Bringing A Deep Learning Approach to Production

Suqiang Song (Jack)
Director & Chapter Leader of Data/AI Engineering @ Mastercard
jackssqcyy@gmail.com
https://www.linkedin.com/in/suqiang-song-72041716/

Agenda

August 21th ,Wed, 1pm-4pm

Module 1: Deep Learning in Production (50 mins)

- Deep learning in enterprise application
- Deep Learning production lifecycle
- Tools/frameworks in production

Code Lab (30 mins)

- Benchmark between Spark Machine learning and Spark Deep learning with a user item propensity model example

Break (10 mins)

Live Demo(45 mins)

- Build an end to end AI Pipeline for AI as a Service with Kafka, NiFi, Spark Streaming and Keras on Spark

Module 2: AI/Deep Learning Engineers (30 mins)

- knowledges and skills are required for AI Engineer
- Career path for AI Engineer

Q & A (15 mins)

Course Prerequisites

- Install Docker at your local laptop or use public cloud account
- Fork the GitHub project

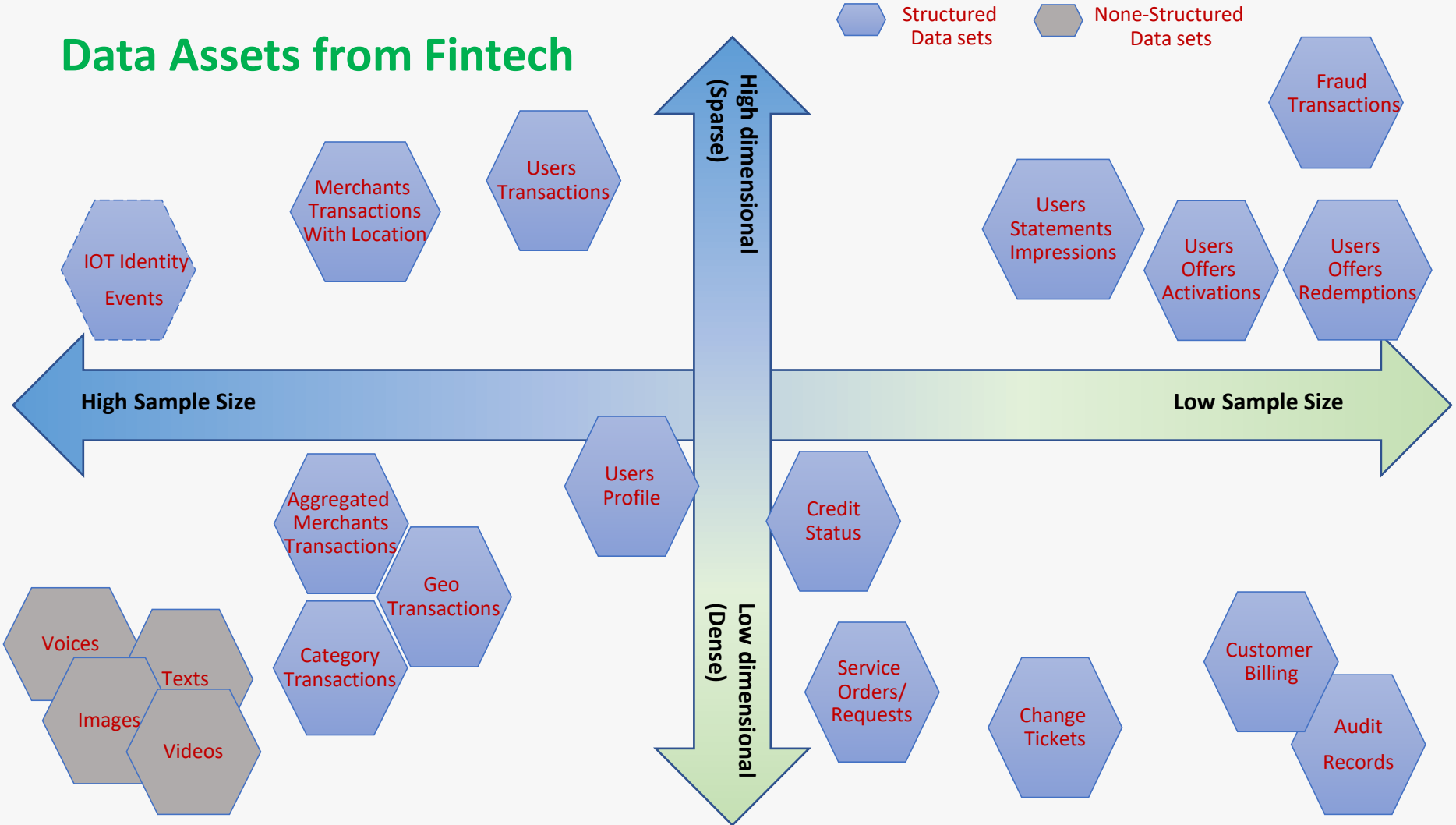
  https://github.com/jack1981/aaas-demo-aicamp

- Pull images to your Docker environment

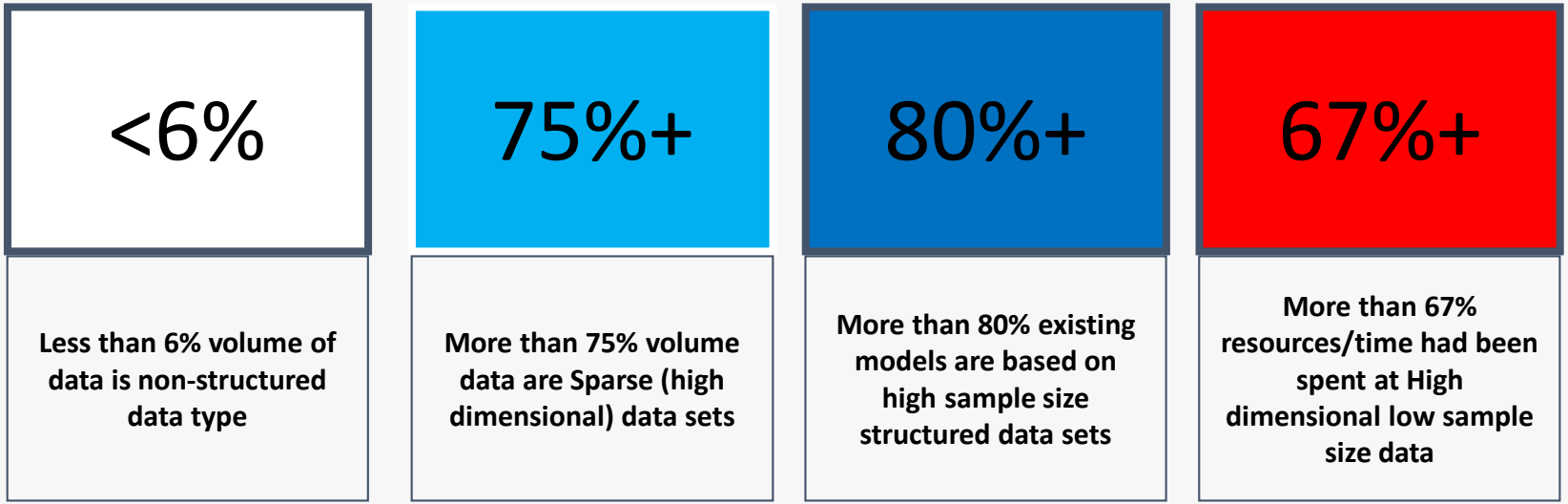
```
$ docker pull msba6212/aaas-demo-base

$ docker pull msba6212/kafka
```

Module 1

Data Assets from Fintech



Data characteristics from Fintech



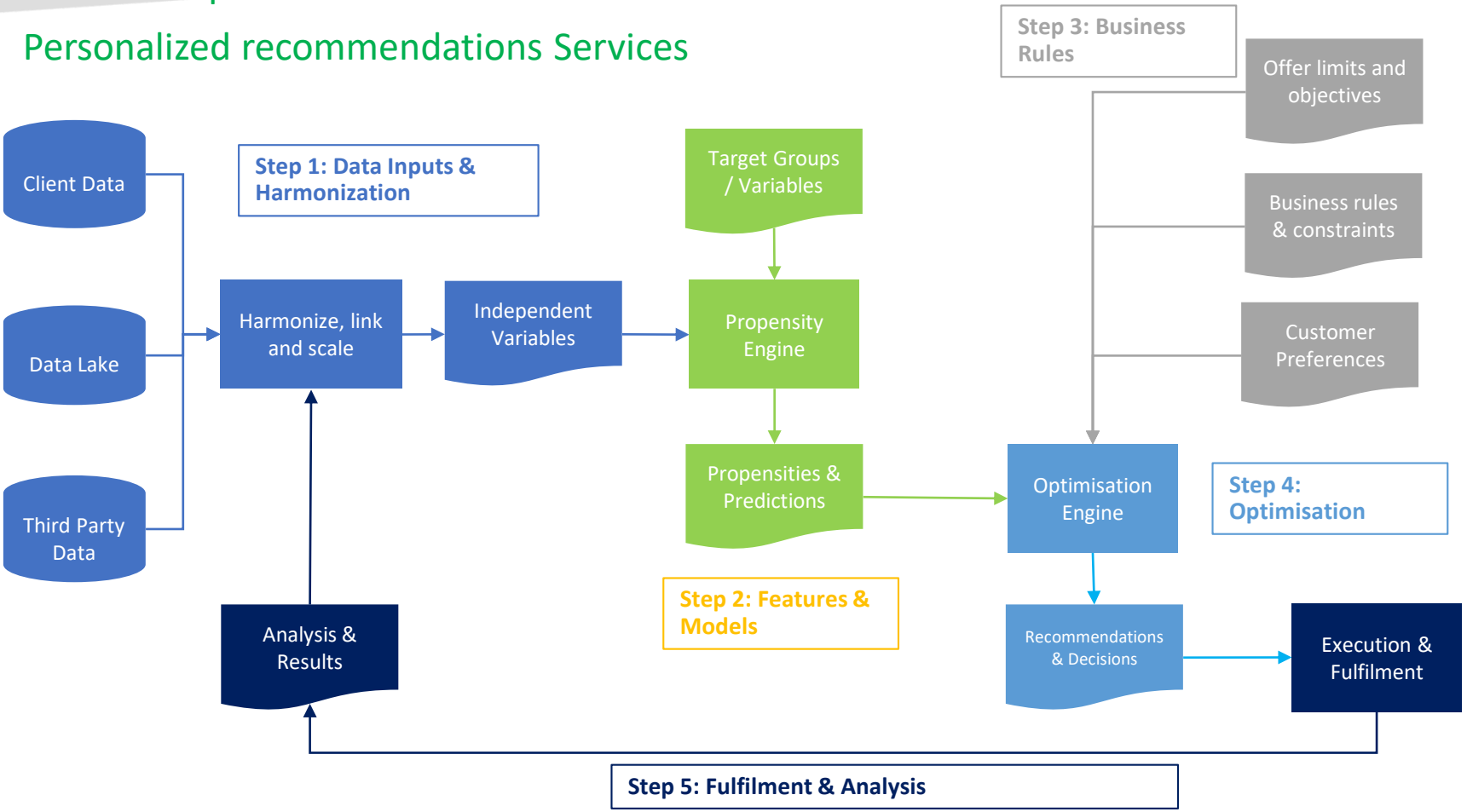
One of the most popular models meets the characteristics -- user item propensity model

- Propensity to buy
- Propensity to use
- Propensity to engage
- Propensity to contract
- Etc.

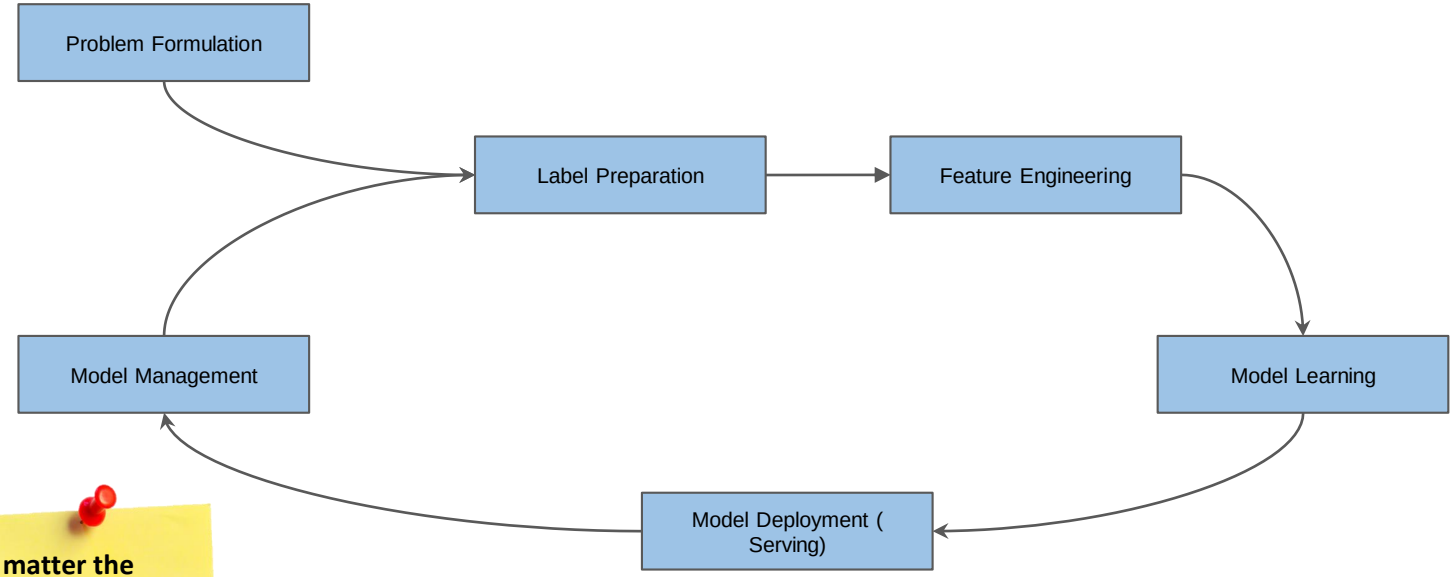
- Life Insurance
- Auto Insurance
- Homeowner's Insurance
- Mortgage
- Re-Financing
- Credit cards
- Personal Loans
- Investments
- Satellite TV
- Cable
- Netflix
- Online banking
- Online money management
- Voting
- Disease

Product requirement :

Personalized recommendations Services

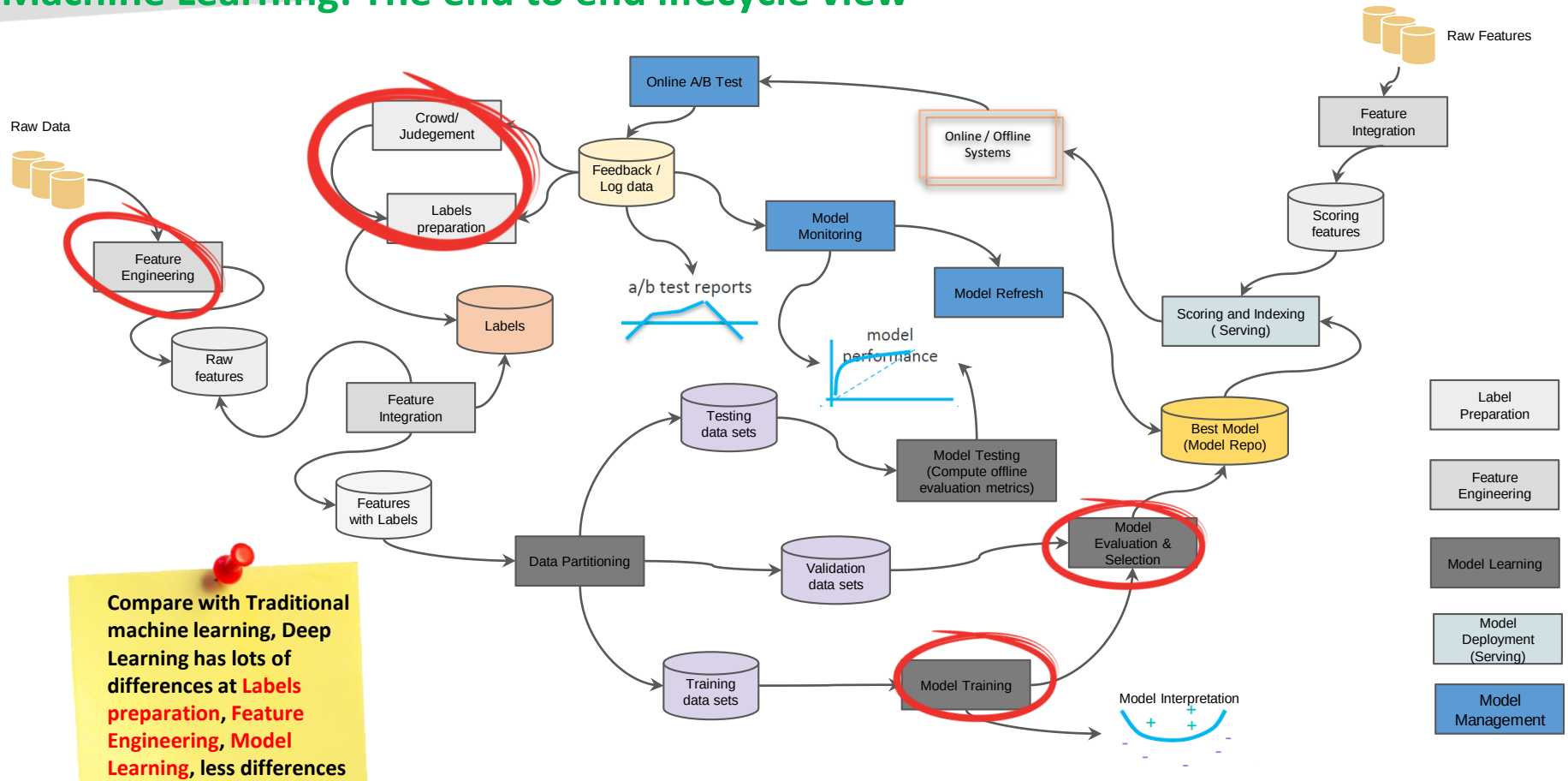


Machine Learning: The General Process



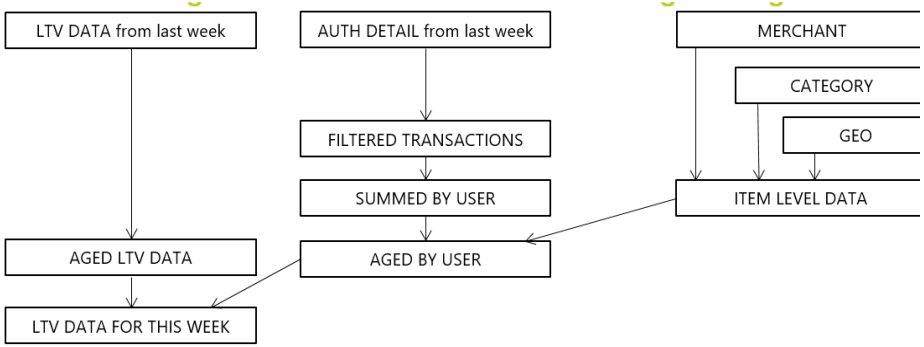
No matter the traditional ML and DL , no difference at general process

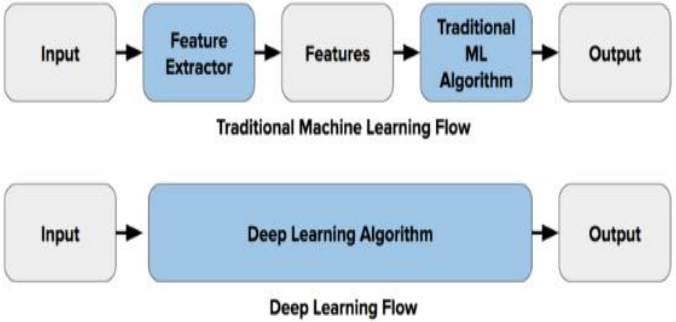
Machine Learning: The end to end lifecycle view



