



# Big Data Analytics and ML Techniques to Drive Impact and Grow Business

Michael Li, Chi-Yi Kuan, Wei Di, Burcu Baran  
Strata Conference, 6 March 2018 Tutorial



**Michael Li**

Global Head of Analytics



**Chi-Yi Kuan**

Director of Analytics



**Wei Di**

Staff Data Scientist



**Burcu Baran**

Senior Data Scientist

# OUTLINE

---

## Introduction

- ❑ Introduction to Data Analytics and Data Science @LinkedIn

## Big Data Analytics

- ❑ Common practices on metric development
- ❑ How can we develop a better product experience?

## Machine Learning

- ❑ Why Machine Learning is important?
- ❑ End-to-end walkthrough of a production modeling solution
- ❑ Common pitfalls and challenges
- ❑ Case Study - B2B Modeling

# OUTLINE

---

## Introduction

- ❑ Introduction to Data Analytics and Data Science @LinkedIn

## Big Data Analytics

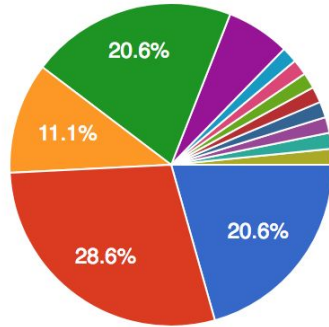
- ❑ Common practices on metric development
- ❑ How can we develop a better product experience?

## Machine Learning

- ❑ Why Machine Learning is important?
- ❑ End-to-end walkthrough of a production modeling solution
- ❑ Common pitfalls and challenges
- ❑ Case Study - B2B Modeling

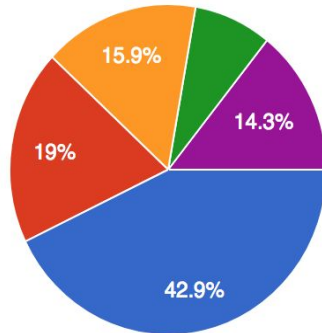


# Survey Results



- Data scientist
- Data analytics
- Engineer
- Business Leader
- Data ETL
- Data Product Manager
- Engineering Executive
- Engineering Director (Management)

