_{a_2}\}}$
....	
$\{V, X_{a_k}\}$	$S_{\{V,X_{a_k}\}}$

Marker	Score
V_{-1}	$S_{V_{-1}}$
V_{-2}	$S_{V_{-2}}$
....	
V_{-kr}	$S_{V_{-kr}}$

4.You are given a train data set having 1000 columns and 1 million rows ..


PCA in a nutshell
1. correlated hi-d data
("urefu" means "height" in Swahili)
urefu [cm]
height [inches]
2. center the points
u
h
want dimension of highest variance
3. compute covariance matrix
$\begin{array}{c|cc} & h & u \\ \hline h & 2.0 & 0.8 \\ u & 0.8 & 0.6 \end{array}$
$\mathrm{cov}(h,u) = \frac{1}{n}\sum_{i=1}^{n} h_i u_i$
4. eigenvectors + eigenvalues
$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}\begin{pmatrix} e_h \\ e_u \end{pmatrix} = \lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$
$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}\begin{pmatrix} f_h \\ f_u \end{pmatrix} = \lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$
eig(cov(data))
5. pick m<d eigenvectors w. highest eigenvalues
u
h
e
f
6. project data points to those eigenvectors
$x_e' = x^T e = \sum_{j=1}^{d} x_{ij} e_j$
x1
x2
x3
e
7. uncorrelated low-d data
u
h
e
Copyright © 2011 Victor Lavrenko


data matrix
A
left singular vectors
U
diagonal of singular values
Σ
right singular vectors
V^T
n
r
r
n
m
≈
r
hashtags/word co-occurance matrix (words as rows, hashtags as columns)
embedding of hashtags by the words they co-occur with
embedding of words by the hashtags they co-occur with



SVD – Example: Users-to-Movies

- **A = U Σ V^T - example: Users to Movies**

SciFi-concept

Romance-concept

$$
\begin{array}{c|ccccc}
 & \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\
\hline
\text{SciFi} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
 & \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
 & \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
 & \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
\text{Romance} & 0 & 0 & 0 & \mathbf{4} & \mathbf{4} \\
 & 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
 & 0 & 0 & 0 & \mathbf{2} & \mathbf{2}
\end{array}
=
\begin{bmatrix}
\mathbf{0.14} & 0.00 \\
\mathbf{0.42} & 0.00 \\
\mathbf{0.56} & 0.00 \\
\mathbf{0.70} & 0.00 \\
0.00 & \mathbf{0.60} \\
0.00 & \mathbf{0.75} \\
0.00 & \mathbf{0.30}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 \\
0 & \mathbf{9.5}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.58} & \mathbf{0.58} & \mathbf{0.58} & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & \mathbf{0.71} & \mathbf{0.71}
\end{bmatrix}
$$

U is “user-to-concept” similarity matrix

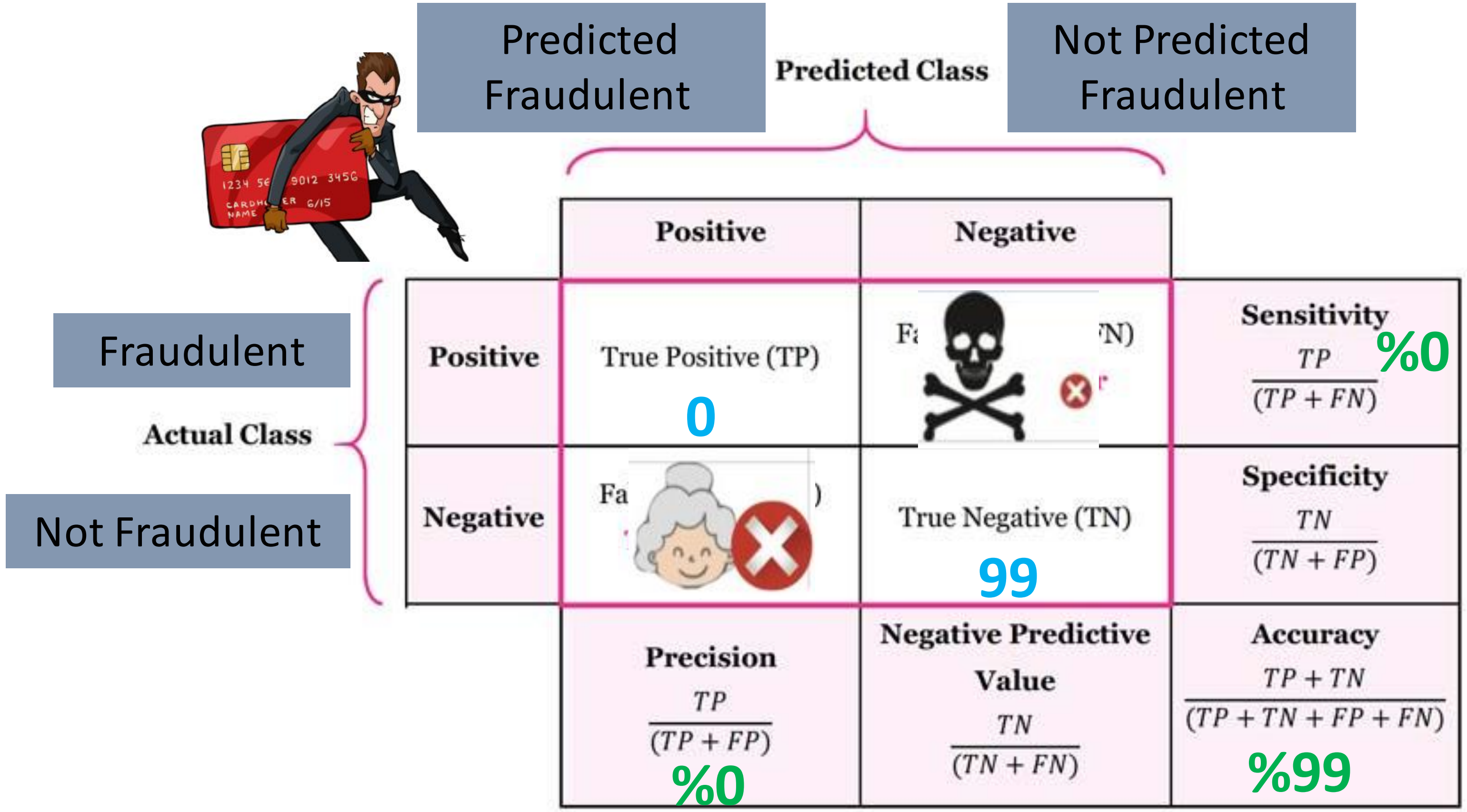
5.You are given a data set on fraud detection.