Compare with Traditional machine learning, Deep Learning has lots of differences at Labels preparation, Feature Engineering, Model Learning, less differences at other processes

How deep learning can help -- Feature engineering bottlenecks





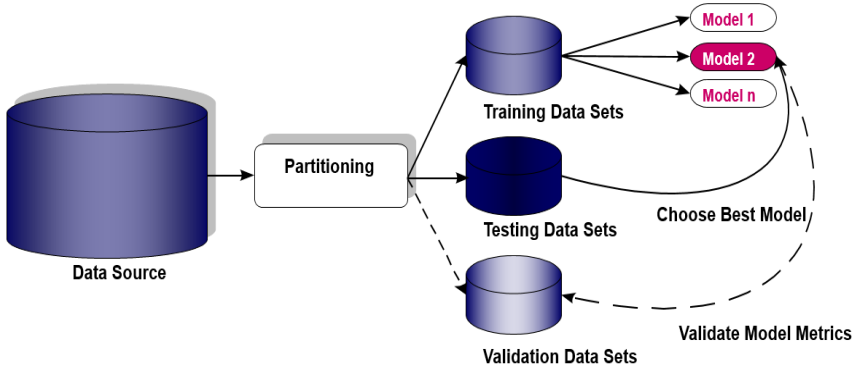
Bottlenecks

- Need to pre-calculate hundreds or thousands Long Term Variables for each user, such as total spends /visits for merchants list, category list divided by week, months and years
- The computation time for LTV features took > 70% of the data processing time for the whole lifecycle and occupied lots of resources which had huge impact to other critical workloads.
- Miss the feature selection optimizations which could save the data engineering efforts a lot

Improvements

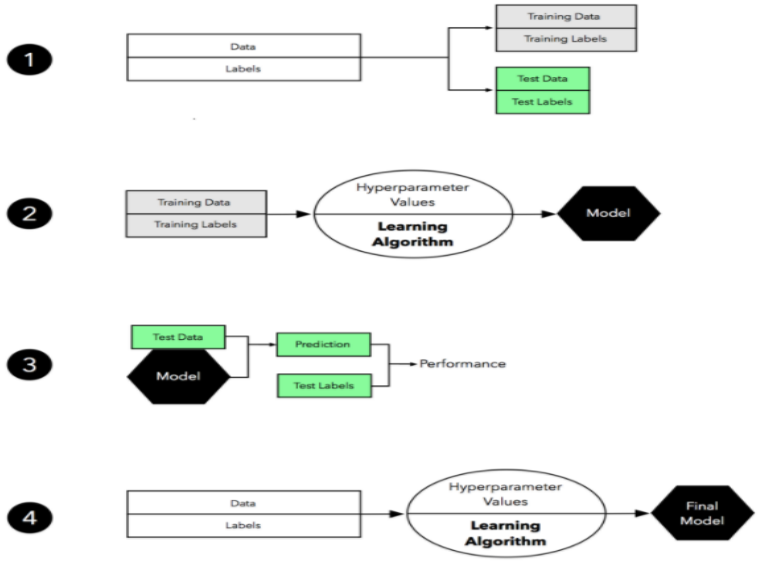
- *When build model , only focus on few pre-defined sliding features and custom overlap features (Users only need to identify the columns names from data source)*
- *Remove most of the LTV pre-calculations works, saved hours time and lots of resources*
- *Deep learning algorithm generates exponential growth of hidden embedding features ,do the internal features selections and optimization automatically when it does cross validation at training stage*

How deep learning can help -- Heavily relies on human machine learning experts



Relies on human to perform the following tasks:

- Select and construct appropriate features.
- Select an appropriate model family.
- Optimize model hyper parameters.
- Post process machine learning models.
- Critically analyze the results obtained.

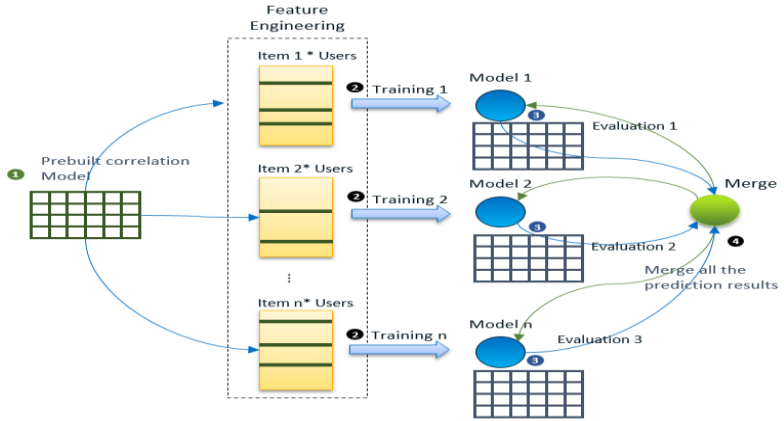


Improvements

- *Common neural network "tricks", including initialization, L2 and dropout regularization, Batch normalization, gradient checking*
- *A variety of optimization algorithms, such as mini-batch gradient descent, Momentum, RMSprop and Adam*
- *Provides optimization-as-a-service using an ensemble of optimization strategies, allowing practitioners to efficiently optimize models faster and cheaper than standard approaches.*

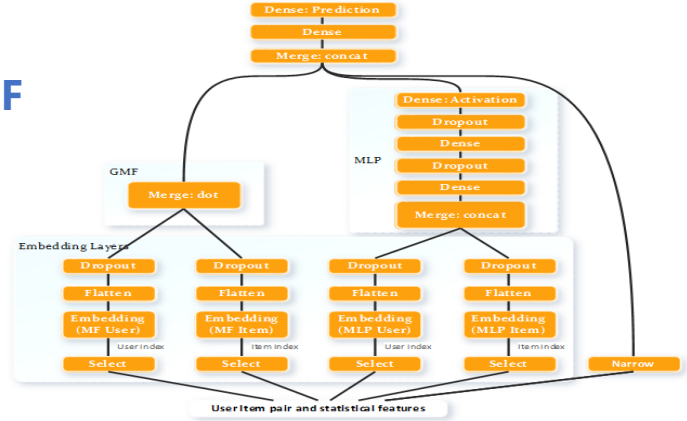
How deep learning can help -- Model scalability

LR

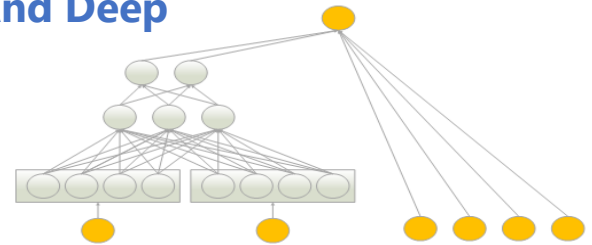


- All the pipelines separated by items and generate one model for each item
- Have to pre-calculate the correlation matrix between items
- Lots of redundant duplications and computations at feature engineering ,training and testing process
- Run items in parallel and occupied most of cluster resources when executed
- Bad metrics for items with few transactions
- It is very hard to scale more items , from hundreds to millions ?

NCF



Wide And Deep

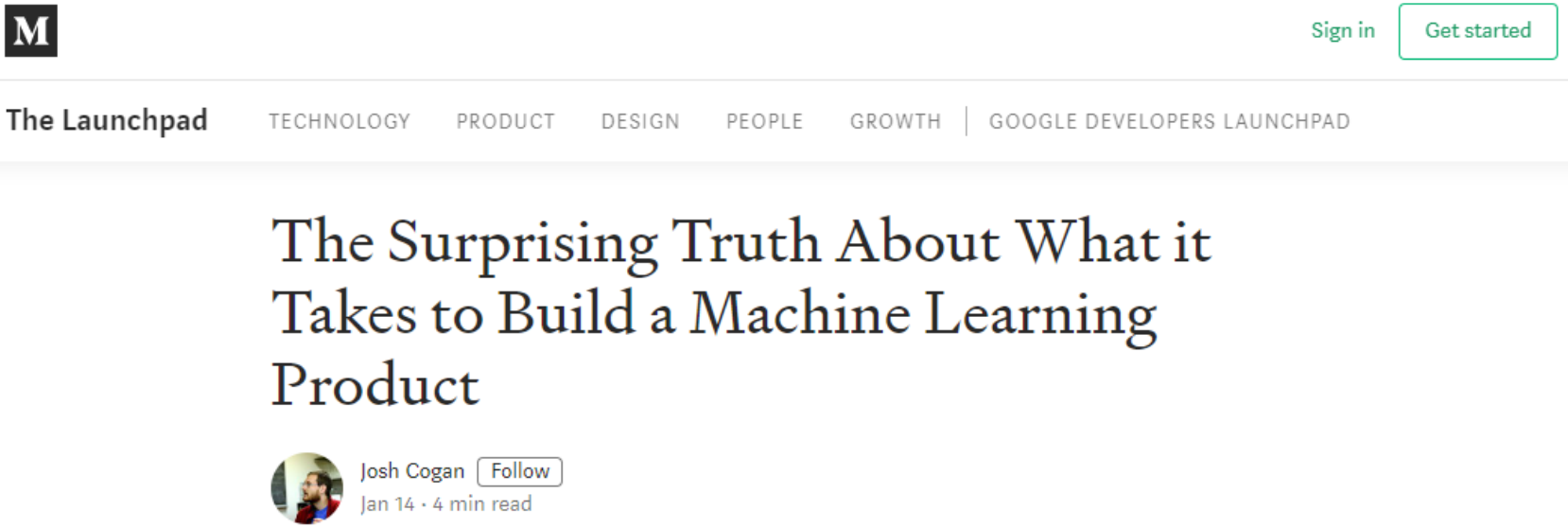


Improvements

- *Scale models in deeper and wider without decreasing metrics*

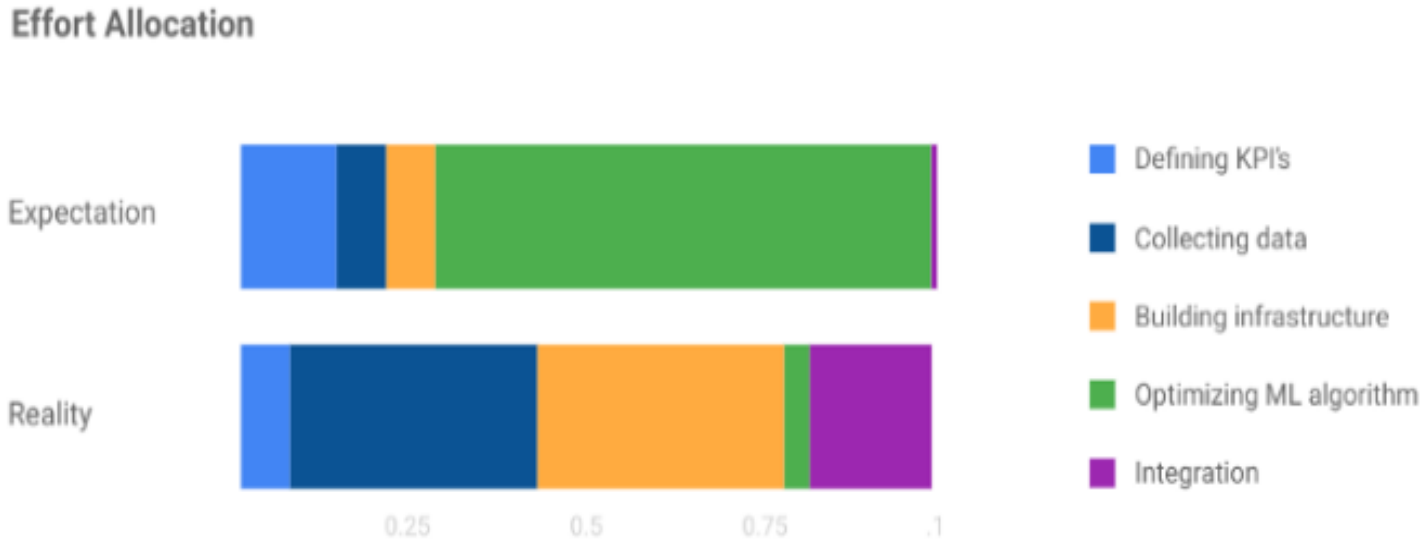
Ready to Production ?

You should know the Surprising Truth …



https://medium.com/thelaunchpad/the-ml-surprise-f54706361a6c

The Surprising Truth …



Hidden truths

"Hidden Technical Debt in Machine Learning Systems," Google NIPS 2015

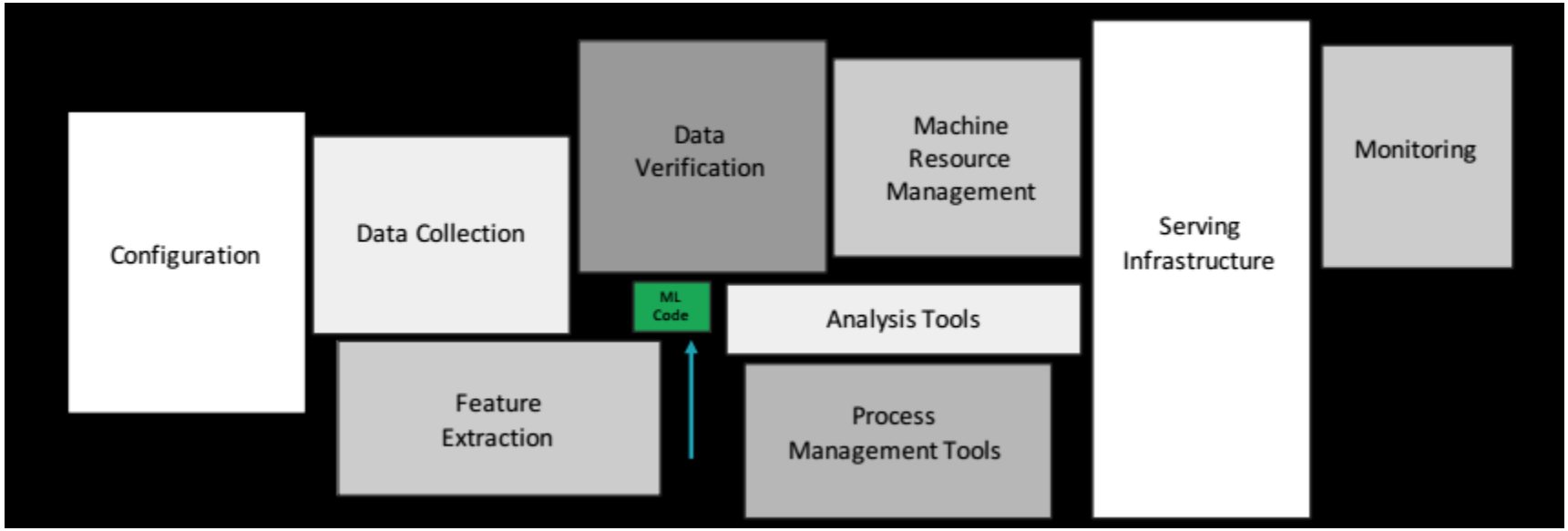
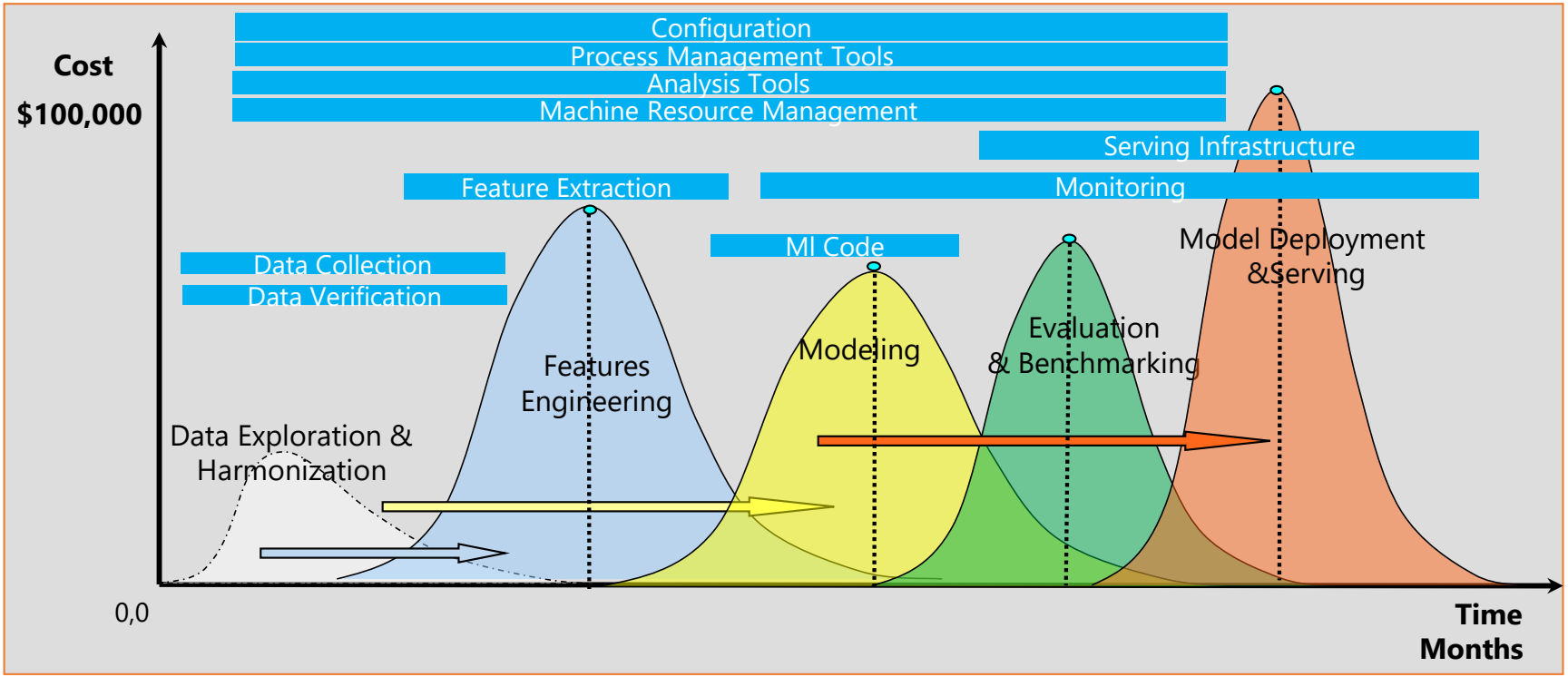