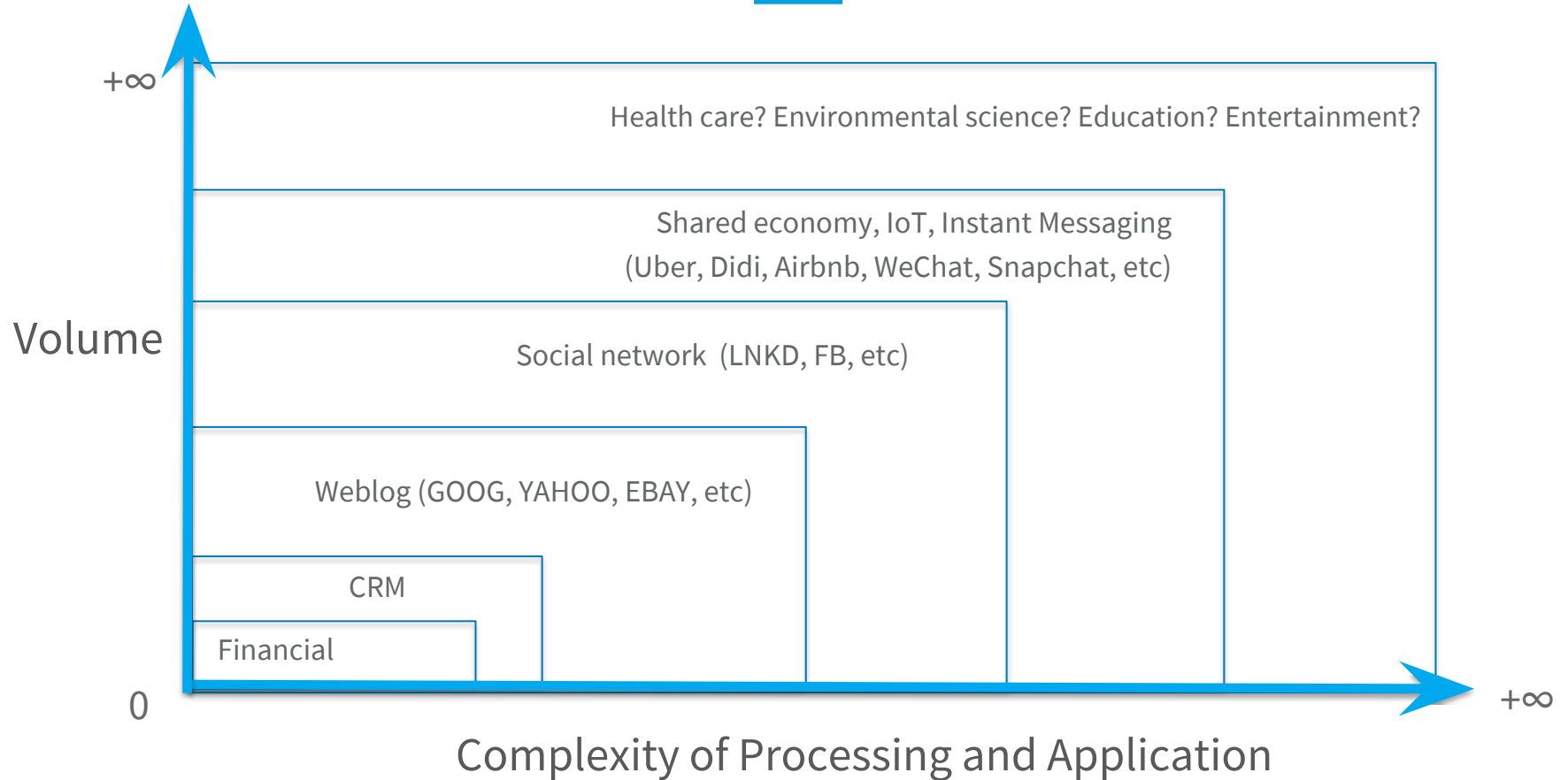
▲ 1/2 ▼



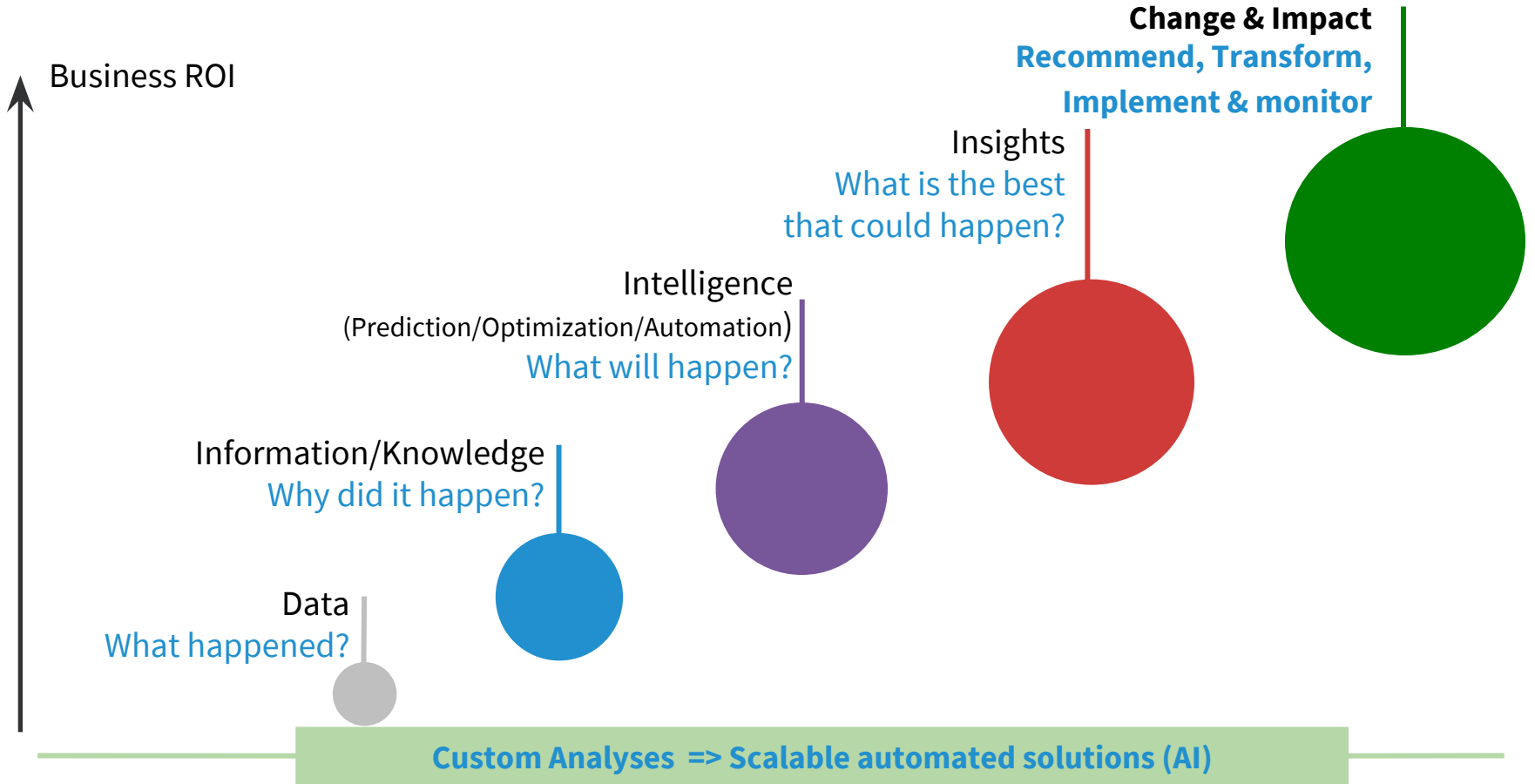
- less than 10%
- 10-30%
- 30-70%
- Over 70%
- I do not know any machine learning technique