			Predicted Class		
			Predicted Fraudulent	Not Predicted Fraudulent	
			Positive	Negative	
Actual Class	Fraudulent	Positive	True Positive (TP) 0	False Negative (FN) Type II Error 1	Sensitivity $\frac{TP}{(TP + FN)}$ %0
	Not Fraudulent	Negative	False Positive (FP) Type I Error 0	True Negative (TN) 99	Specificity $\frac{TN}{(TN + FP)}$
			Precision $\frac{TP}{(TP + FP)}$ %0	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$ %99

- Since imbalanced data, accuracy is not a good metric.
- If there is 1 fraud out of 100 transactions and my model says all are clean non fraud transactions, it is 99% accuracy but not a very good model because it can’t detect fraud. We should look at precision-recall metrics.

High Precision <=> Less FP
High Recall <=> Less FN

5.You are given a data set on fraud detection.

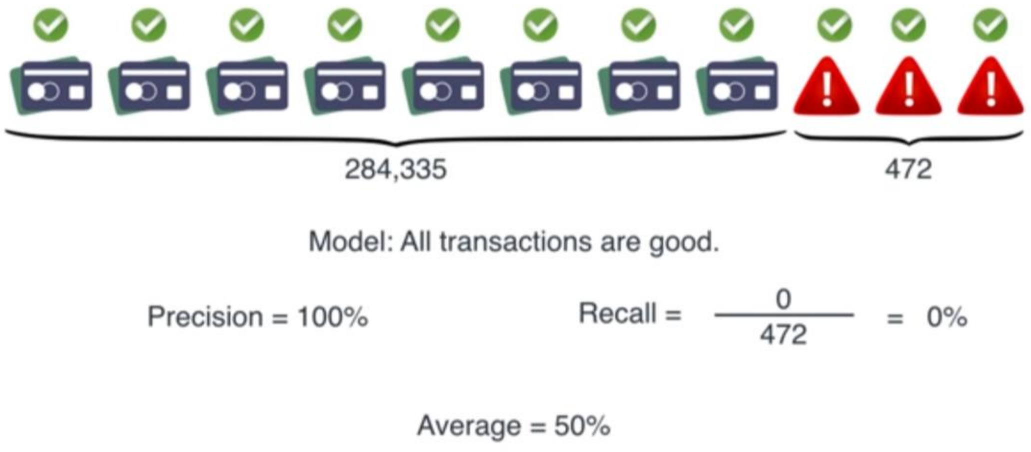


		Predicted Class		
		Predicted Fraudulent Positive	Not Predicted Fraudulent Negative	
Actual Class	Fraudulent Positive	True Positive (TP) 0	Fa N)	Sensitivity $\frac{TP}{(TP+FN)}$ %0
	Not Fraudulent Negative	Fa)	True Negative (TN) 99	Specificity $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$ %0	Negative Predictive Value $\frac{TN}{(TN+FN)}$	Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$ %99

- Since imbalanced data, accuracy is not a good metric.
- If there is 1 fraud out of 100 transactions and my model says all are clean non fraud transactions, it is 99% accuracy but not a very good model because it can't detect fraud. We should look at precision-recall metrics.

High Precision <=> Less FP
High Recall <=> Less FN

5.You are given a data set on fraud detection.

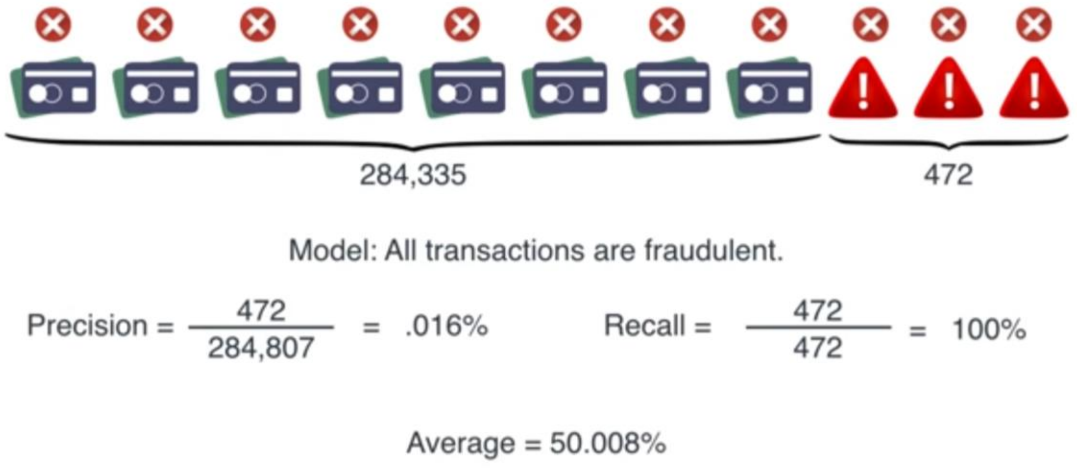


Model: All transactions are good.