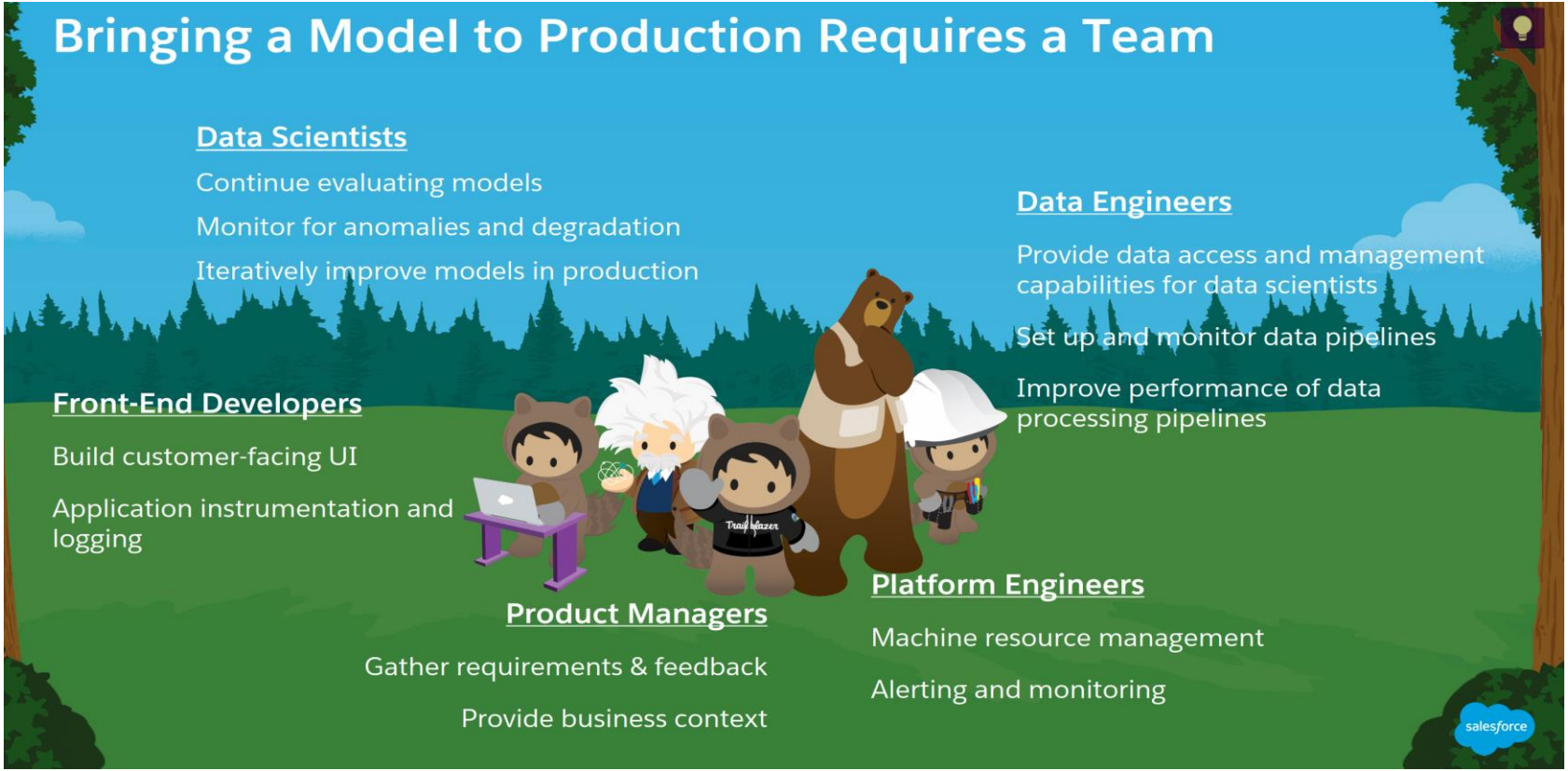
Figure 1: Only a small fraction of real-world ML systems is composed of the ML code. The required surrounding infrastructure is vast and complex. These include boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies, configuration issues, changes in the external world, and a variety of system-level anti-patterns

Hidden truths for AI -- continue

A Long and Expense Journey



Root of causes No.1

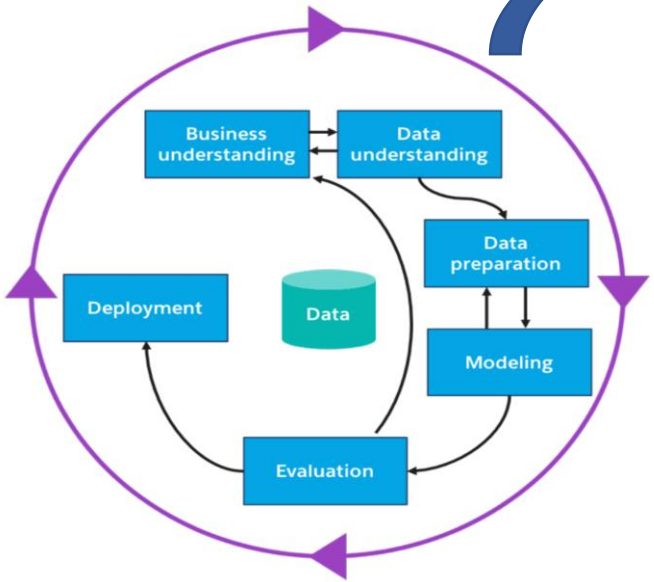


https://www.infoq.com/presentations/salesforce-ai-microservices/

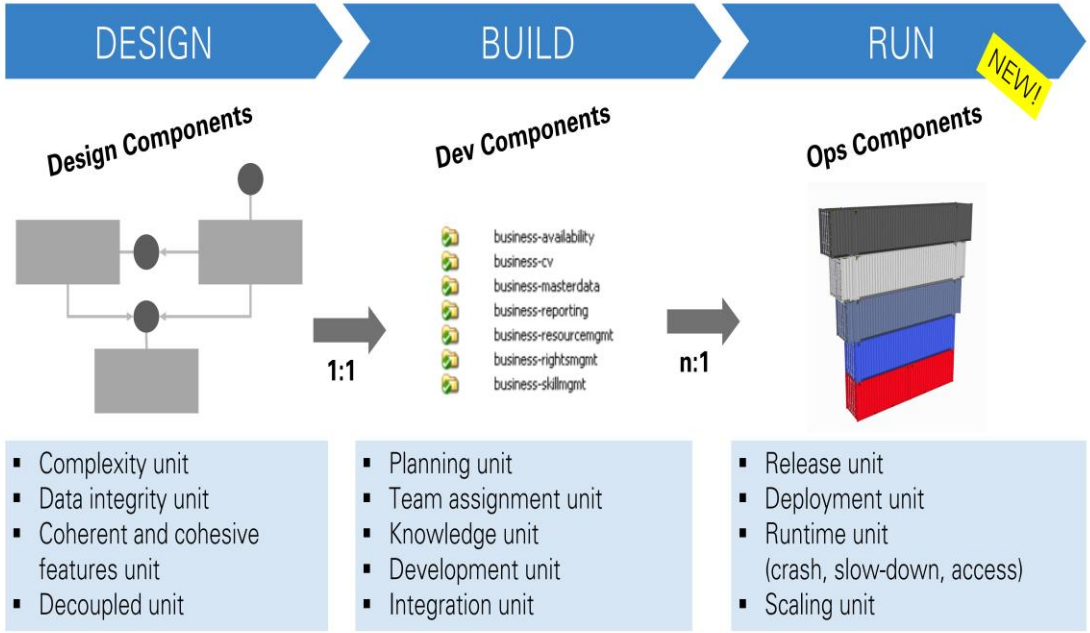
Root of causes No.2

GAP of Process to Production

Models All Along the ML Lifecycle



Components All Along the Software Lifecycle



Root of causes No.3

For large Enterprises which normally develop complex Softwares, Microservice Architecture is the only way to tackle complexity and to be competitive in the market

Monolith

Modular Monolith

Microservices

image above, Monolithic application is one single unit (tightly coupled) like a single Cube. Modular application is like Rubric Cube which can contain small modules but the modules cannot be separated and can only be deployed together.
Microservices are like a lego cube made by lego blocks where the blocks are loosely coupled, easily separable and all the lego blocks together made the cube.

Application Scaling

Development Speed

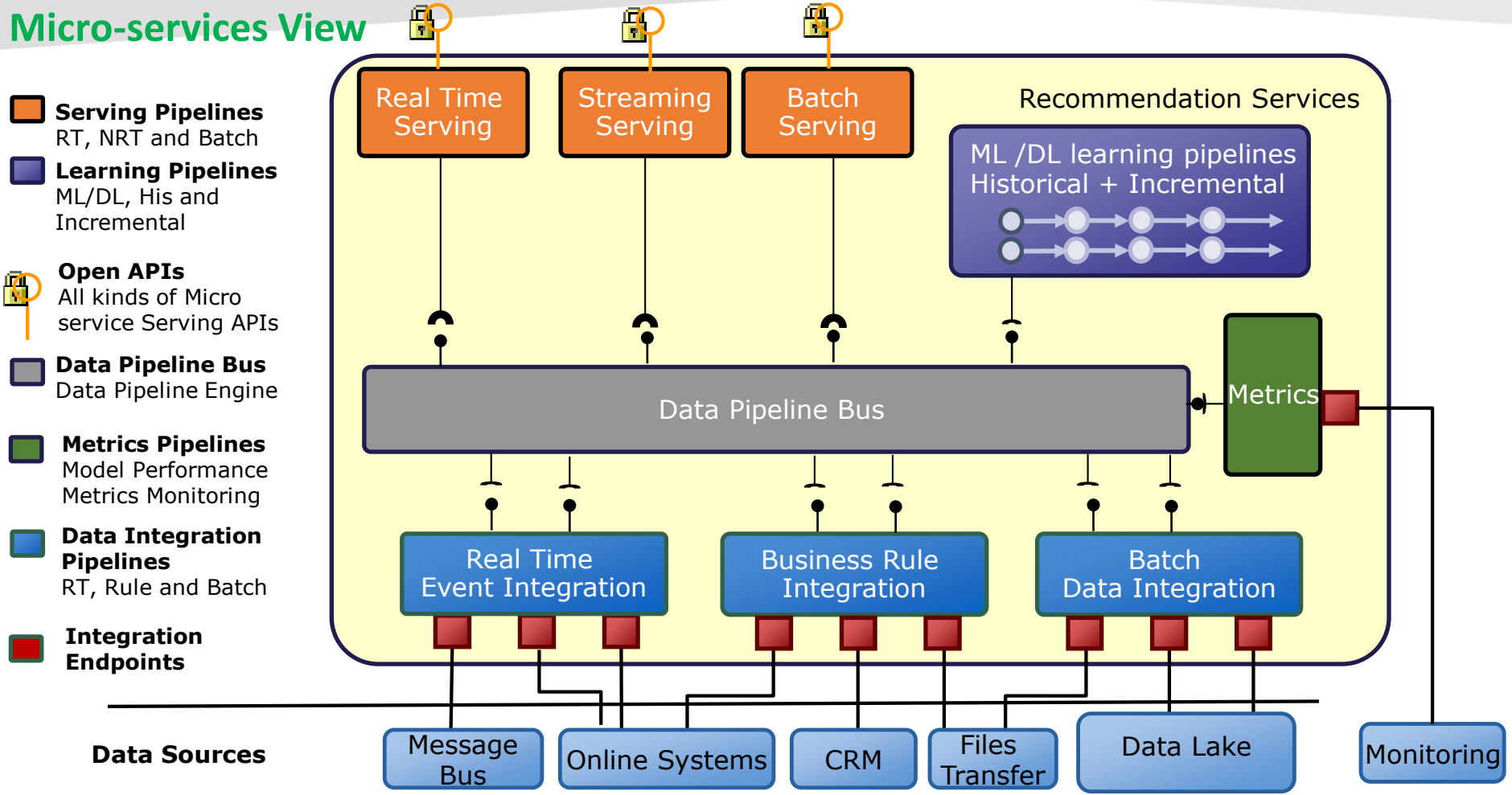
Modularization

Development Scaling

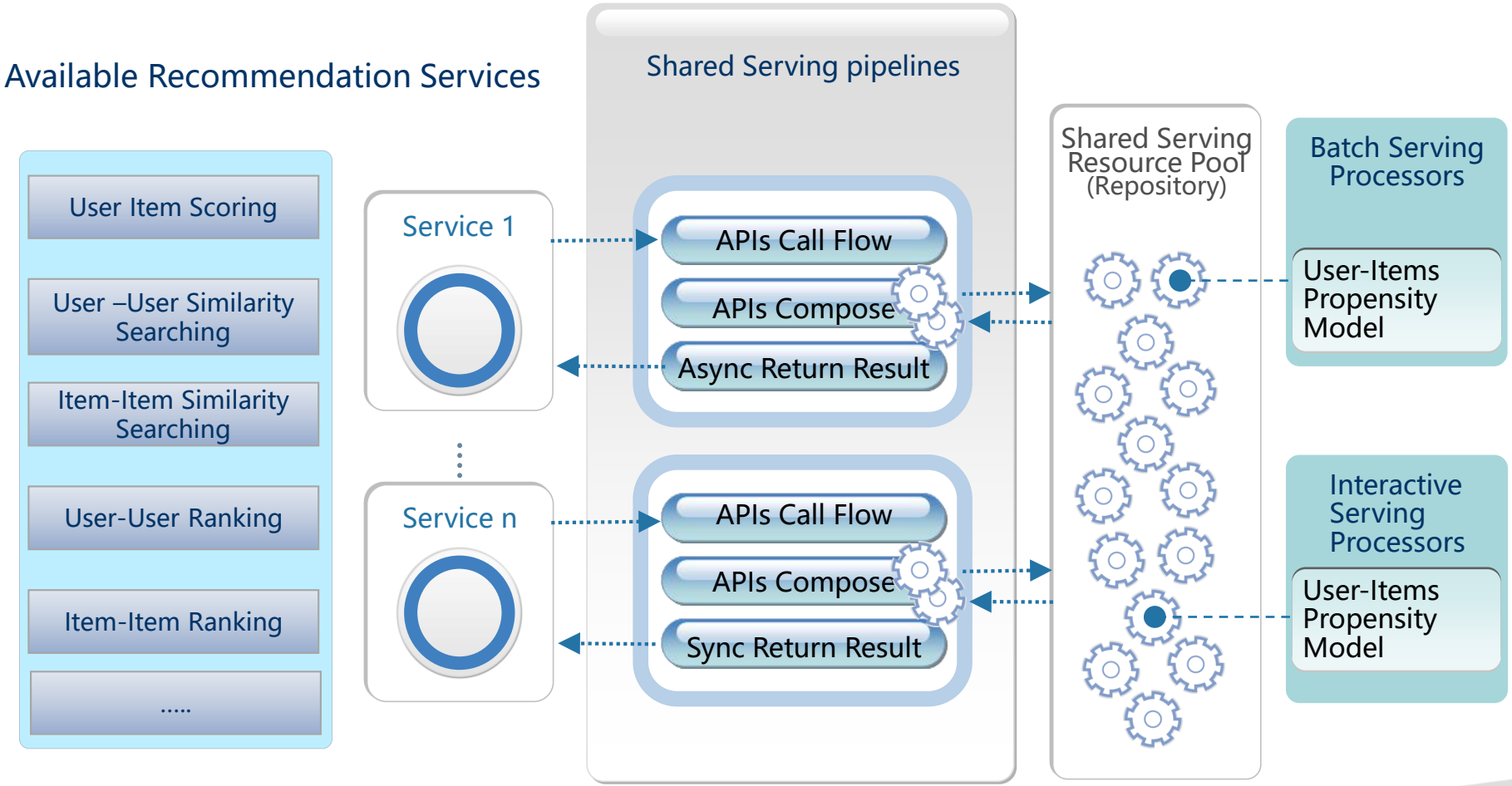
Release Cycle

Modernization

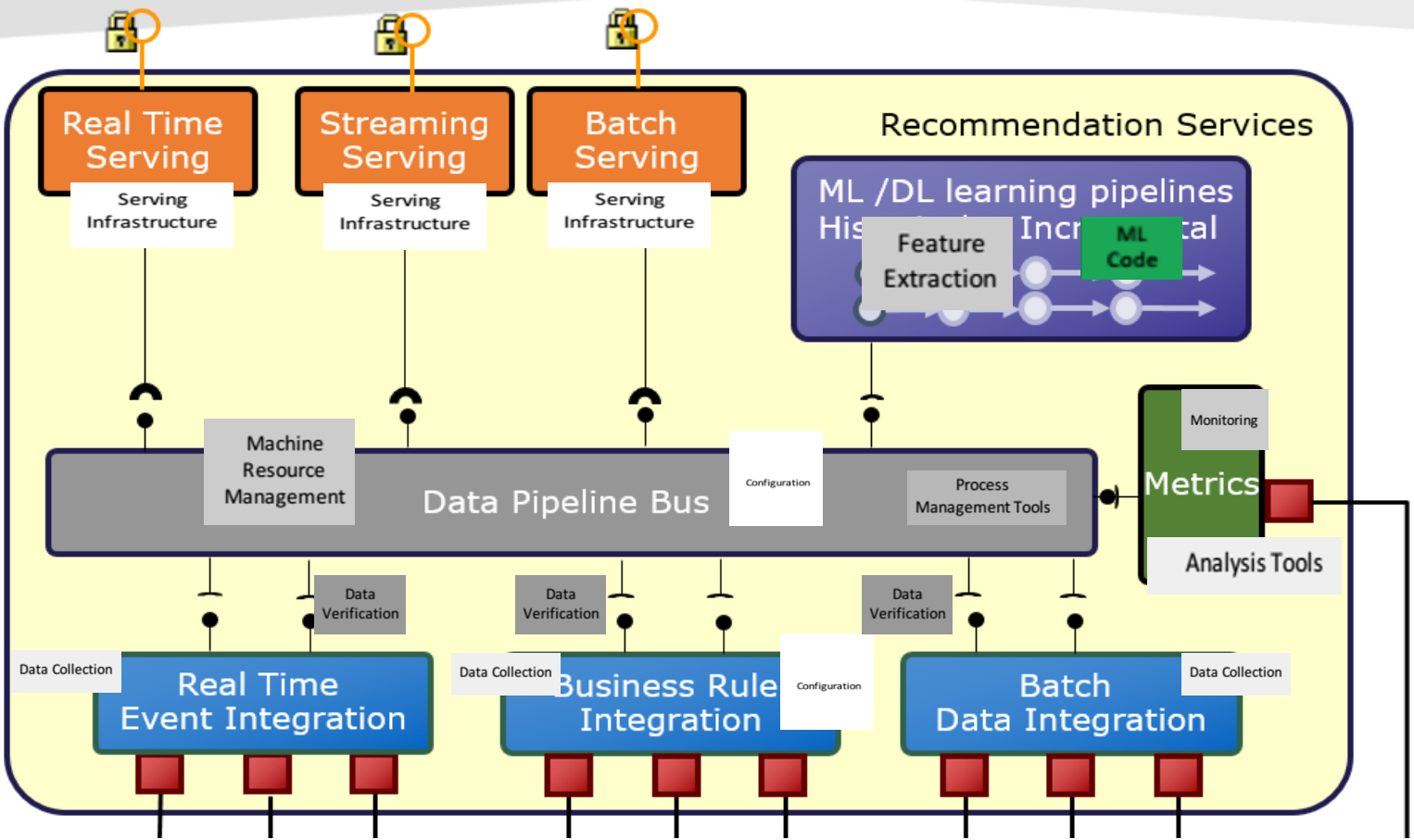
Micro-services View



Glance of model serving



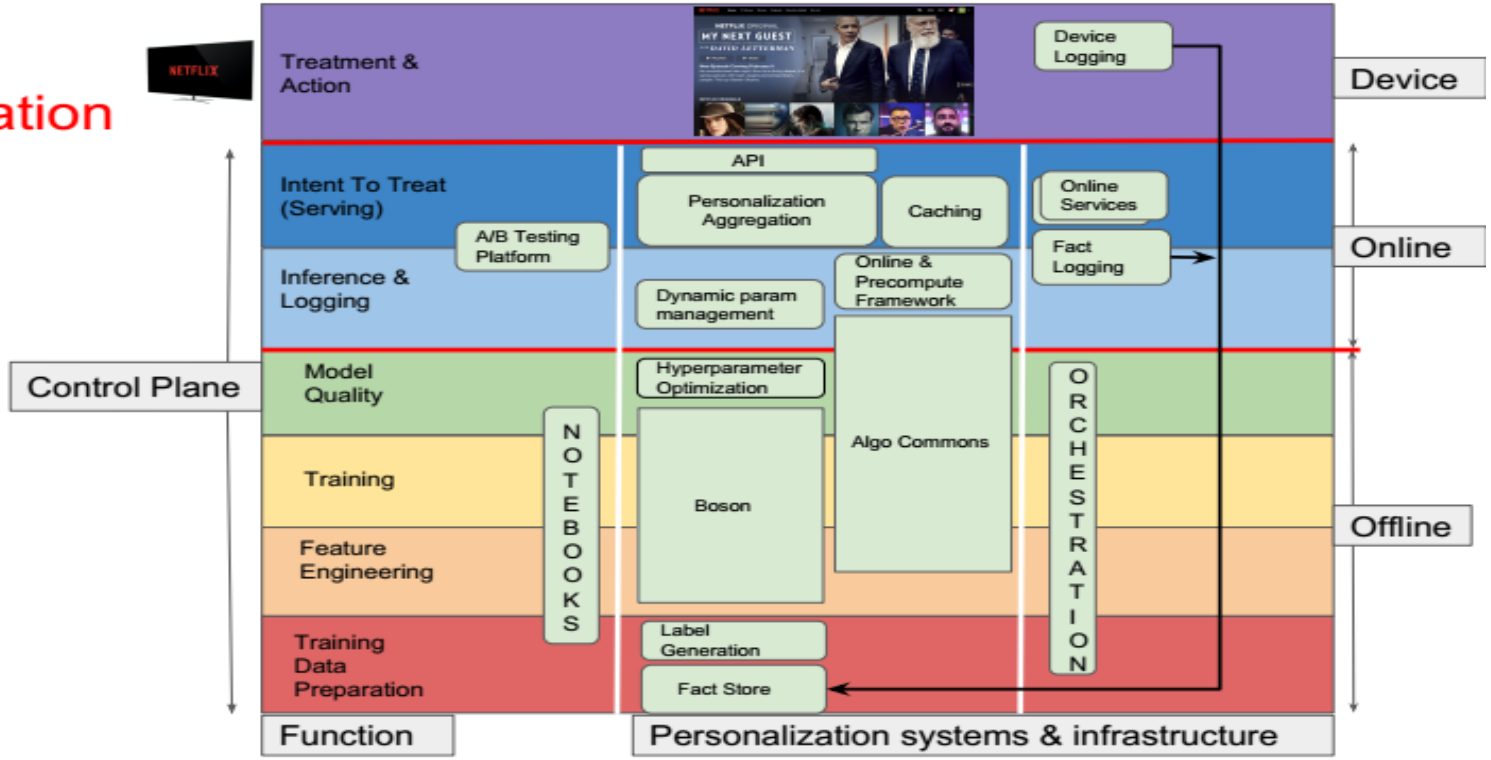
Where the debts been covered ?