# From “Big data” to “AI”?

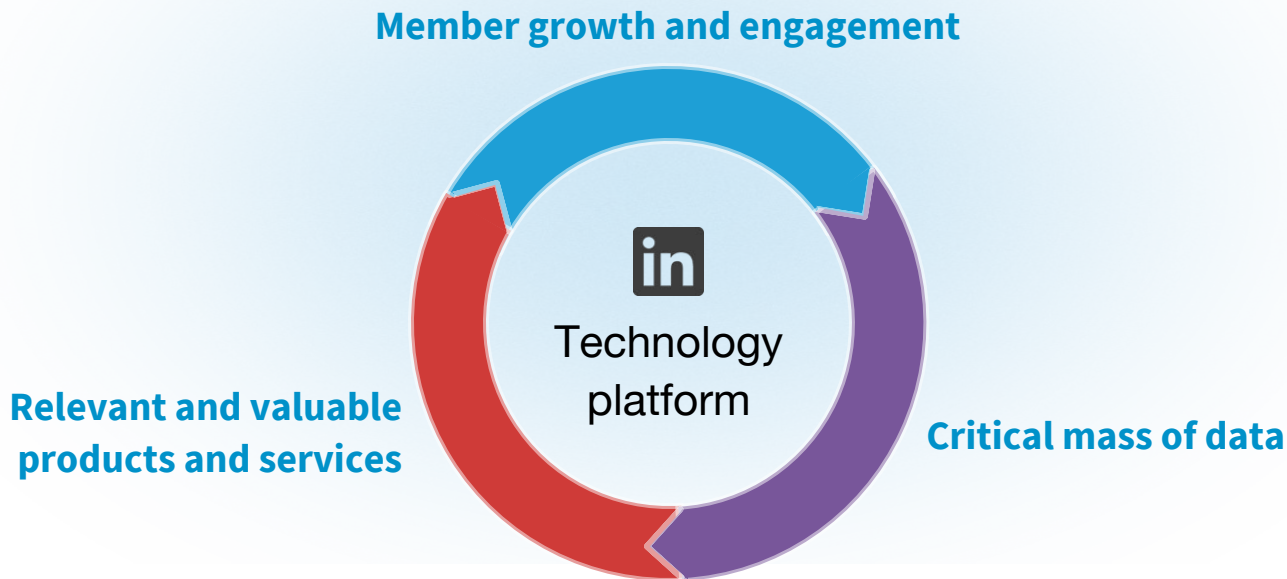


# Analytics evolution: from data to impact



# LinkedIn's business model & why analytics is important

---





## Team Mission

*“Drive understanding and impactful decision making through rigorous use of data.”*

# Analytics Data Science drives business value through the EOI framework leveraging big data

## Empower

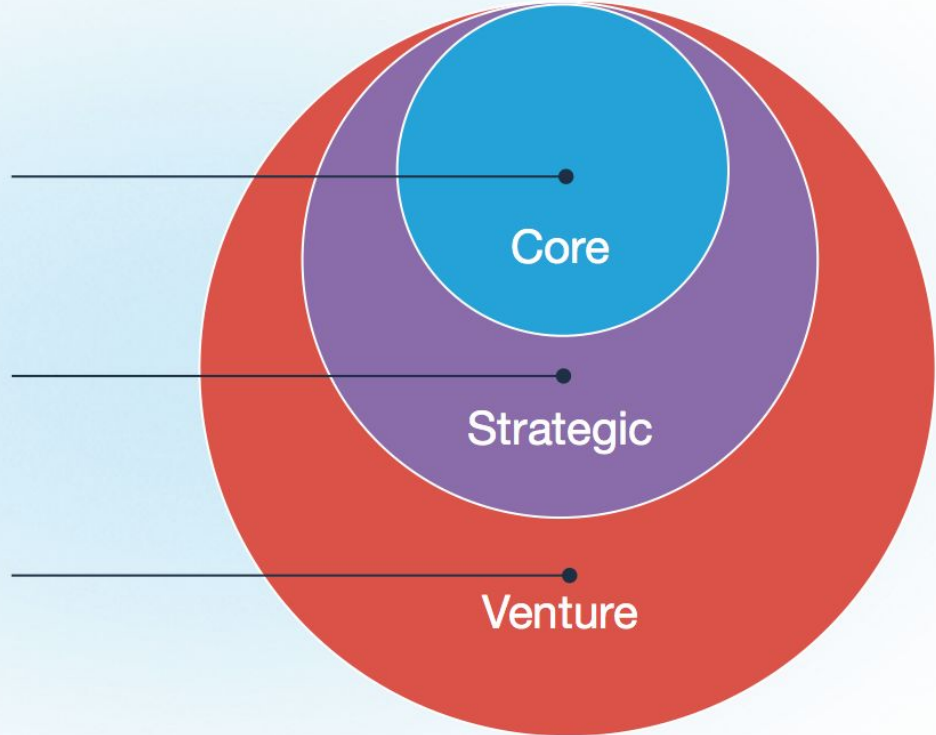
Empower business partners to have access to the data and insights they need when they need them

## Optimize

Optimize business performance leveraging the powerful & unique LinkedIn data we have

## Innovate

Innovate the way on how analytics can help our business grow leveraging both internal & external data



# OUTLINE

---

## Introduction

- ❑ Introduction to Data Analytics and Data Science @LinkedIn

## Big Data Analytics

- ❑ Common practices on metric development
- ❑ How can we develop a better product experience?

## Machine Learning

- ❑ Why Machine Learning is important?
- ❑ End-to-end walkthrough of a production modeling solution
- ❑ Common pitfalls and challenges
- ❑ Case Study - B2B Modeling

# True north metrics guide business

---

**Measuring success:** ultimately differentiates between a sustainable growth strategy and momentary growth

- For each LinkedIn product, we have one primary success metric, which we call our “true-north metric.”
- True north metrics quantify strategy
- Making decisions based on predefined true north metrics ensures that decisions push the business or product in the correct direction.