Precision = 100%

Recall = $\frac{0}{472}$ = 0%

Average = 50%

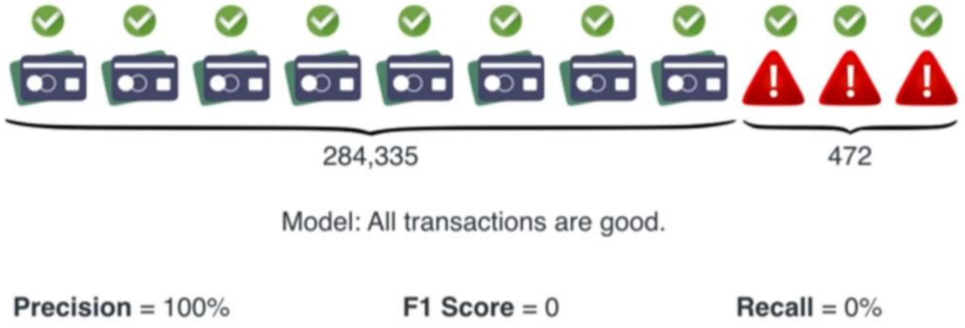


Model: All transactions are fraudulent.

Precision = $\frac{472}{284,807}$ = .016%

Recall = $\frac{472}{472}$ = 100%

Average = 50.008%

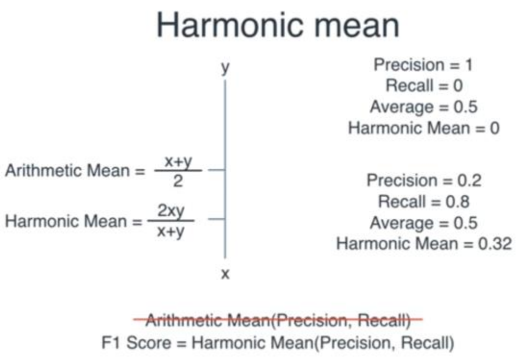


Model: All transactions are good.

Precision = 100%

F1 Score = 0

Recall = 0%



Harmonic mean

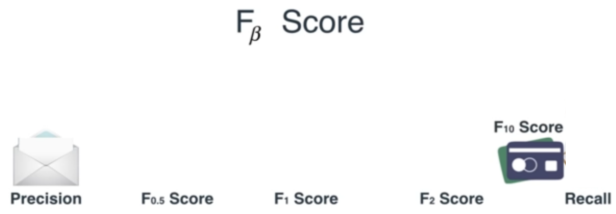
Arithmetic Mean = $\frac{x+y}{2}$

Harmonic Mean = $\frac{2xy}{x+y}$

Precision = 1
Recall = 0
Average = 0.5
Harmonic Mean = 0

Precision = 0.2
Recall = 0.8
Average = 0.5
Harmonic Mean = 0.32

~~Arithmetic Mean(Precision, Recall)~~
F1 Score = Harmonic Mean(Precision, Recall)



F_β Score

5.You are given a data set on fraud detection.

- We can use undersampling, oversampling or SMOTE to make the data balanced.
- We can alter the prediction threshold value by doing probability caliberation and finding a optimal threshold using AUC-ROC curve.
- We can assign weight to classes such that the minority classes gets larger weight.
- We can also use anomaly detection.

6.You are working on a time series data set.

- We forecast the time series of interest y assuming that it has a linear relationship with other time series x. For example, we might wish to forecast monthly sales y using total advertising spend x as a predictor. Or we might forecast daily electricity demand y using temperature x1 and the day of week x2 as predictors.
- On the other hand, a decision tree algorithmis known to work best to detect nonlinear problem. That is the reason why decision tree failed since it does not robust predictions like time series regression.