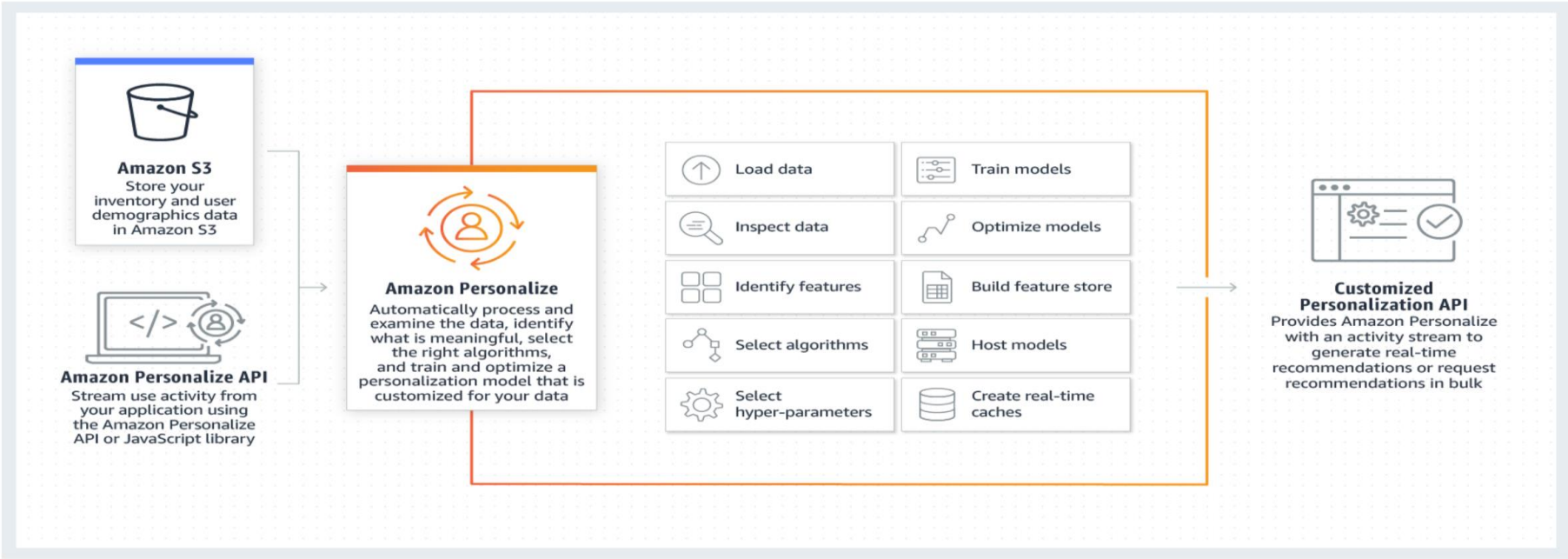
What we can learn from Netflix ?

https://www.slideshare.net/FaisalZakariaSiddiqi/ml-infra-for-netflix-recommendations-ai-nextcon-talk

The Personalization Rainbow

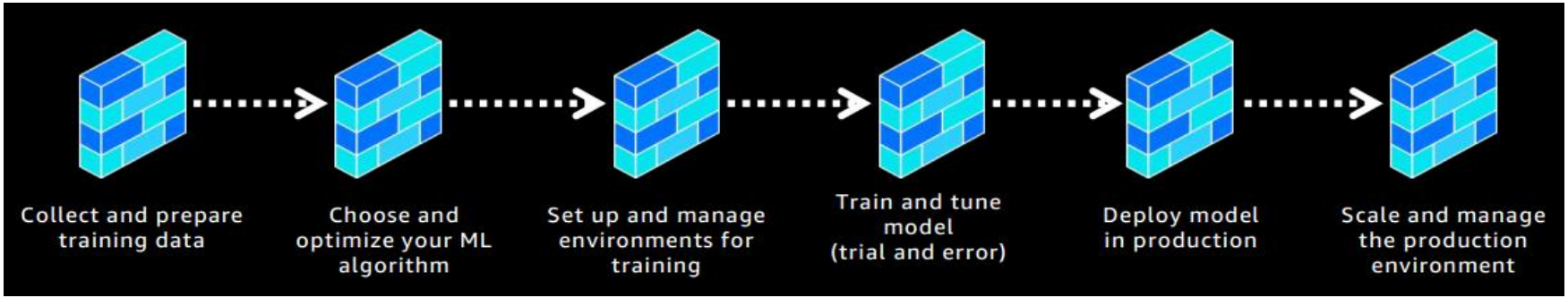


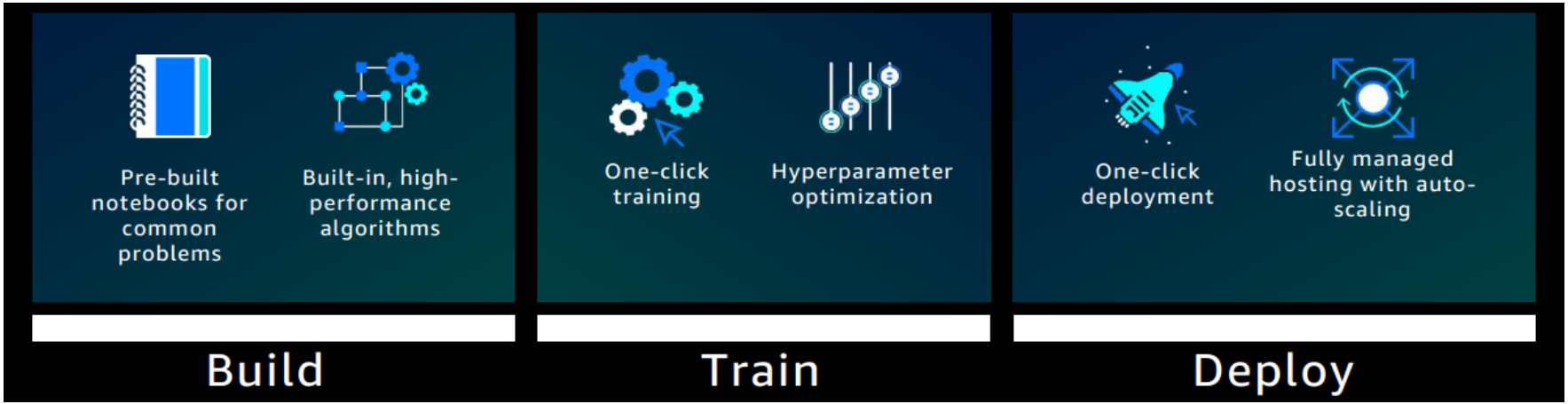
What we can learn from Amazon ? -- Personalize



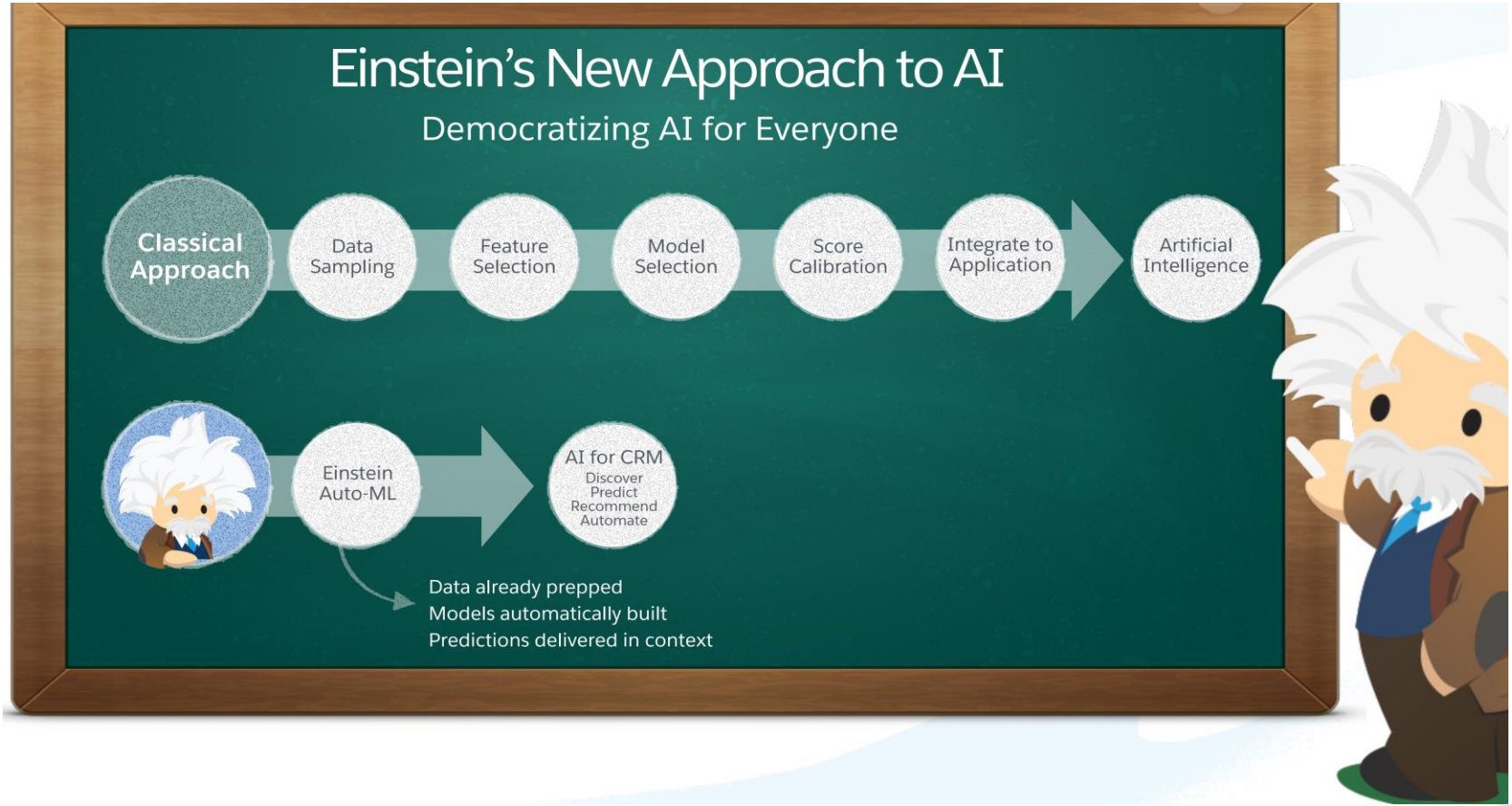
https://docs.aws.amazon.com/personalize/latest/dg/getting-recommendations.html#recommendations

What we can learn from Amazon ? -- SageMaker





What we can learn from Salesforce.com ? -- Einstein



What we can learn from Prediction IO ? -- Prediction IO

http://predictionio.apache.org/

Engine Template Gallery

Edit this page

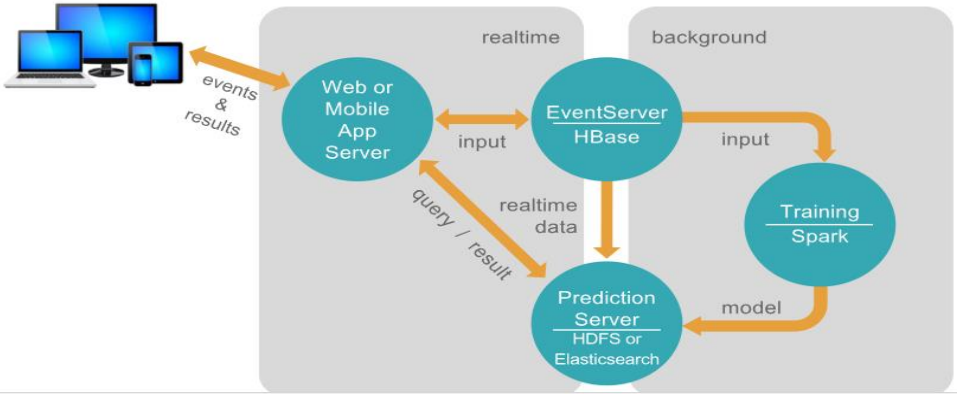
Pick a tab for the type of template you are looking for. Some still need to be ported (a simple process) to Apache PIO and these are marked. Also see each Template description for special support instructions.

Recommenders | Classification | Regression | NLP | Clustering | Similarity | Other

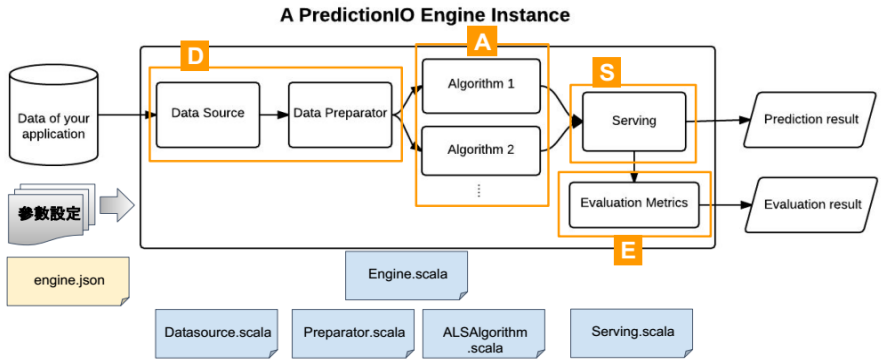
The Universal Recommender

Star 404

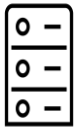
Use for:



Customizing your Engine with D-A-S-E



Additional challenges for adopting Deep Learning at Enterprise



Collocated with mass data storage

- Analyze a large amount of data on the same Big Data clusters where the data are stored (HDFS, HBase, Hive, etc.) rather than ***move or duplicate data***

Data governance with restricted Processing

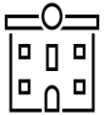
- Follow data privacy, regulation and compliance (such as PCI/PII compliance and GDPR rather than ***operate data in unsecured zones***

Adopt structured sparse data sets

- More challenges also more benefits to adopt structured high dimensional sparse data sets rather than ***non-structured dense data sets***

Seamless integration with Products Internal & External

- Add deep learning capabilities to existing Analytic Applications and/or machine learning workflows rather than ***rebuild all of them***



Shared infrastructure with Multi-tenant isolated resources

- Leverage existing Big Data clusters and deep learning workloads should be managed and monitored with other workloads (ETL, data warehouse, traditional ML etc..) rather than ***run ML/DL workloads standalone in separate clusters***

Automation and easy to go

- End to end automation rather than ***manual efforts***
- Easy API enablement rather than ***big learning curve or depend on special experts***

Deep learning approaches evaluation journey



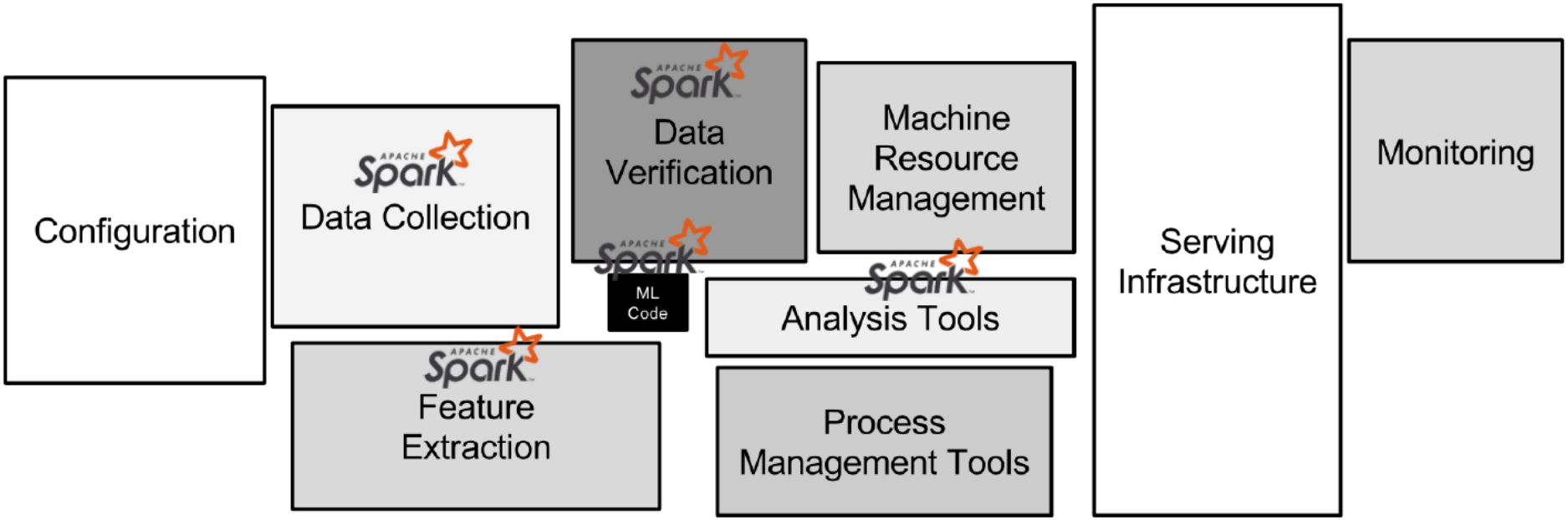




- Examples are good for dense high sample size data sets (But won't help us)
- Claimed that the GPU computing are better than CPU which requires new hardware infrastructure (very long timeline normally)
- Success requires many engineer-hours (Impossible to Install a Tensor Flow Cluster at STAGE ...)
- Low level APIs with steep learning curve (Where is your PhD degree ?)
- Not well integrated with other enterprise tools and need data movements (couldn't leverage the existing ETL, data warehousing and other analytic relevant data pipelines, technologies and tool sets. And it is also a big challenge to make duplicate data pipelines and data copy to the capacity and performance.)
- Tedious and fragile to distribute computations (less monitoring)
- The concerns of Enterprise Maturity and InfoSec (such as use GPU cluster with Tensor Flow from Google Cloud)

.....

Why Spark ?


Configuration
Data Collection
Data Verification
Machine Resource Management
Monitoring
Serving Infrastructure
ML Code
Analysis Tools
Feature Extraction
Process Management Tools

What does Spark offer for deep learning ?