# True north metrics are common in industry

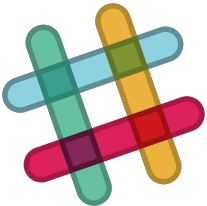
---



Facebook: connect with 10 friends in seven days



Pinterest: Weekly Active Repinners



Slack: teams who have sent 2,000 messages

# A good true north metric needs to be

---



**Aligned**  
with company  
goals or mission



**Comparative**  
Comparison is  
context



**Accurate**  
measures true  
success



**Actionable**  
can be driven by  
product team

*Good examples of true-north metrics for growth are measures like long-term engagement or member retention*

# Potential pitfalls of choosing your metrics

---

Vanity vs. Actionable	Page Views vs. Purchases
Leading indicator	DAU* vs. Active Purchasers
Member retention	DAU vs. Cohort AU
Effective channel	CPC** vs. CPA***
Removing seasonality	WoW vs. YoY

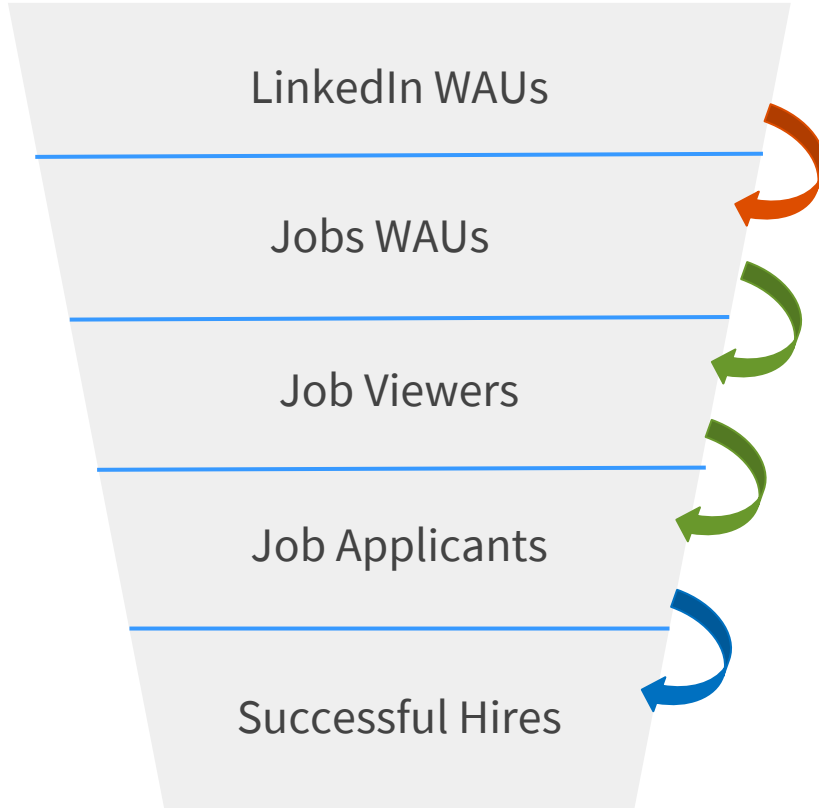
*Example 1: Content A with 1,000 PVs vs. Content B with 2,000 PVs*

*Example 2: For a true-north metric like long-term engagement, we might relate it back to a desired outcome, such as “active members after six months.”*

Notes: (\*) DAU means Daily Active Users; (\*\*) CPC means Cost Per Click; (\*\*\*) CPA means Cost Per Action.

# Example 3: Website funnel analysis

---



## 1. Grow

Drive Jobs WAUs

## 2. Discover

From Jobs WAUs to Job Viewers & Applicants

## 3. Get

From Interested Job Viewers to Applicants and ultimately successful hires

# Quality Member

LinkedIn's true north metric for Growth

# Quality Member definition

---

Quality Member represents the minimum threshold where **any** member can consistently receive value on LinkedIn.