7. Main Cases

Customer			
customer_id	birthday	gender	city

Purchase				
purchase_id	customer_id	product_id	date	price

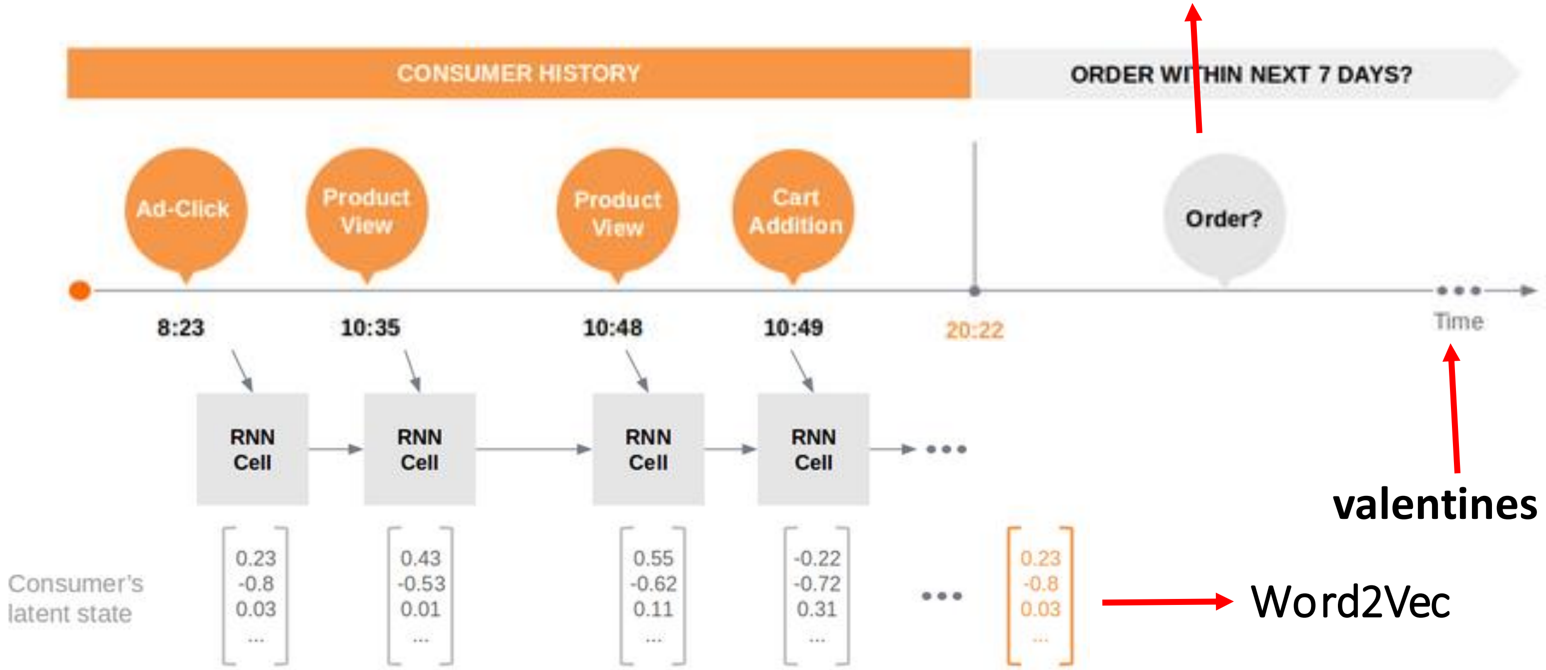
Product				
product_id	height	color	category	type

Events						
customer_id	product_id	session_id	date	event_type	page_path	...

1. We need to improve sales of "teddy bear" product in valentines day. How would you build a model which predicts customers who likely to buy a teddy bear

- What kind of model you would build? Explain your model
  - I can build a LSTM based model over clickstream dataset.
  - And then, a customer segmentation by building Decision Tree (Random Forest) and using result of LSTM to make advertisement etc.
- How would you build your model? (Explain all steps including data preparation)
  - We create sequence by session_id and order by date and time.
    Costumer0 : <page_path:product_id:event_type, … , …,>
    Costumer0: <…>
    Custumer1: <…>

Order within
1 week **earlier** for **valentines day**



valentines day

Word2Vec

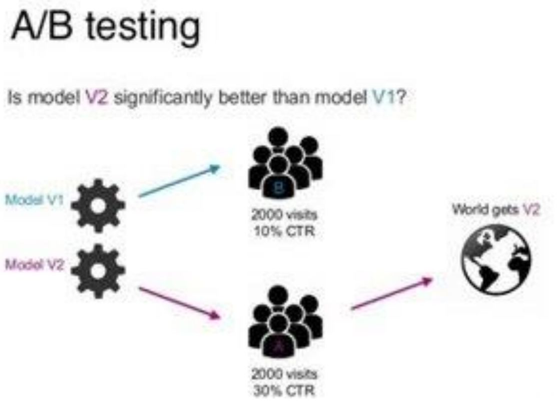
1. We need to improve sales of "teddy bear" product in valentines day. How would you build a model which predicts customers who likely to buy a teddy bear

- (cont.) How would you build your model? (Explain all steps including data preparation)
    - I can use word2vec or graph embedding to create embedding matrix.
    - I label the sequence which has teddy bear 1 otherwise 0.
    - Spitted data into test, train, val. (80-10-10)
    - And applying K-fold model in building.
    - After building model, all sequences to calculate probability of buying teddy bear for each sequence over model. I calculate mean of probability for each customer.
    - In customer segmentation, we can create a bunch of feature total purchasing history (weekly, montly, yearly), and customer information (age, gender). OHE
- How would you test your model?
    - I check out bias-variance errors to make sure model is either underfit or overfit. In that case, valid metrics should be picked best accurate model.
- How would you find customers who likely to buy teddy bear whom has not bought it before?
    - I can apply the customers' sequence history on the LSTM model to calculate probability.

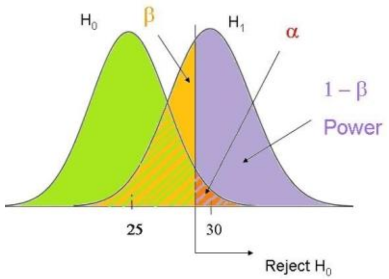
2.Marketing teams need to know how customers behave?

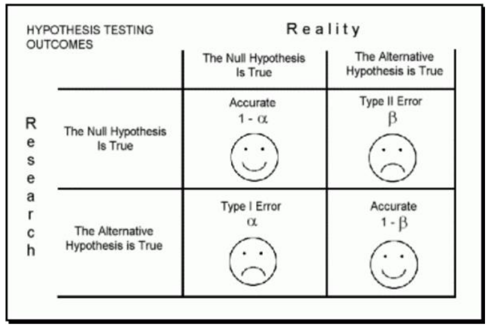
- What kind of model you would build? How would you extract rules?
  - In previous question, I gave answer partially. Decision Tree or Random Forest could be reasonable.
- How would you build your model? (Explain all steps including data preparation)
  - In customer segmentation, we can create a bunch of new features, such as total purchasing history (weekly, monthly, yearly), and customer information (age, gender). OHE.
  - Some aggregated features (mean, max, min) of total purchasing price and times for each customer.