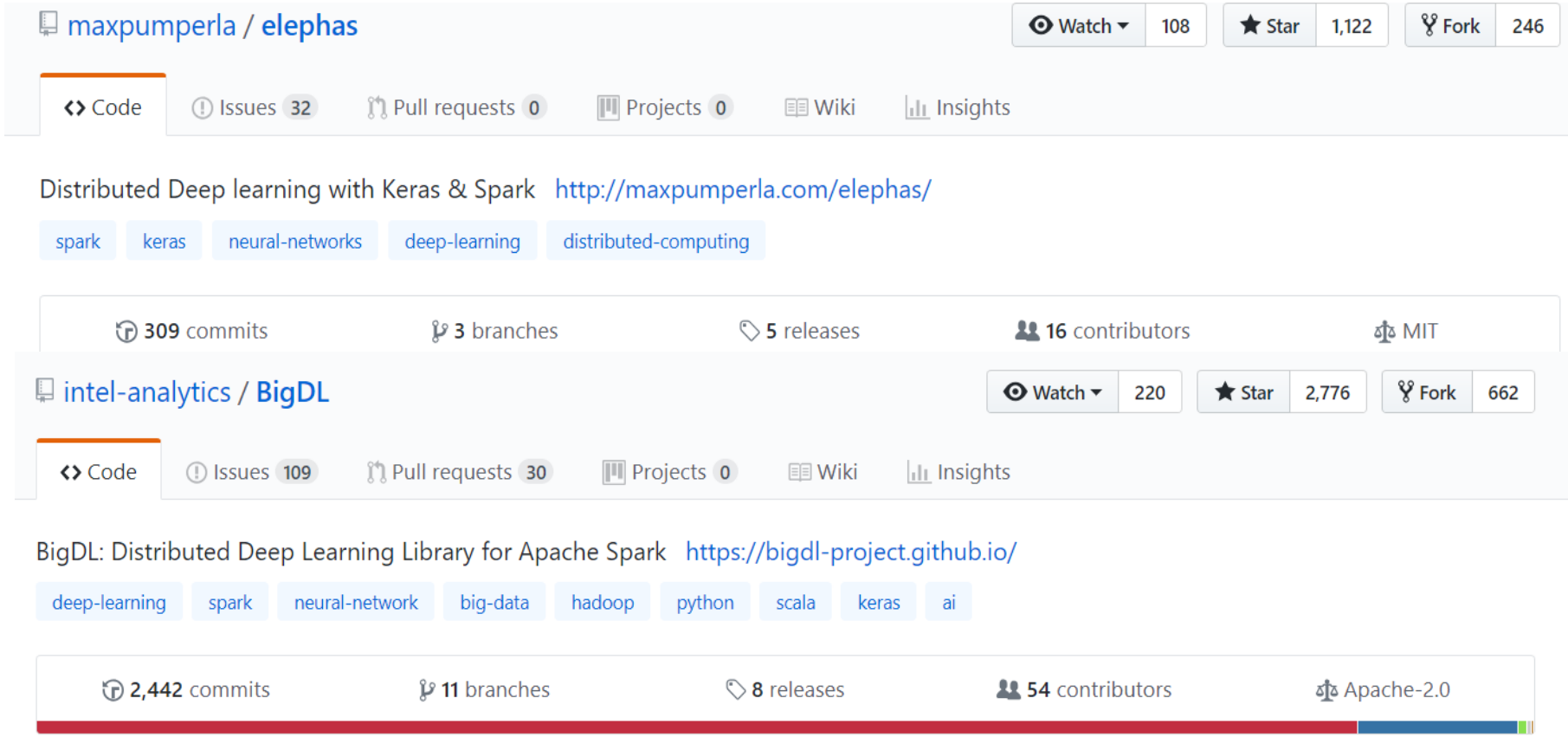
Integrations with existing DL libraries

- Deep Learning Pipelines (from Databricks)
- Caffe (CaffeOnSpark)
- Keras (Elephas)
- mxnet
- Paddle
- TensorFlow (TensorFlow on Spark, TensorFrames)
- CNTK (mmlspark)

Implementations of DL on Spark

- BigDL+ Analytic Zoo
- DeepDist
- DeepLearning4J
- SparkCL
- SparkNet

Options of Keras on Spark



Why BigDL ?

Lenet | Inception | Vgg | Resnet

Fast RCNN | SSD | RNN | LSTM

Deep Speech | Seq2Seq | Auto Encoder | Recommendation

Examples | Documents

Notebook | Tensor Board

Scala | Python

Layers

Spatial Convolution | Spatial MaxPooling | Volumetric Convolution | Volumetric MaxPooling | Fully Connected

ReLu | LRN | RNN | Batch Normalization | Other Layers

Optimization

SGD | Adagrad | LBFGS | Adamx

Adam | Adadelta | RMSprop

Criterion

Cross Entropy | ClassNLL | MSE

Dice Coefficient | Margin | Abs

Deep Learning Building Blocks
Layers, Optimizers, Criterion
Deep Learning Models

Scala* and Python* support
Spark ML Pipeline integration
Jupyter* notebook integration
Tensorboard* integration
OpenCV* support
Model Interoperability (Caffe*/TensorFlow*/Keras*)

CONSUMER

CALL CENTER ROUTING
IMAGE SIMILARITY SEARCH
SMART JOB SEARCH

HEALTH

ANALYSIS OF 3D MRI MODELS FOR KNEE DEGRADATION

FINANCE

FRAUD DETECTION
RECOMMENDATION
CUSTOMER/MERCHANT PROPENSITY

RETAIL

IMAGE FEATURE EXTRACTION

MANUFACTURING

STEEL SURFACE DEFECT DETECTION

SCIENTIFIC COMPUTING

WEATHER FORECASTING

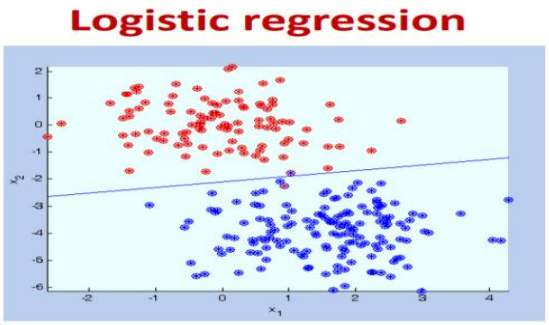
Why Analytics Zoo ?

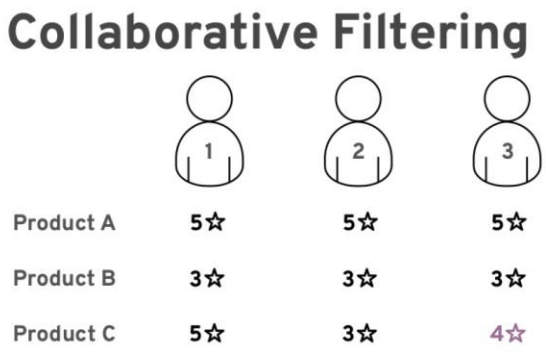
Analytics Zoo -> Unified Analytics + AI Platform for Spark and BigDL

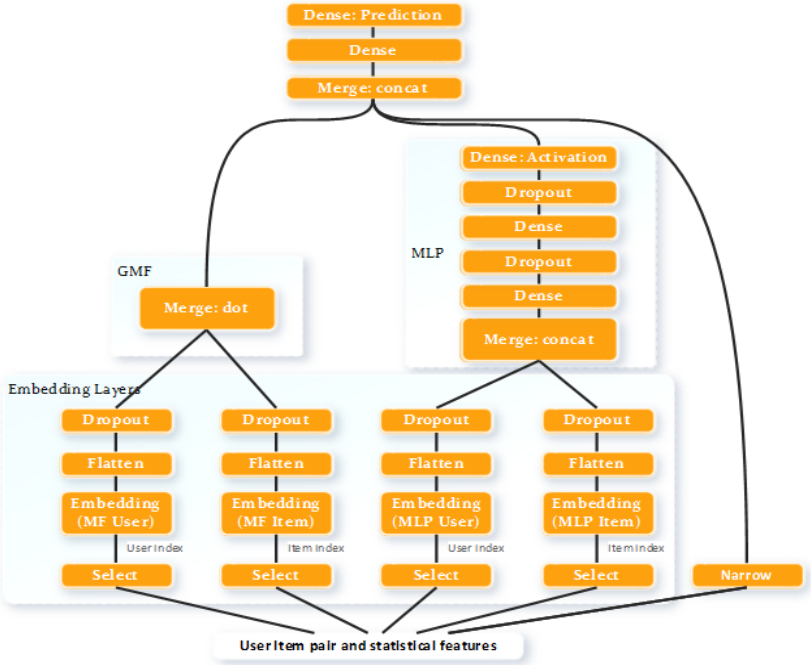
Reference Use Cases	Anomaly detection, sentiment analysis, fraud detection, chatbot, sequence prediction, etc.
Built-In Algorithms and Models	Image classification, object detection, text classification, recommendations, GAN, etc.
Feature Engineering and Transformations	Image, text, speech, 3D imaging, time series, etc.
High-Level Pipeline APIs	DataFrames, ML Pipelines, Autograd, Transfer Learning, etc.
Runtime Environment	Spark, BigDL, Python, etc.

https://github.com/intel-analytics/analytics-zoo

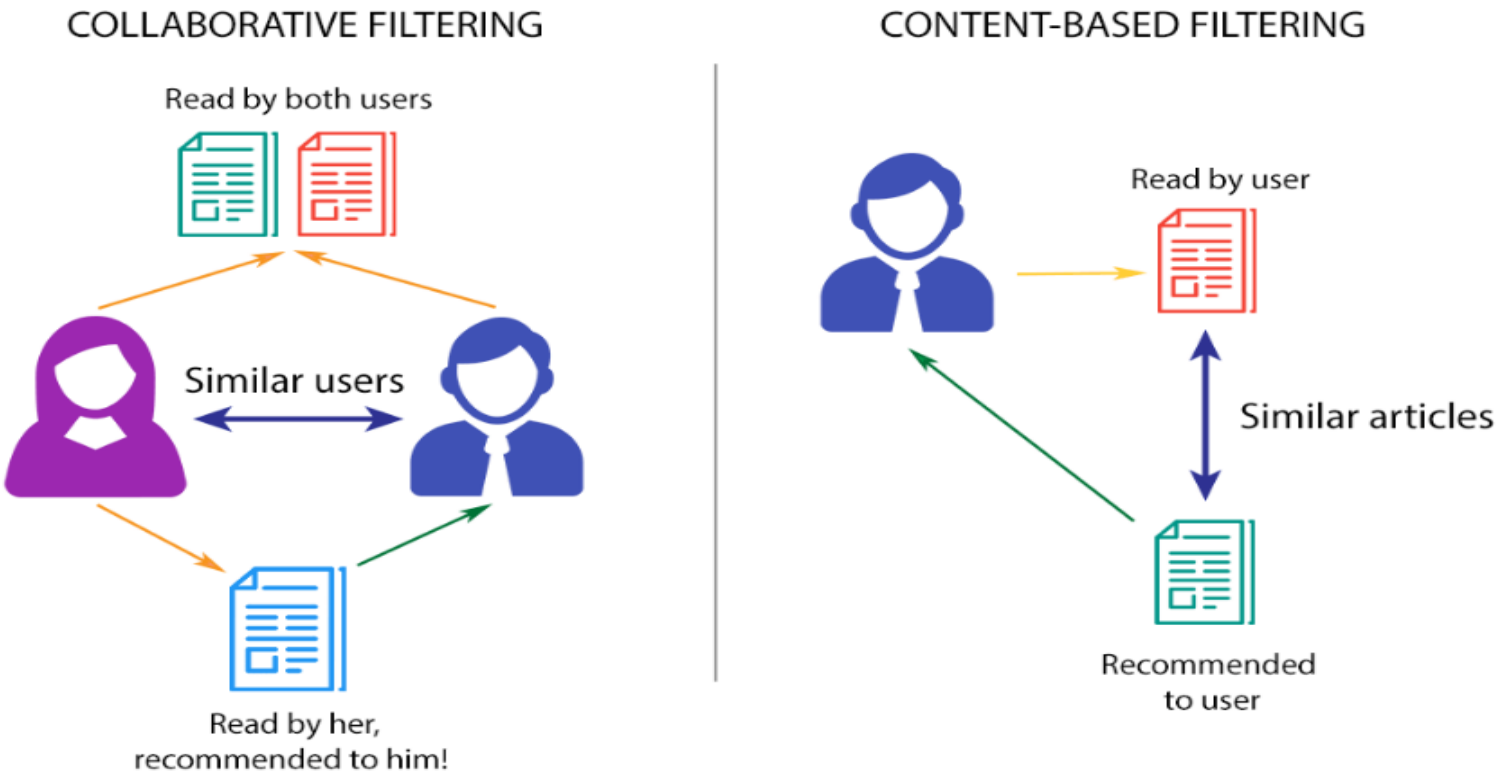
Code Lab : Benchmark between Spark Machine learning and Spark Deep learning with a user item propensity model example



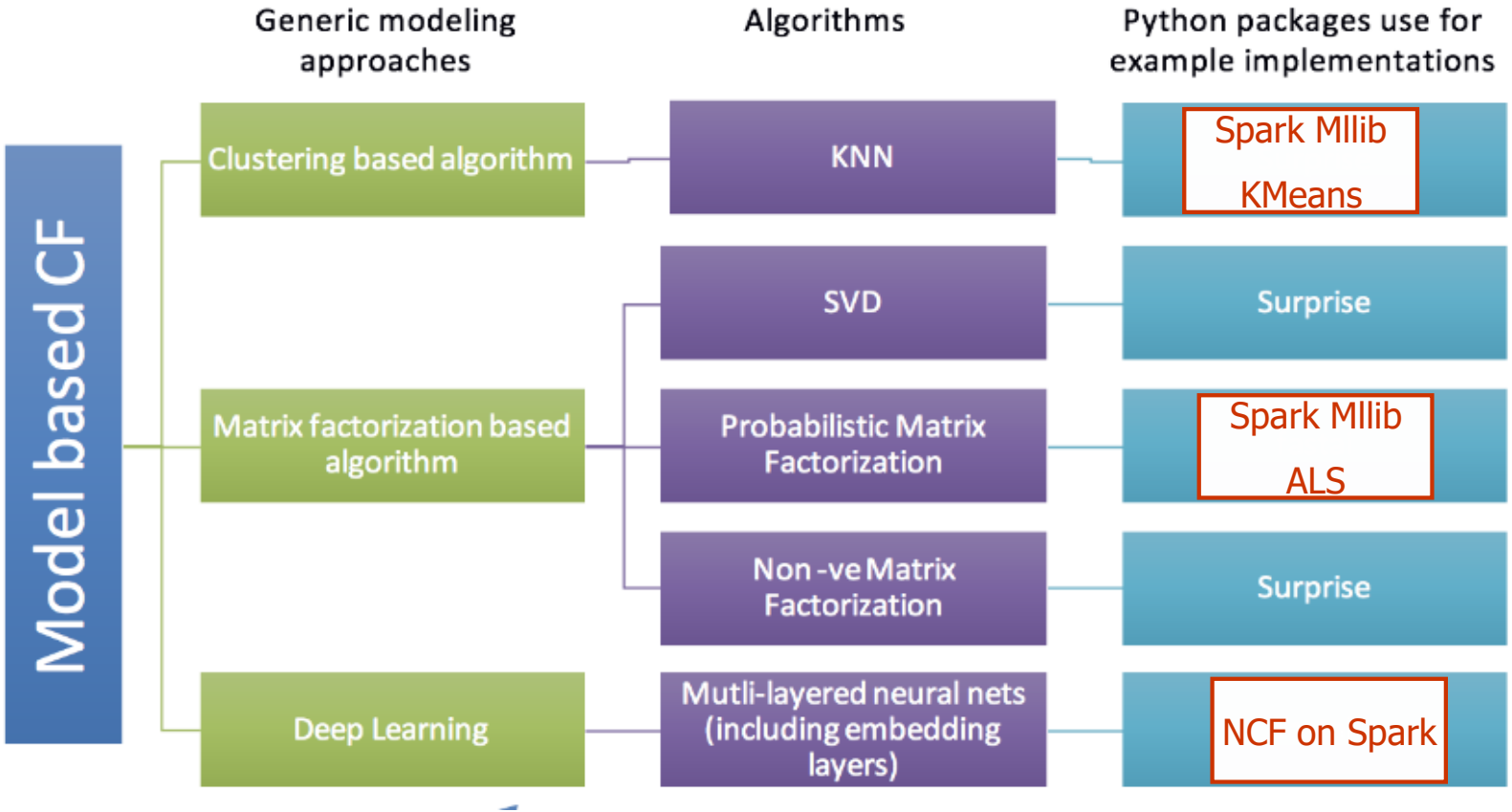




Collaborative Filtering -- concept



Collaborative Filtering -- Model selection



Spark Mllib

Enables Parallel, Distributed ML for large datasets on Spark Clusters

- Offers a set of parallelized machine learning algorithms for ML
- Supports Model Selection (hyper parameter tuning) using Cross Validation and Train-Validation Split.
- Supports Java, Scala or Python apps using Data Frame-based API

Spark Mllib Algorithms

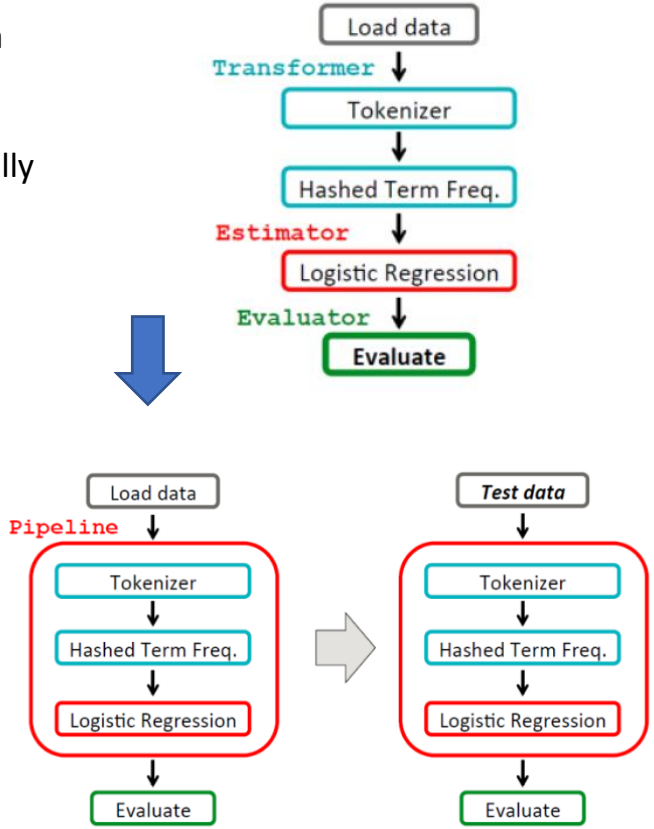
Spark MLlib Algorithms

Classification and Regression	• Linear Models (SVMs, logistic regression, linear regression) • Naïve Bayes • Decision Trees • Ensembles of trees (Random Forest, Gradient-Boosted Trees) • Isotonic regression
Clustering	• k-means and streaming k-means • Gaussian mixture • Power iteration clustering (PIC) • Latent Dirichlet allocation (LDA)
Collaborative Filtering	• Alternating least squares (ALS)
Dimensionality Reduction	• SVD • PCA
Frequent Pattern Mining	• FP-growth • Association rules
Basic Statistics	• Summary statistics • Correlations • Stratified sampling • Hypothesis testing • Random data generation

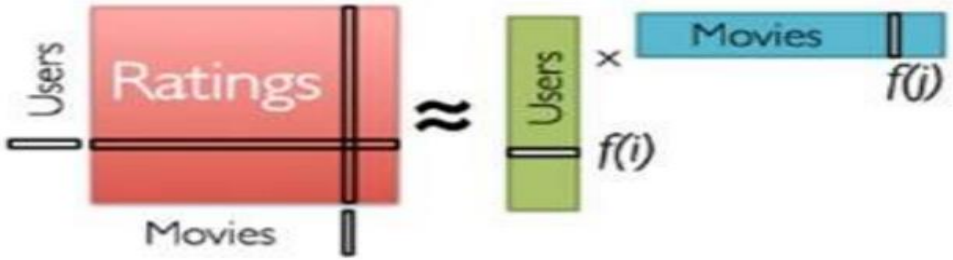
Spark Mllib ML Pipeline

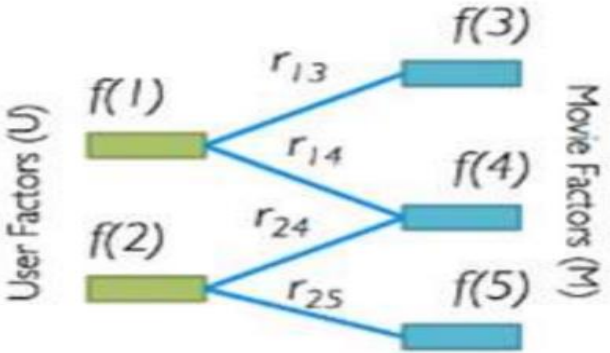
DataFrame: Spark ML uses DataFrame rather than regular RDD as they hold a variety of data types (e.g. feature vectors, true labels, and predictions).
Transformer: a transformer converts a DataFrame into another DataFrame usually by appending columns. (since Spark DataFrame is immutable, it actually creates a new DataFrame). The implement method for a transformer is "transform()".
Estimator: An Estimator is an algorithm which can be fit on a DataFrame to produce a Transformer. Implements method fit() taking a DataFrame and a model (also a transformer) as input.
Pipeline: Chains multiple Transformers and Estimators each as a stage to specify an ML workflow. These stages are run in order, and the input DataFrame is transformed as it passes through each stage.
Parameter: All Transformers and Estimators now share a common API for specifying parameters.
Evaluator: Evaluate model performance. The Evaluator can be

- RegressionEvaluator for regression problems,
- BinaryClassificationEvaluator for binary data, or
- MulticlassClassificationEvaluator for multiclass problems.