## Profile

~A business card worth of identifying information



## Network

At least 30 connections



## Reachable

Can be contacted by other members

# Create your metric: align with company goals or mission

---

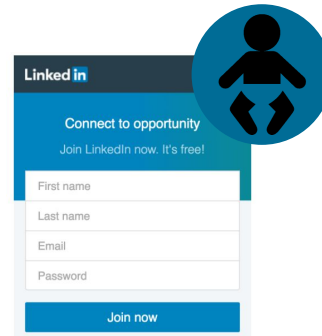
## True north metrics quantify strategy

- Understand the business first, then apply data science (e.g., segmentation, cohort analysis, machine learning, etc.) and A/B testing for the validation
- Have the goal in mind to ensure your metric is aligned

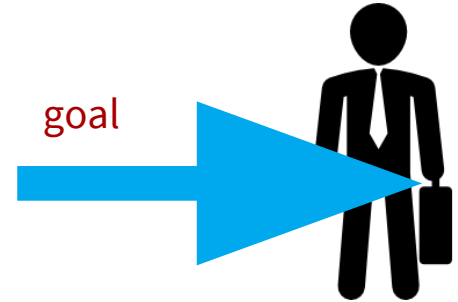
---

## Goal of Quality Member

Designed to measure the number of members who can consistently receive value from LinkedIn.



Account creation is insufficient for value



Quality Member

# Data collection: find examples of success in the data

---

These examples will be the label for machine learning

- The most difficult and influential part of the machine learning process
  - "Am I measuring long-term and sustainable value?"
- 

Success label for LinkedIn growth:

Members who **visit monthly over the last year**

- Proxy for members who are receiving value
- Alternative to defining specific use-cases





# Machine learning: convert your data into insights

---

Any machine learning classifier works. Keep in mind:

- good features and labels beat good algorithms
- one of the end goals is simplicity

Starter pack: logistic regression with L1 regularization

- easily interpreted
- prunes highly correlated features, as well as irrelevant features
- verify the model's performance using standard validation techniques (e.g., ROC curve, etc.)

## **Make it actionable:** everyone should understand the metric

---

General practice: It's okay to sacrifice a moderate amount of predictive power for simplicity and actionable insights.

*Example: the case of Facebook's metric: 10 friends in seven days*

To simplify your ML model:

1. convert your model into a recipe
2. aggressively cut features
3. make it easy to remember



# OUTLINE

---

## Introduction

- ❑ Introduction to Data Analytics and Data Science @LinkedIn

## Big Data Analytics

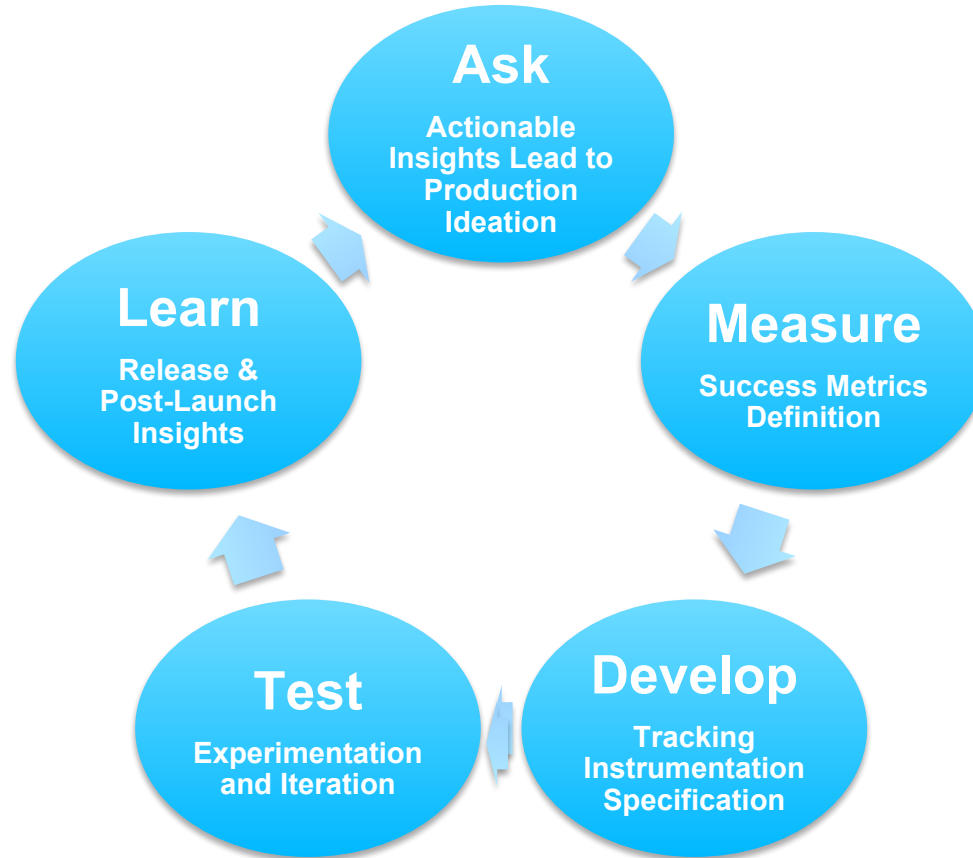
- ❑ Common practices on metric development
- ❑ How can we develop a better product experience?