2.Marketing teams need to know how customers behave?



- How would you test your model?
  - I check out bias-variance errors to make sure model is either underfit or overfit. In that case, valid metrics should be picked best accurate model.
  - AB testing.
- How can you make sure that founded segments will help marketing teams to improve revenue or reduce costs?
  - We make AB testing to make sure the model is working flawless on the costumers. (null and alternative hypothesis). It could be phone call or survey.
  - After I check out it, we can calculate ROI (Return On Invesment) since we can measure impacts of results on advertisement, and campaign.
  - So, they can evaluate the performance of the advertisement and campaign methods (newspaper, tv, radio etc.) in that way.





HYPOTHESIS TESTING OUTCOMES

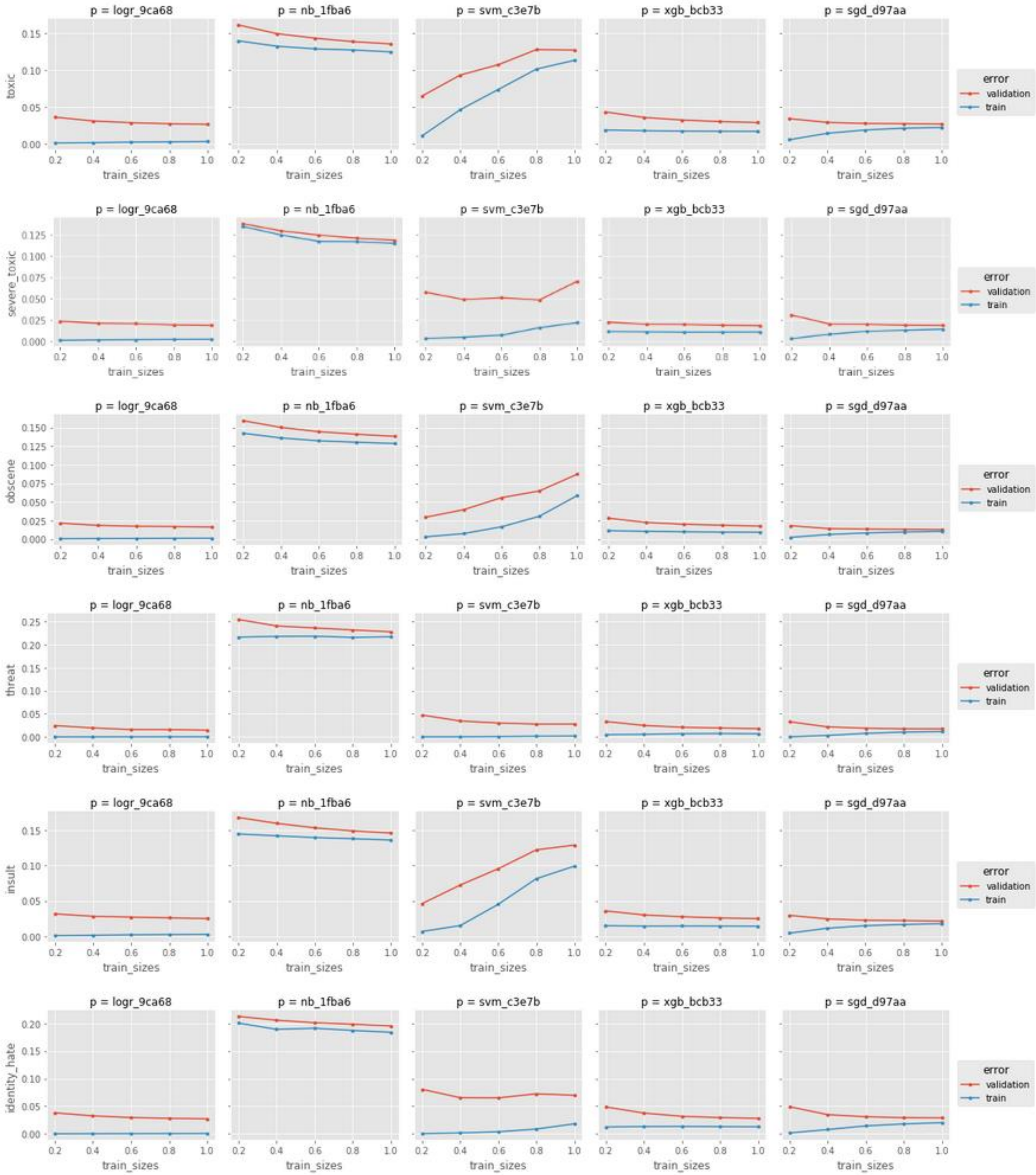
		Reality	
		The Null Hypothesis Is True	The Alternative Hypothesis is True
Research	The Null Hypothesis Is True	Accurate 1 - α ☺	Type II Error β ☹
	The Alternative Hypothesis is True	Type I Error α ☹	Accurate 1 - β ☺

Bias – Variance Learning Rate (extra)

- A pipeline consist of CountVectorizer, TfidfTransformer, and a classifier (e.g. Logistic Regression, Naive Bayes (NB), SVM, Xgboost, Stochastic Gradient Descent (SGD))