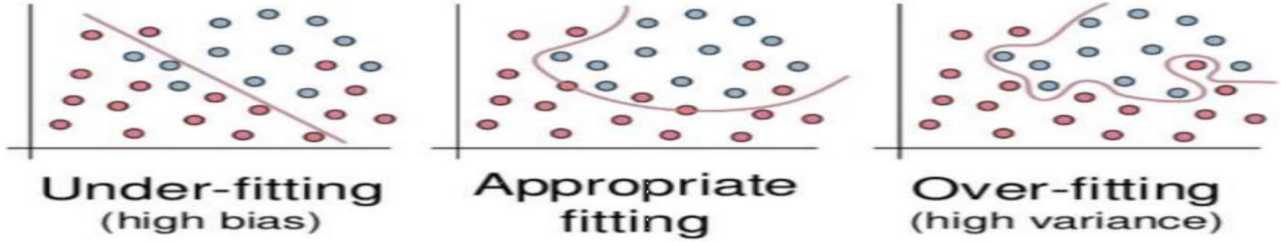


Alternating Least Squares (ALS) Spark ML





- Hyperparameters which can be adjusted:
  - **rank** = the number of latent factors in the model
  - **maxIter** = the maximum number of iterations
  - **regParam** = the regularization parameter



Neural Collaborative Filtering deep learning algorithm

https://arxiv.org/pdf/1708.05031.pdf
https://github.com/hexiangnan/neural_collaborative_filtering

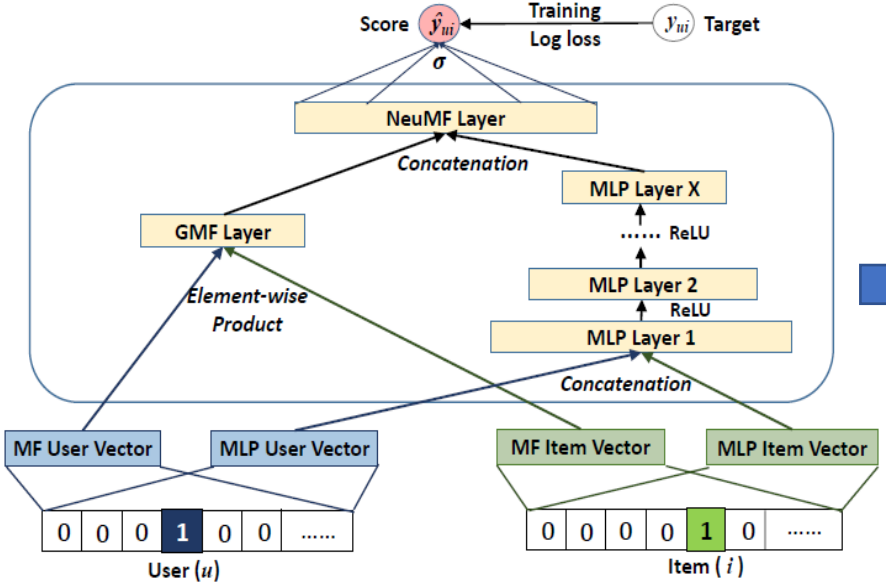
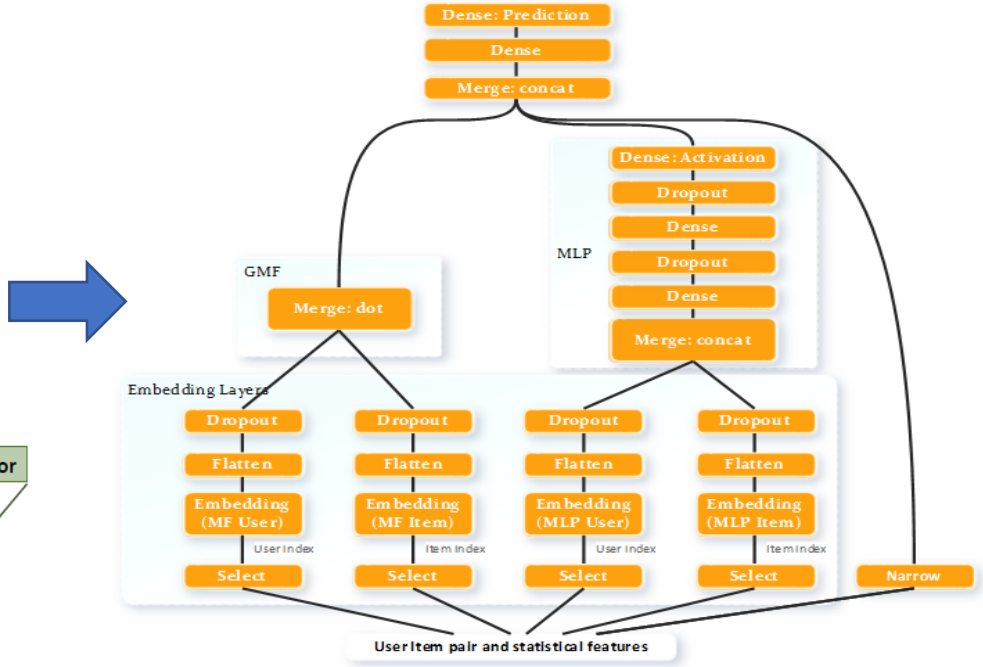
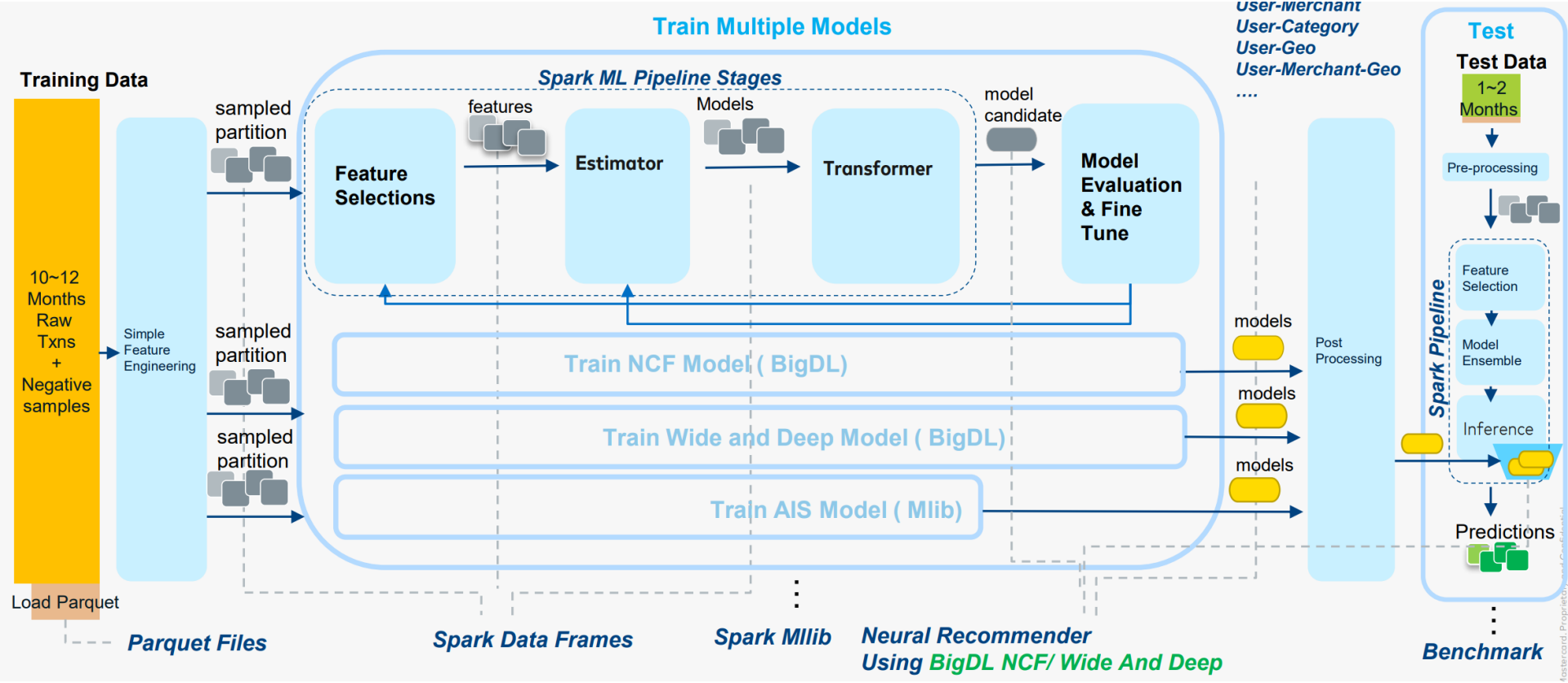


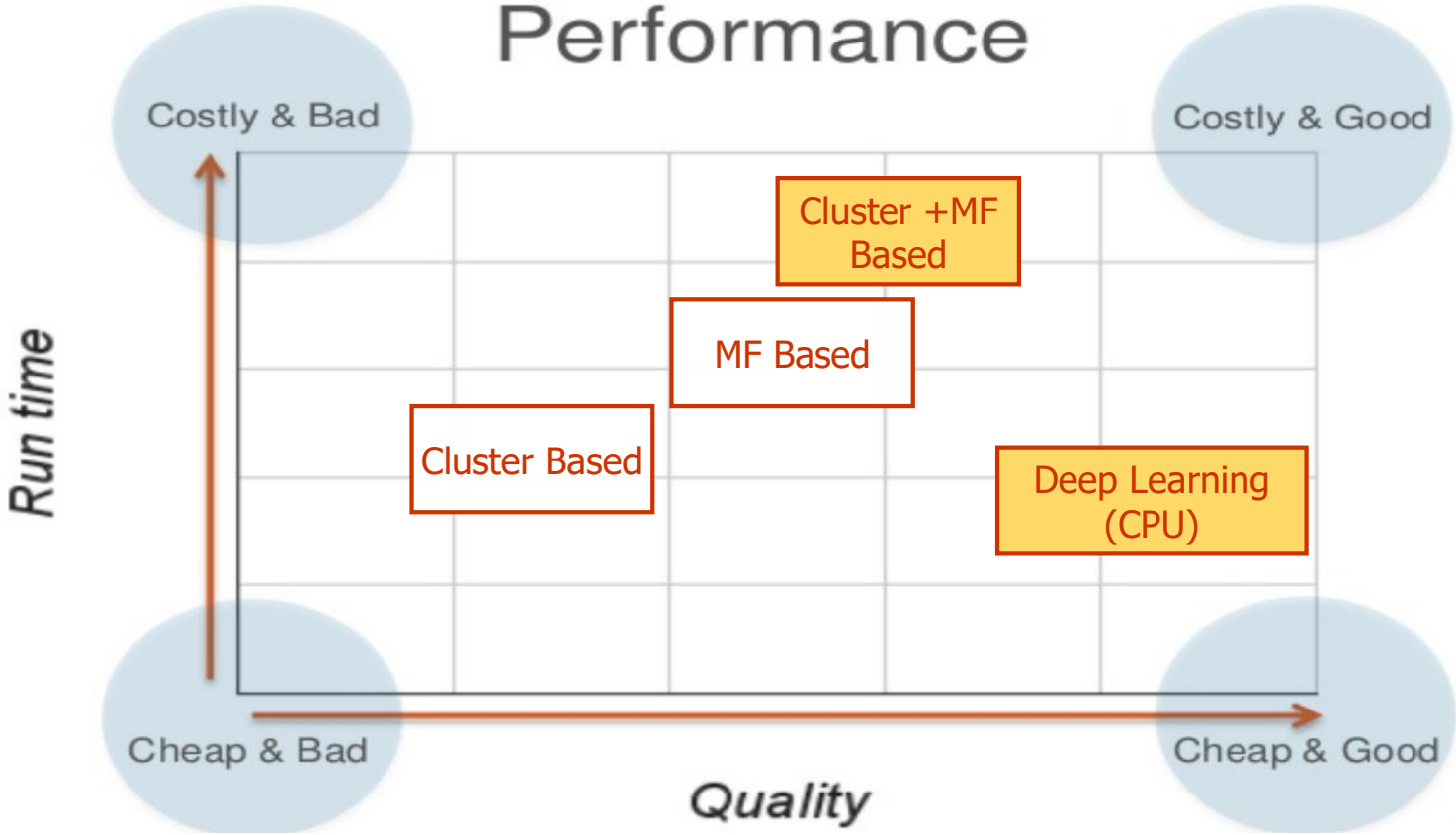
Figure 3: Neural matrix factorization model



Benchmark framework



Collaborative Filtering -- Trade Offs

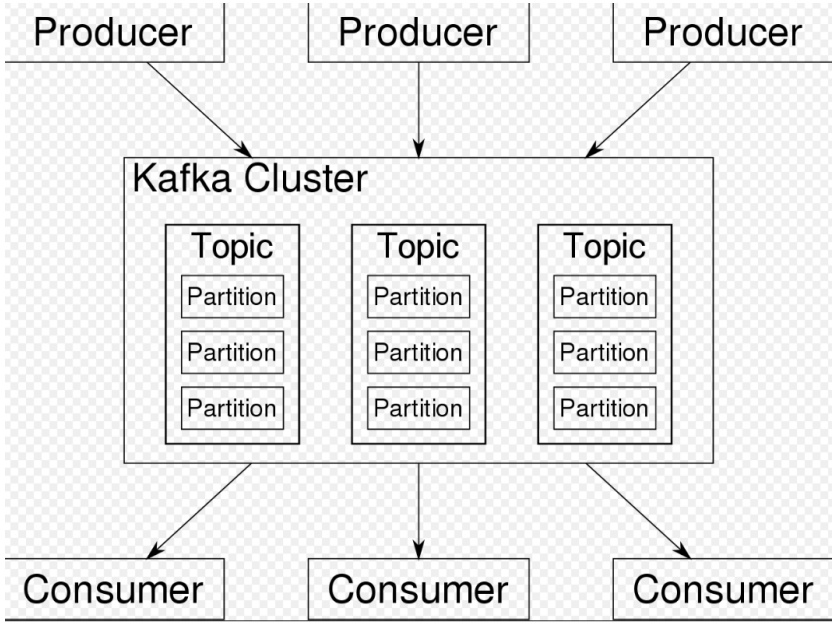


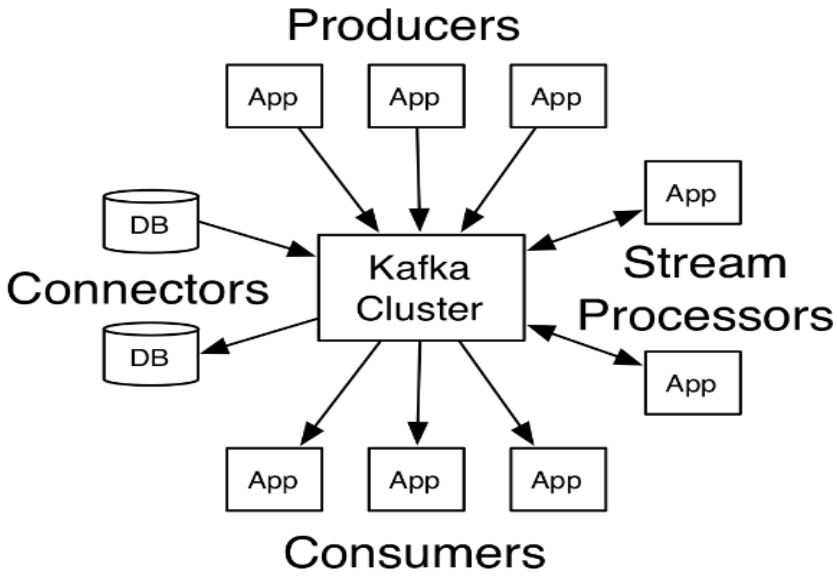
Show the code and run the benchmark

Let us break 10 mins

Live Demo (Build an end to end AI Pipeline with Kafka, NiFi, Spark Streaming and Keras on Spark)

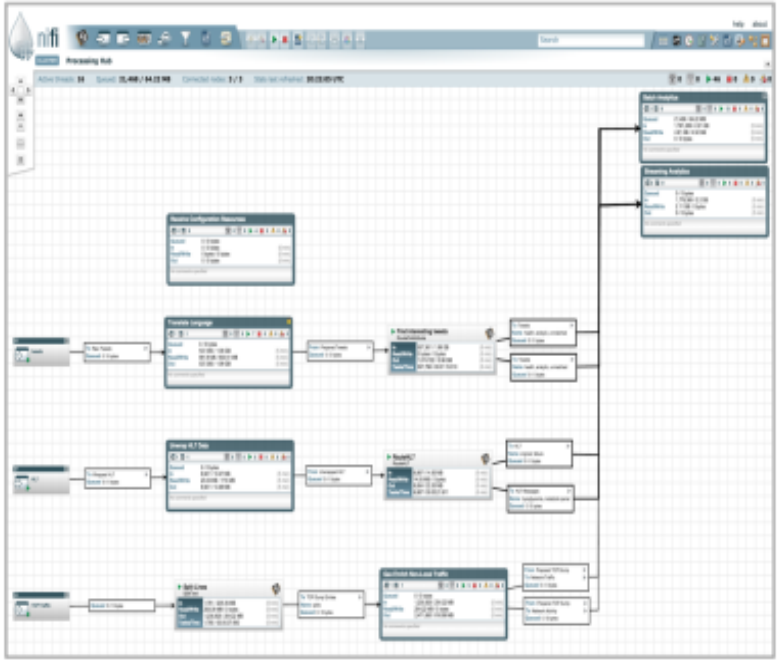
Kafka


Producer
Producer
Producer
Kafka Cluster
Topic
Partition
Partition
Partition
Topic
Partition
Partition
Partition
Topic
Partition
Partition
Partition
Consumer
Consumer
Consumer


Producers
App
App
App
App
DB
Connectors
Kafka
Cluster
Stream
Processors
DB
App
App
App
App
Consumers