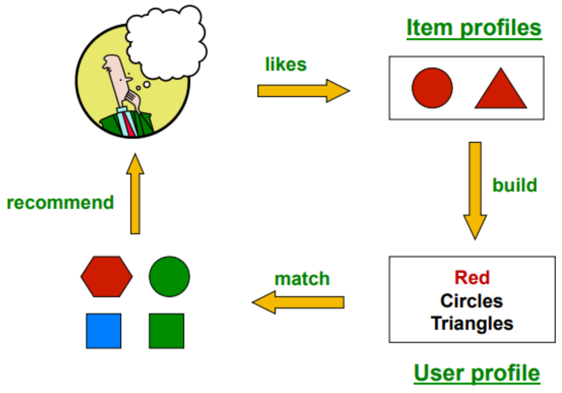
- It seems SGD is the best one amongst those classifiers on the graph since the gap between validation and train error (1 - AUC) is smaller than the others so it has low variance and low bias. The second one is Xgboost because it has similar trend like SGD.

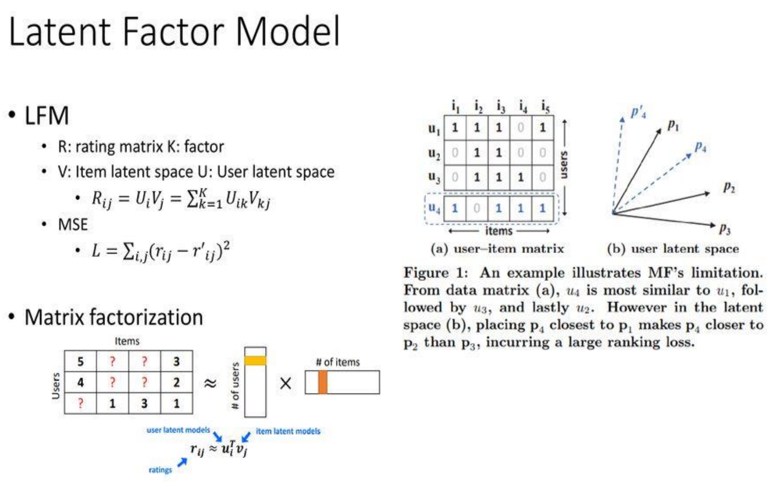
- We can think NB has also small gap between validation and train error. However, it has high train error than the others. So, it has high bias.
- In this case, SVM is not really working well.



3.We need a recommendation system to personalize offers on web site

- What kind of system you would build?
  - Content-Based
  - Collaborative Filtering
    - User-based filtering
    - Item-based filtering
  - Latent Factor Based





Latent Factor Model

- LFM
  - R: rating matrix K: factor
  - V: Item latent space U: User latent space
    - $R_{ij} = U_i V_j = \sum_{k=1}^{K} U_{ik} V_{kj}$
  - MSE
    - $L = \sum_{i,j} (r_{ij} - r'_{ij})^2$

- Matrix factorization

	Items			
Users	5	?	?	3
	4	?	?	2
	?	1	3	1

≈
# of users
×
# of items
user latent models
item latent models
ratings

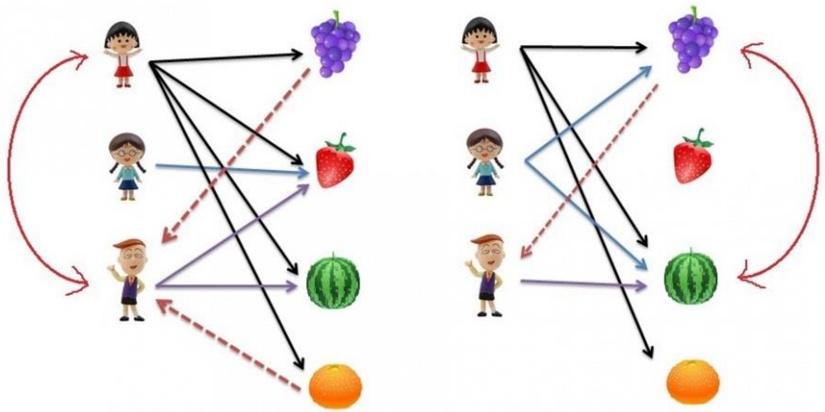
$r_{ij} \approx u_i^T v_j$

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	0	1
u_2	0	1	1	0	0
u_3	0	1	1	1	0
u_4	1	0	1	1	1