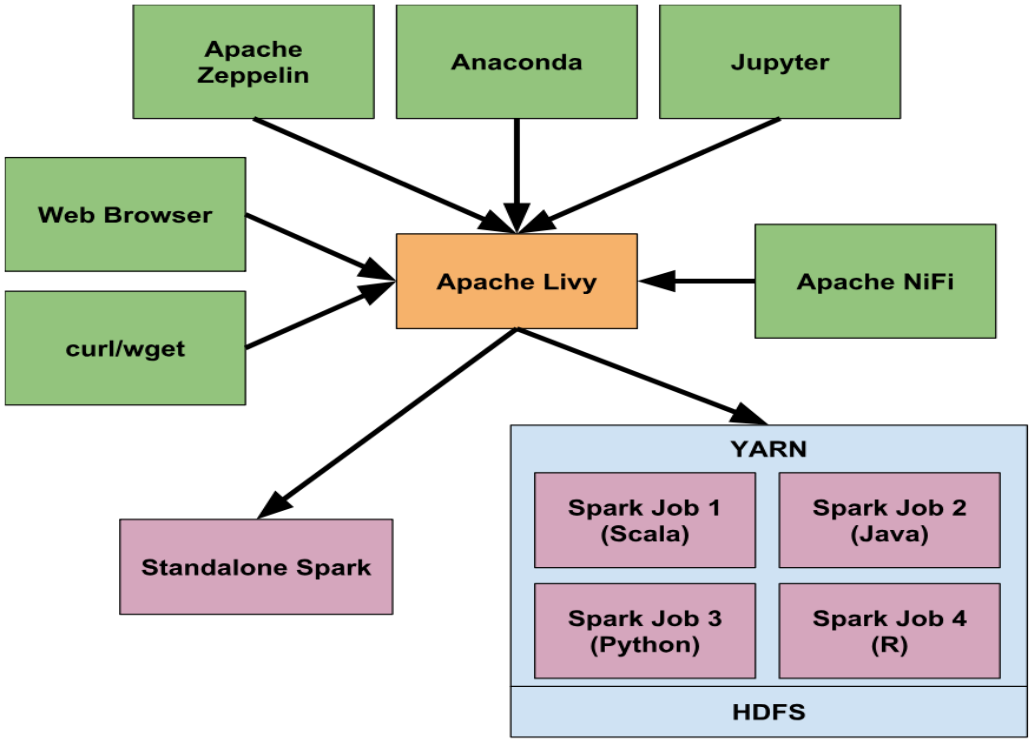
Apache NiFi

- Powerful and reliable **Distributed Real-time computation engine**
- Directed graphs of **data routing and transformation**
- Web-based **User Interface** for creating, monitoring, & controlling data flows
- Highly configurable - **modify data flow at runtime**, dynamically prioritize data
- **Data Provenance** tracks data through entire system
- **Easily extensible** through development of custom components

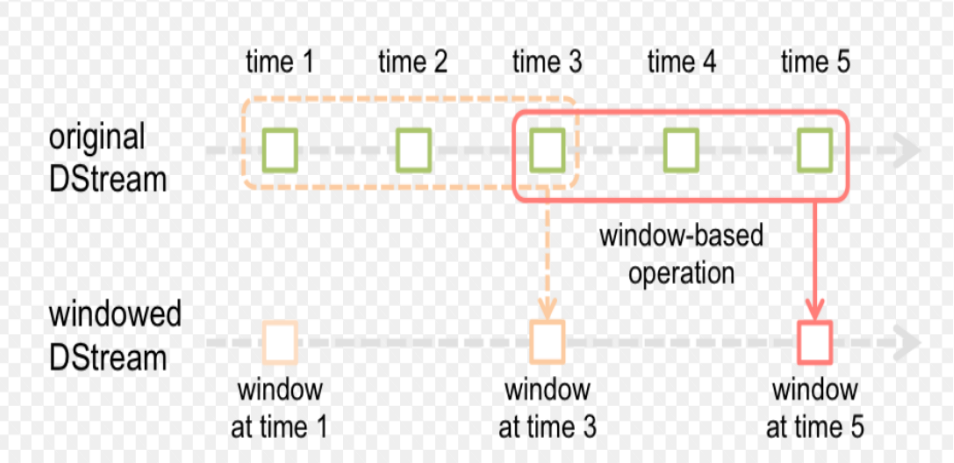


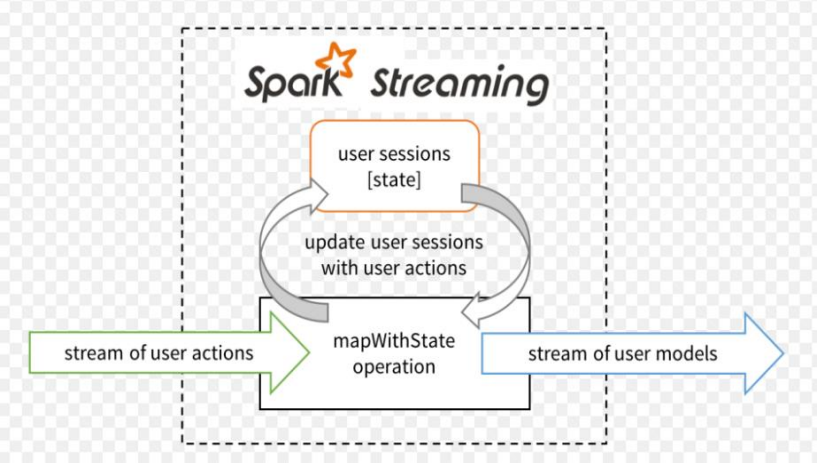
[1] https://nifi.apache.org/

Livy


Apache Zeppelin
Anaconda
Jupyter
Web Browser
curl/wget
Apache Livy
Apache NiFi
Standalone Spark
YARN
Spark Job 1 (Scala)
Spark Job 2 (Java)
Spark Job 3 (Python)
Spark Job 4 (R)
HDFS

Spark Streaming

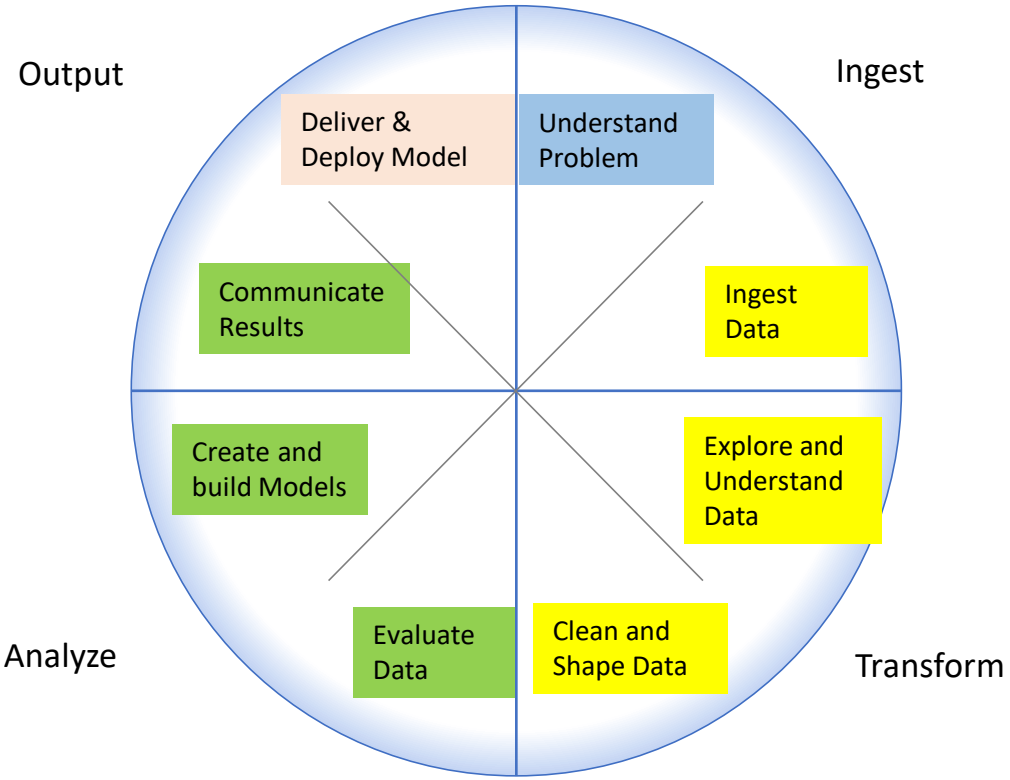


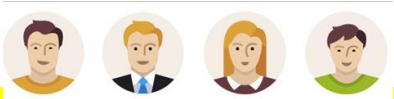


Build the pipeline

Module 2

Traditional AI Tribe





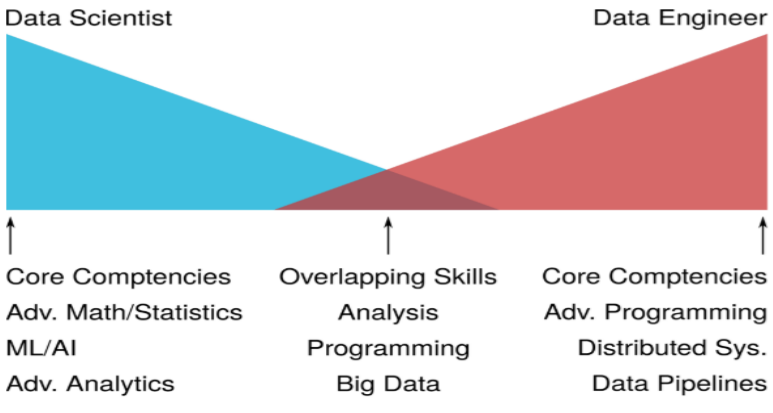
Data Engineer
Architect how data is organized & ensure operability

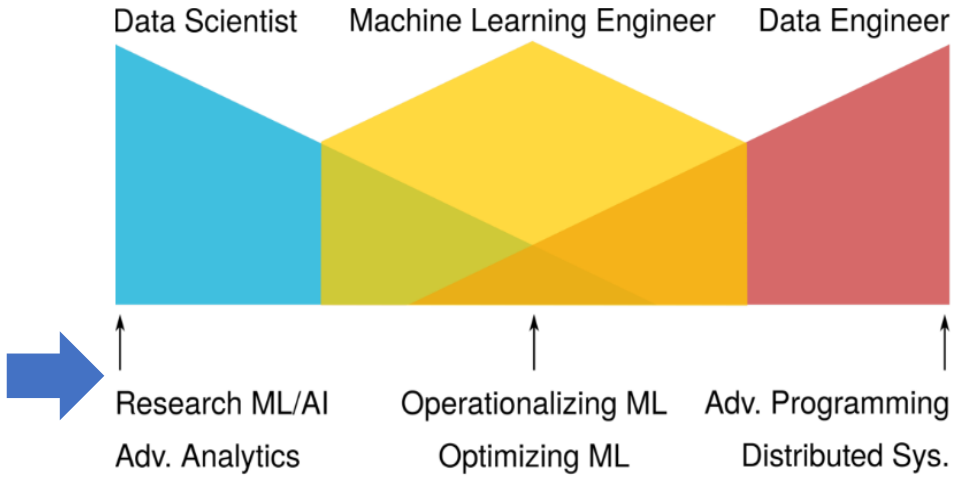
Data Scientist
Deep analytics and modeling for hidden insights

Business Analyst
Work with data to apply insights to business strategy

App Developer
Integrates data & insights with existing or new applications

Trends





About You

You have significant experience architecting, designing, and building scalable cloud-native software platforms that embed ML/AI models, and have hands-on experience with operationalizing and optimizing ML based end-to-end solutions. You have a good understanding of mainstream machine learning algorithms and experience using popular frameworks to train, test, and deploy machine learning models. You're are an excellent communicator and are able to convey complex ideas to different audiences. Above all, you're excited about working with data, and you know how to realize your ideas with code.

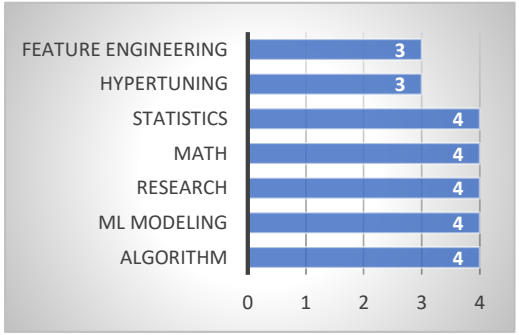
https://vmware.wd1.myworkdayjobs.com/VMware/job/USA-California-Palo-Alto/Staff-Machine-Learning-Engineer_R1813174?from=timeline

Responsibilities

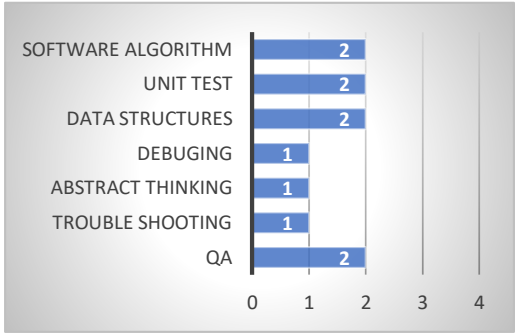
- Architect, design and implement highly scalable ML pipelines (data ingestion, feature extraction, training, testing and validation, inference, and continual learning) in production environments using cloud-native methodologies.
- Stay up to date with state-of-the-art technologies for large scale data processing and machine learning; work with devops team to choose the right tools and operational architecture.
- Have a thorough understanding of the larger ecosystem that encompasses the ML pipeline and implement appropriate interfaces for upstream and downstream services to use.
- Build necessary frameworks and tools, such as logging and A/B testing, to monitor the performance of ML pipelines and models, and constantly look for ways in which the overall product performance can be improved.
- Work closely with data scientists during all stages of data science projects including data cleansing, verifying integrity of data, and developing ML algorithms to address specific use-cases.
- Build prototypes of systems to demonstrate feasibility of proposed ML algorithms.
- Monitor and analyze the effectiveness of new features after they are introduced in the product.
- Design and build abstractions to easily integrate ML models into production data pipelines; enforce software best practices so that the ML code is easy to debug and maintain.
- Work with teams across all functions including Data Science, Data Engineering, and Product Management.

Ratings for traditional data scientist

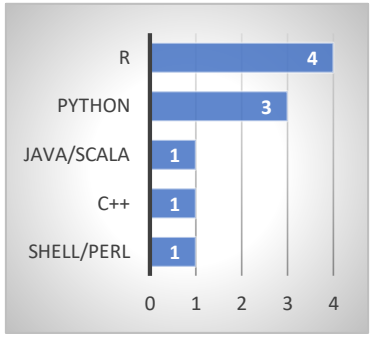
Data Mining



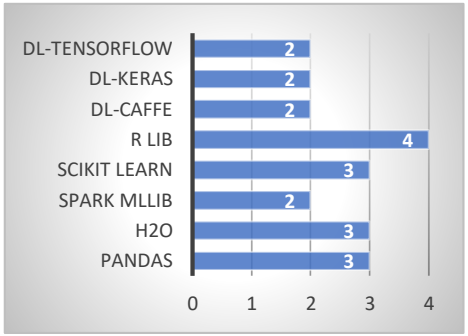
Programing / Coding



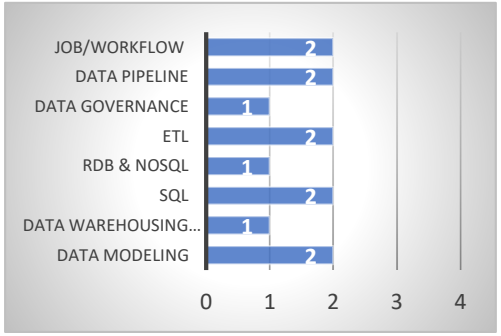
Languages



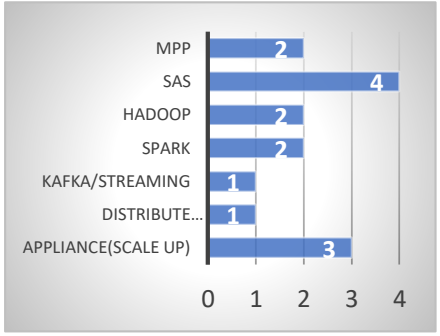
ML/DL Frameworks



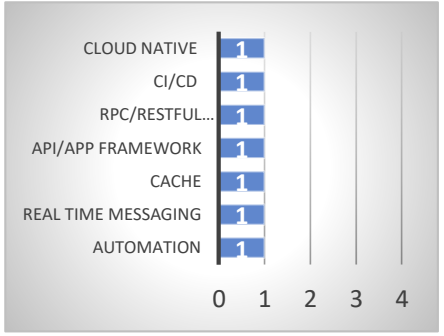
Data Engineering



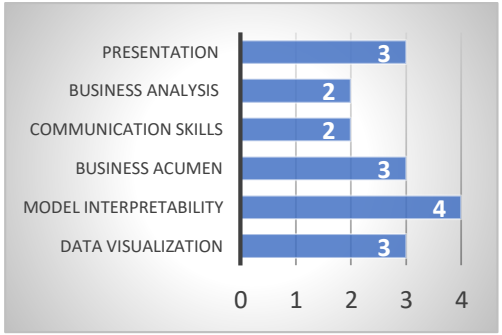
Big Data Stacks



APIs /Services/App (Model Serving)

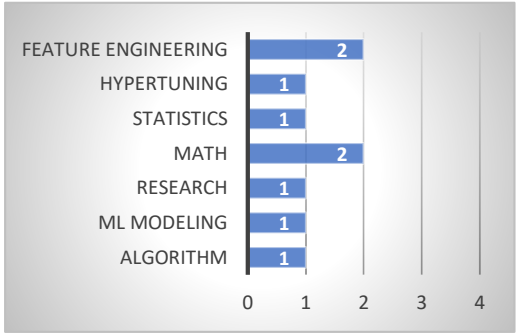


Visualization & Communications

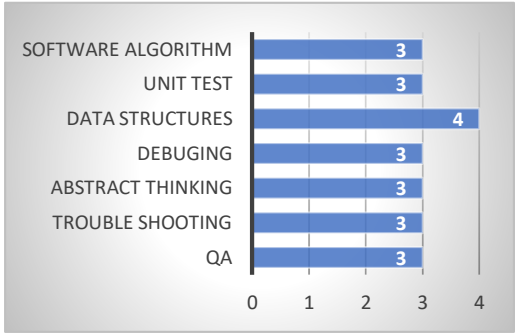


Ratings for traditional data engineer

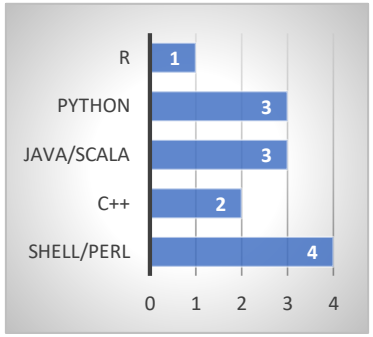
Data Mining



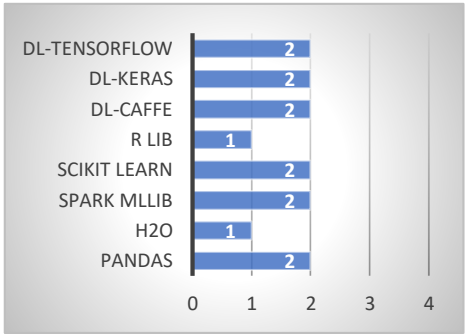
Programing / Coding



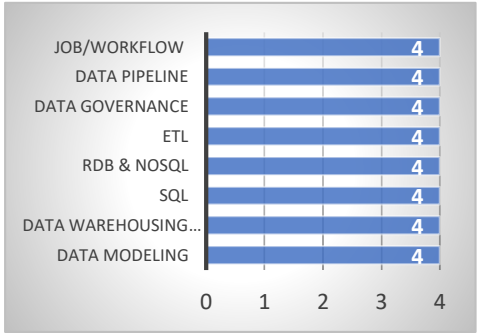
Languages



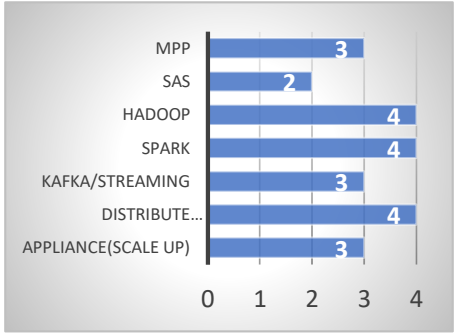
ML/DL Frameworks



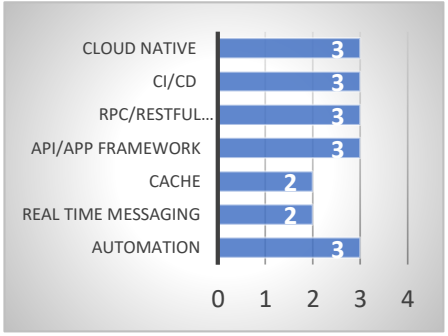
Data Engineering



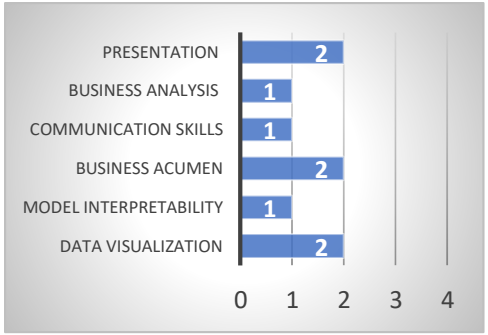
Big Data Stacks



APIs /Services/App (Model Serving)

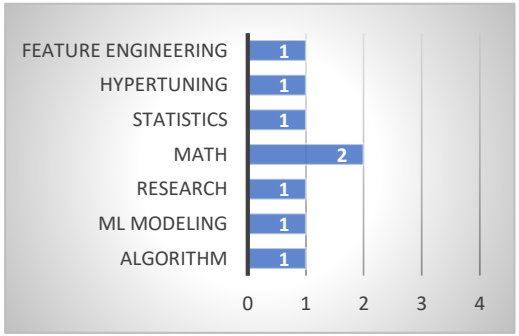


Visualization & Communications

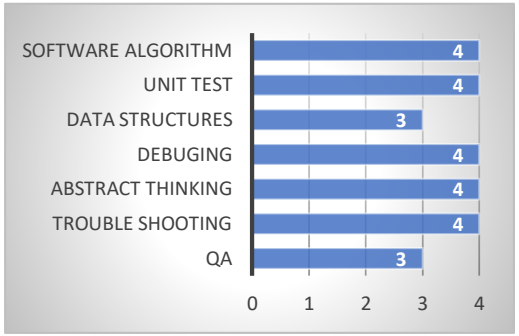


Ratings for traditional application developer

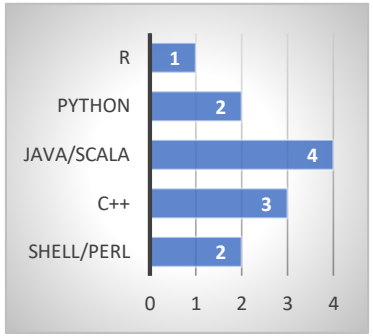
Data Mining



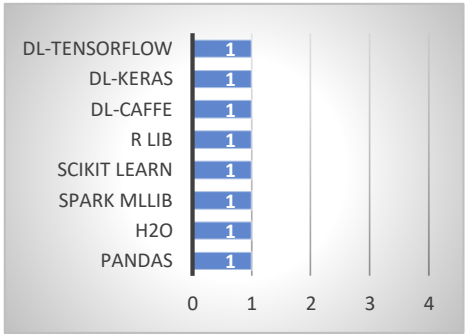
Programing / Coding



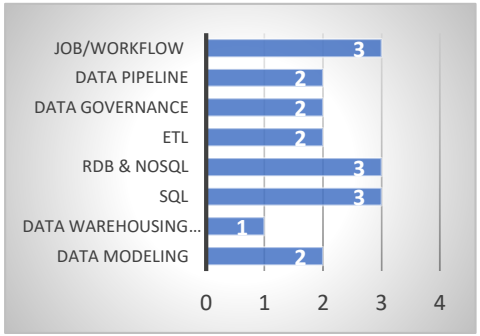
Languages



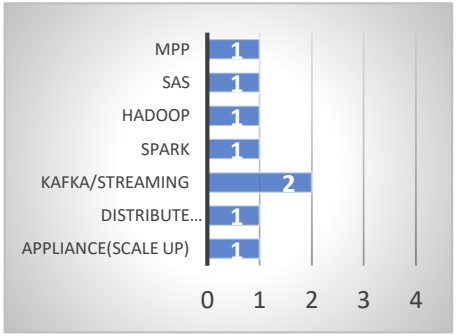
ML/DL Frameworks



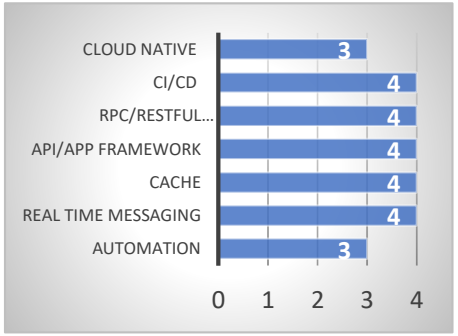
Data Engineering



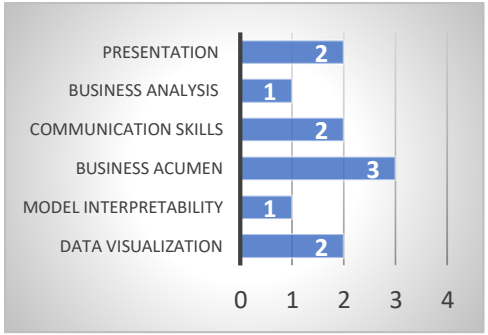
Big Data Stacks



APIs /Services/App (Model Serving)

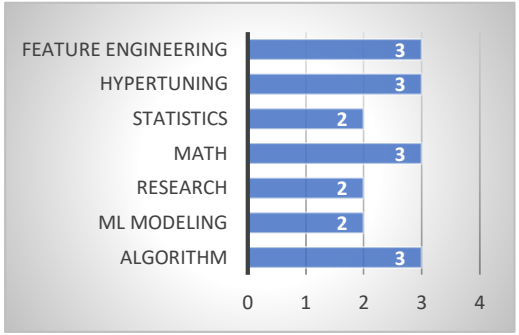


Visualization & Communications

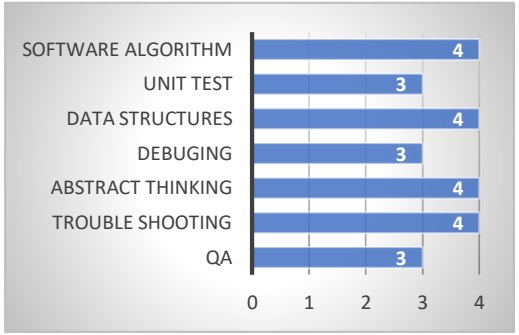


Ratings for modern AI engineer

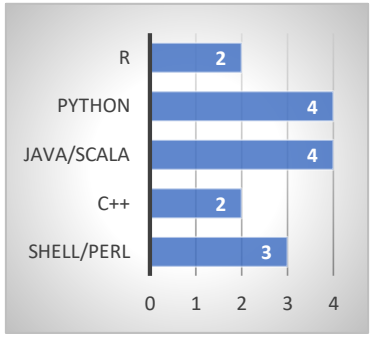
Data Mining



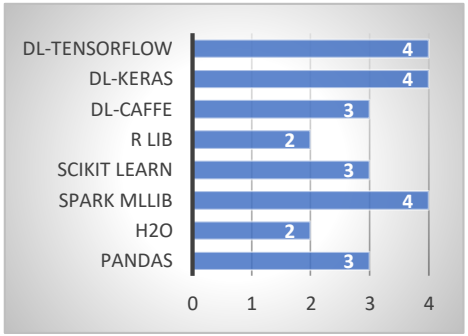
Programing / Coding



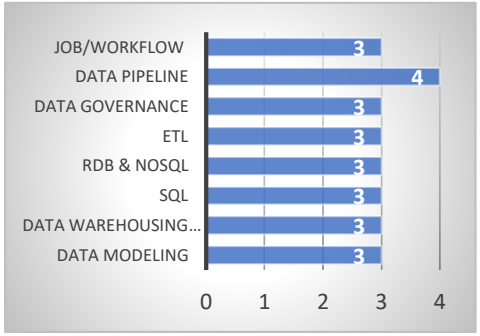
Languages



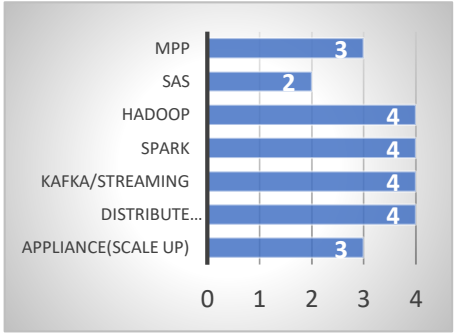
ML/DL Frameworks



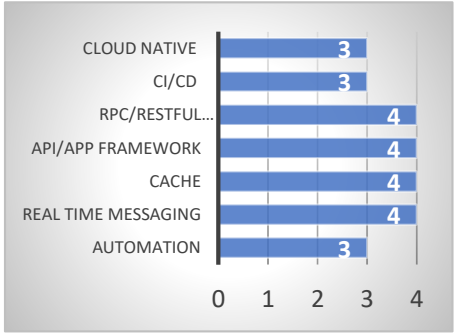
Data Engineering



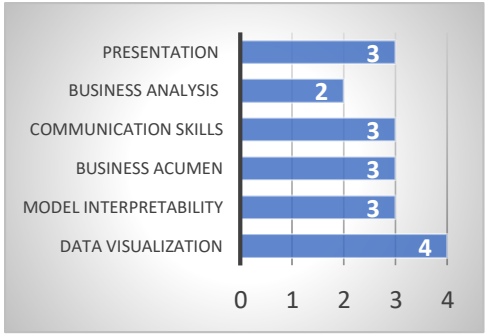
Big Data Stacks



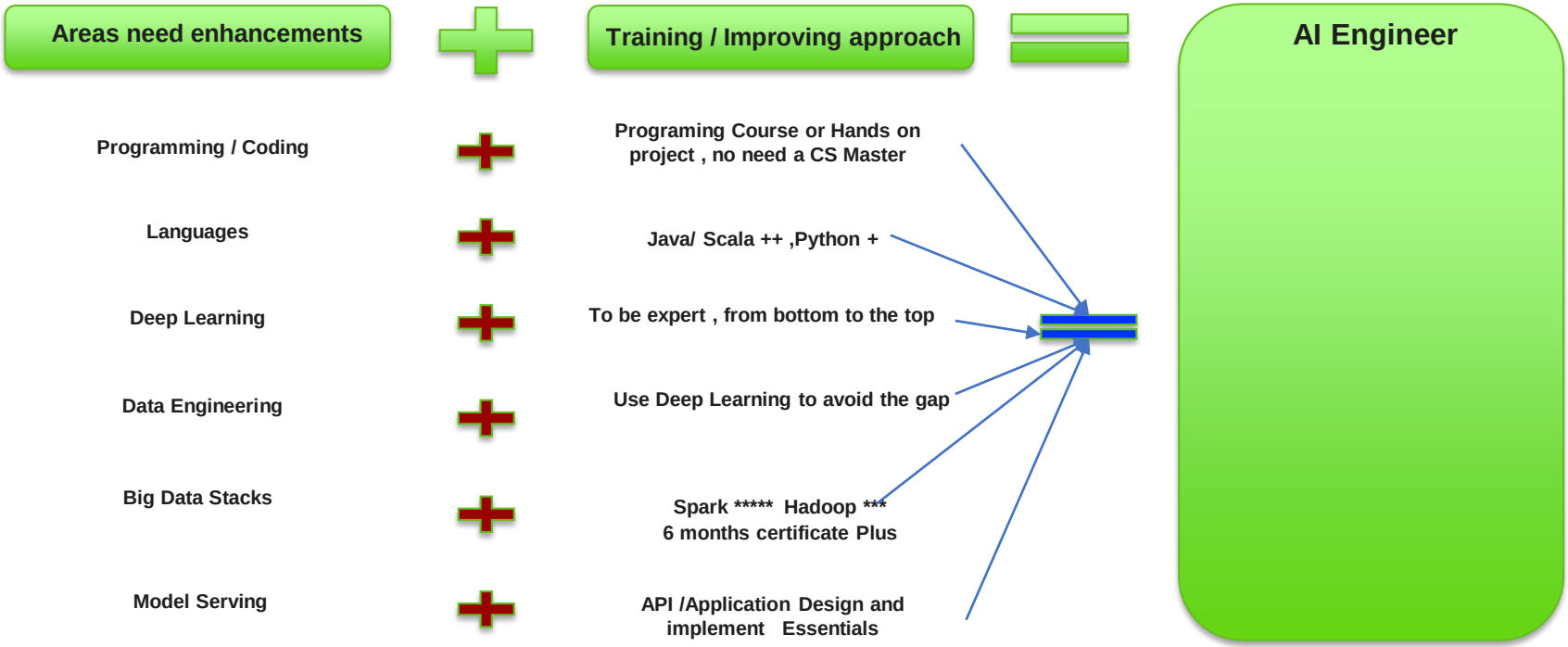
APIs /Services/App (Model Serving)



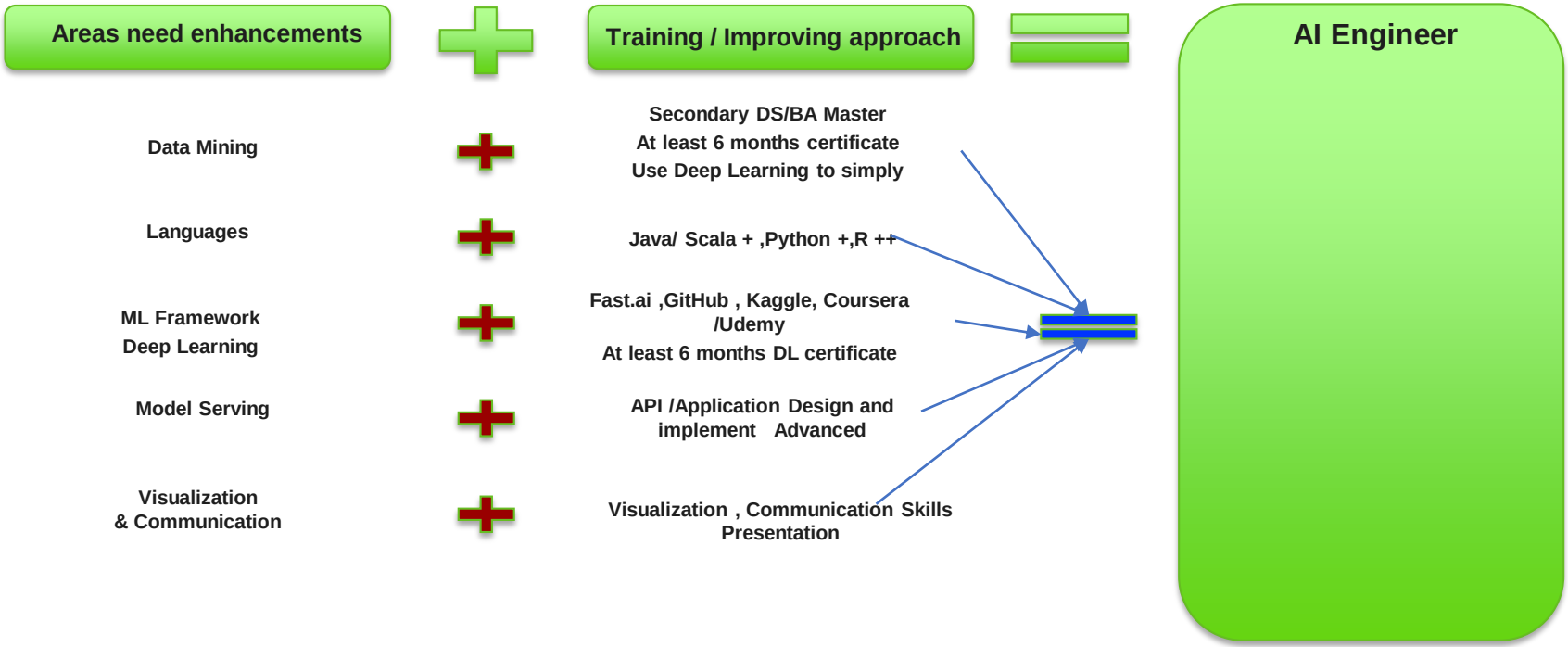
Visualization & Communications



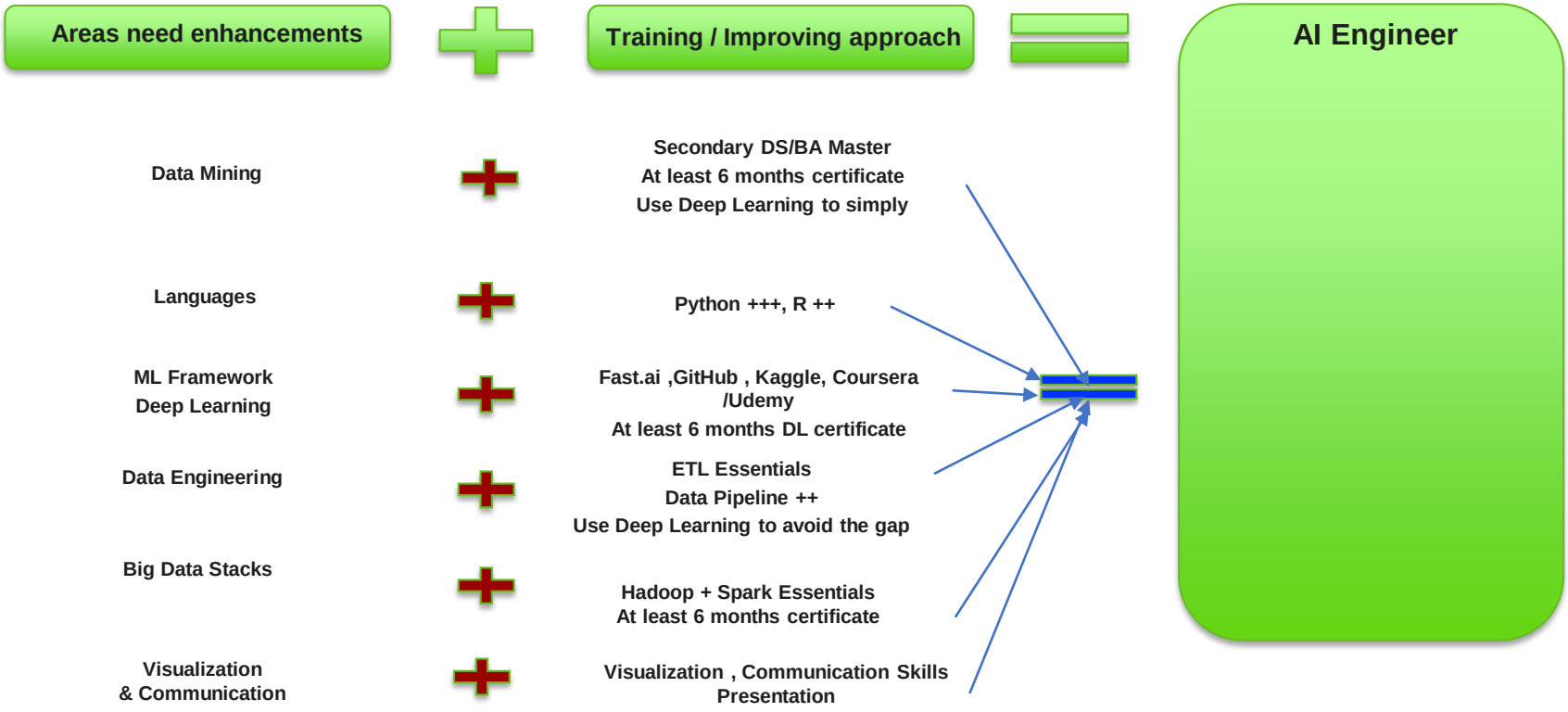
Growing path 1 : traditional data scientist - > AI Engineer



Growing path 2 : traditional data engineer - > AI Engineer



Growing path 3 : traditional application developer - > AI Engineer



Q & A