users
items
p'_4
p_1
p_4
p_2
p_3

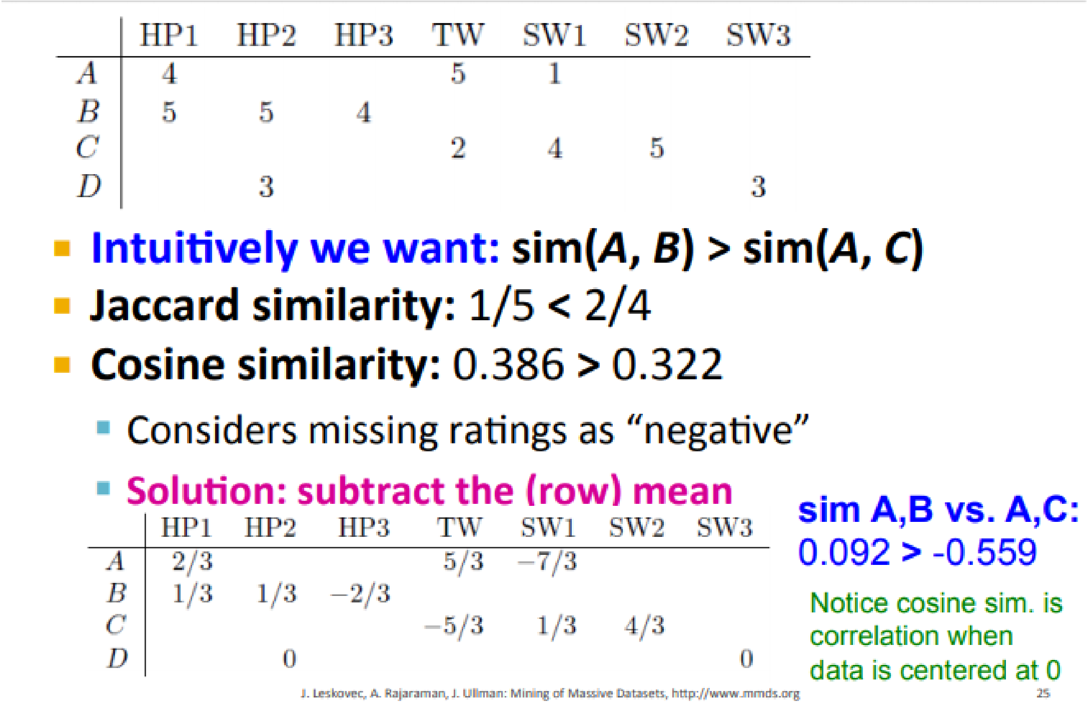
(a) user–item matrix (b) user latent space

Figure 1: An example illustrates MF's limitation. From data matrix (a), u_4 is most similar to u_1, followed by u_3, and lastly u_2. However in the latent space (b), placing $\mathbf{p}_4$ closest to $\mathbf{p}_1$ makes $\mathbf{p}_4$ closer to $\mathbf{p}_2$ than $\mathbf{p}_3$, incurring a large ranking loss.



User-based filtering Item-based filtering

- How would you build your system? (Explain all steps including data preparation)
  - **Content-Based :** Create user profile and item profiles such as vector, which consists of important words (TF-IDF). Measure similarity user profile and item by using cosine distance.
  - **Collaborative Filtering:**
    - **User-User:**
      - Finding Similar Users
        - Jaccard similarity measure
        - Cosine similarity measure
        - Pearson correlation coefficient



	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

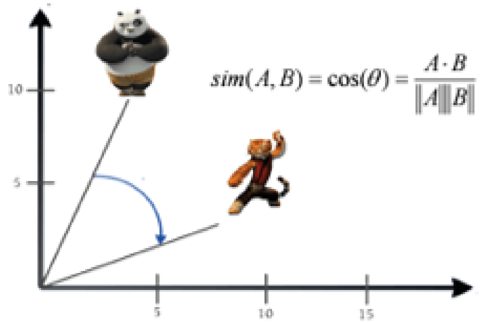
- **Intuitively we want: sim(*A*, *B*) > sim(*A*, *C*)**
- **Jaccard similarity:** 1/5 < 2/4
- **Cosine similarity:** 0.386 > 0.322
  - Considers missing ratings as “negative”
  - **Solution: subtract the (row) mean**

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	$2/3$			$5/3$	$-7/3$		
B	$1/3$	$1/3$	$-2/3$				
C				$-5/3$	$1/3$	$4/3$	
D		0					0

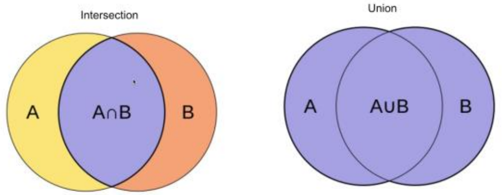
sim A,B vs. A,C:
0.092 > -0.559

Notice cosine sim. is correlation when data is centered at 0

J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org 25



Jaccard coefficient



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

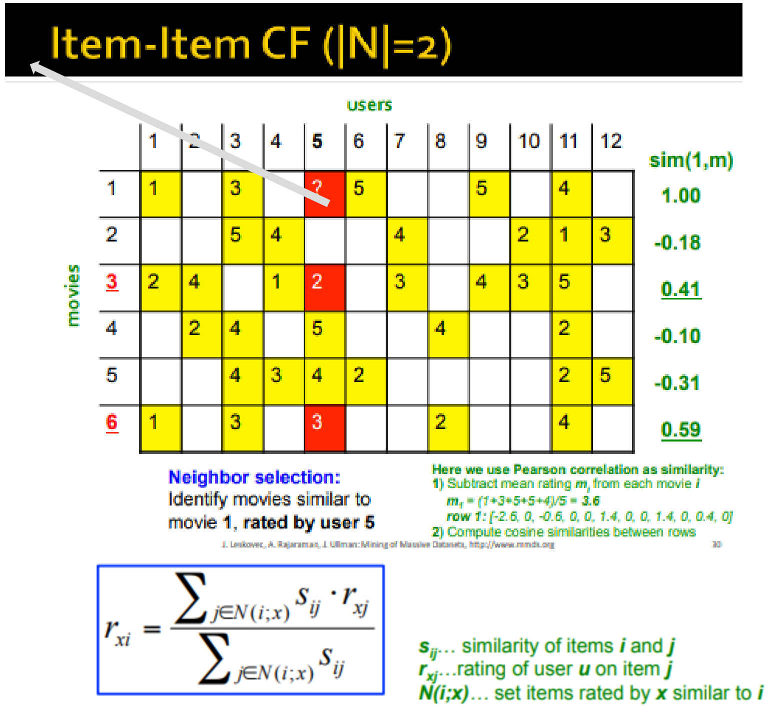
r1.5 = (0.41*2 + 0.59*3) / (0.41+0.59) = 2.6

- **(cont.) Item-item:**
  - For item i, find othersimilar items
  - Estimate rating for i based on ratings for similar items

In practice,
it has been observed that item-item often works better than user-user. Because, items are simpler, users have multiple tastes

- How would you test performance of your system?

Compute similarity weights:
$s_{1,3}$=0.41, $s_{1,6}$=0.59



Item-Item CF (|N|=2)

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		0.41
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		0.59

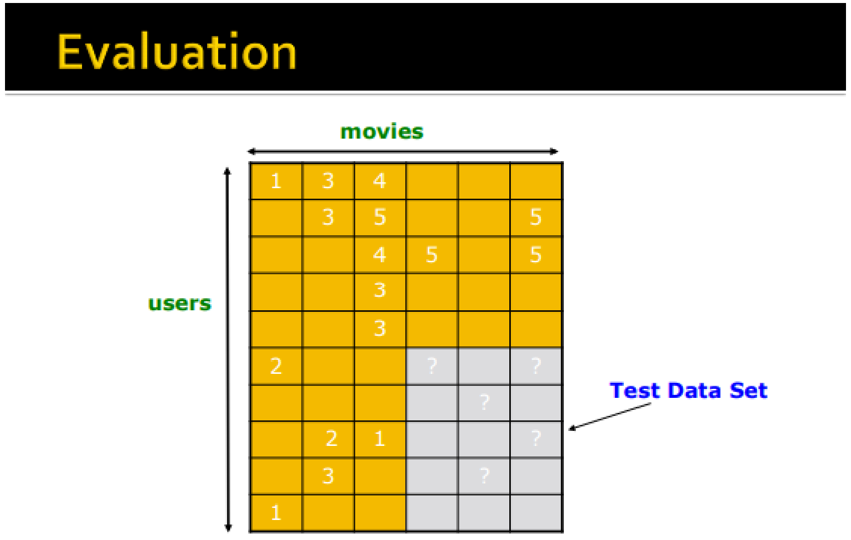
Neighbor selection:
Identify movies similar to movie **1**, **rated by user 5**

Here we use Pearson correlation as similarity:
1) Subtract mean rating m_i from each movie i
$m_1 = (1+3+5+5+4)/5 = 3.6$
row 1: [-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0.4, 0]
2) Compute cosine similarities between rows

J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org 30

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

s_{ij}… similarity of items *i* and *j*
r_{xj}…rating of user *u* on item *j*
N(i;x)… set items rated by *x* similar to *i*



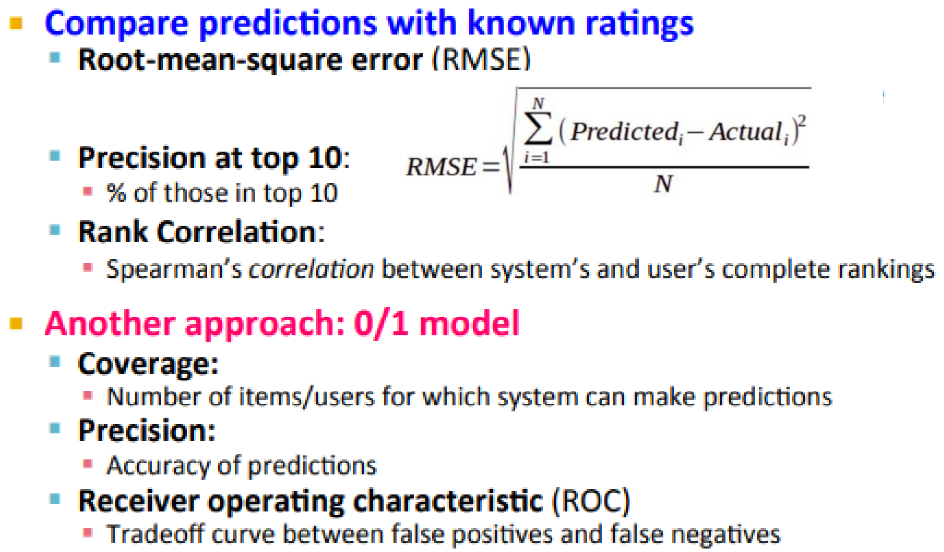
Evaluation

movies

users

1	3	4			
	3	5			5
		4	5		5
		3			
		3			
2			?		?
				?	
	2	1			?
	3			?	
1					

Test Data Set



- **Compare predictions with known ratings**
  - **Root-mean-square error** (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

  - **Precision at top 10**:
    - % of those in top 10
  - **Rank Correlation**:
    - Spearman's *correlation* between system's and user's complete rankings
- **Another approach: 0/1 model**
  - **Coverage:**
    - Number of items/users for which system can make predictions
  - **Precision:**
    - Accuracy of predictions
  - **Receiver operating characteristic** (ROC)
    - Tradeoff curve between false positives and false negatives

- How can we make (almost) sure that system recommend (almost) correct products even if there are not enough transactions for specified customer.
  - We need to perform AB testing for that model again.

+ Works for any kind of item

* No feature selec-on needed

- Cold Start:

* Need enough users in the system to find a match

- Sparsity:

* The user/ra-ngs matrix is sparse

* Hard to find users that have rated the same items

- First rater:

* Cannot recommend an item that has not been previously rated

* New items, Esoteric items

- Popularity bias:

* Cannot recommend items to someone with unique taste

* Tends to recommend popular items

Hyrid approches can help:

- Implement two or more different recommenders and combine predictors
- Item profiles for new item problem
- Demographics to deal with new user problem