## Machine Learning

- ❑ Why Machine Learning is important?
- ❑ End-to-end walkthrough of a production modeling solution
- ❑ Common pitfalls and challenges
- ❑ Case Study - B2B Modeling

# Data driven product innovation framework:

Use data to ask, measure, understand, and improve the product experience:



# Example 1:

## Introducing Job Tab

Ask

Measure

Develop

Test

Learn

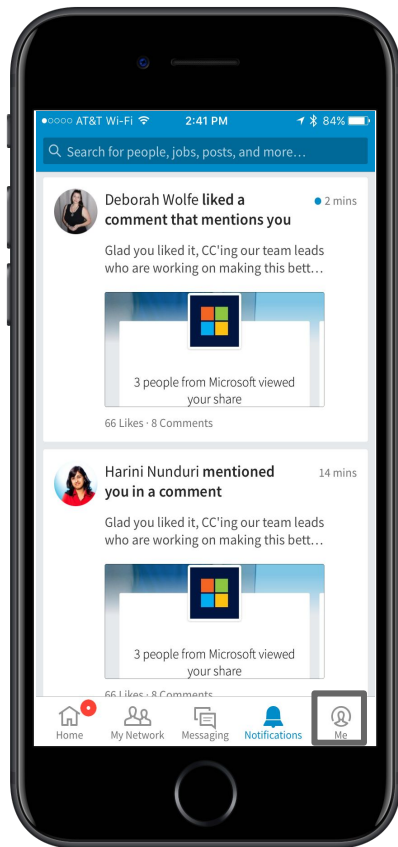
Actionable Insights  
Lead to Product Ideation

Success Metric  
Definition

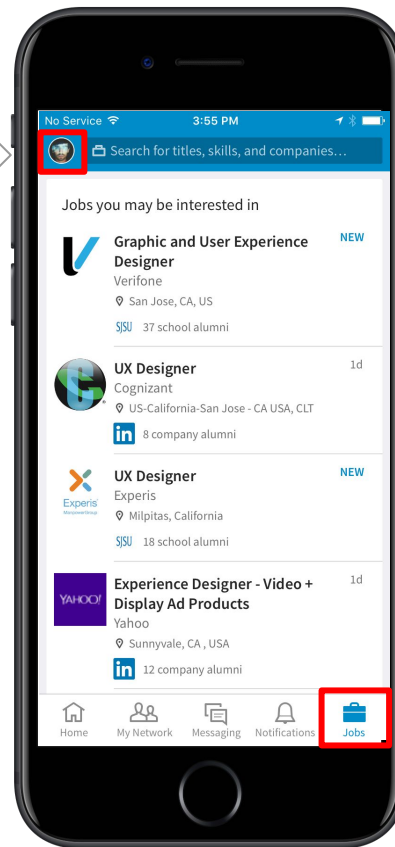
Tracking  
Instrumentation  
specification

Experimentation  
and Iteration

Release &  
Post-Launch  
Insights



Move Profile  
Entry Point

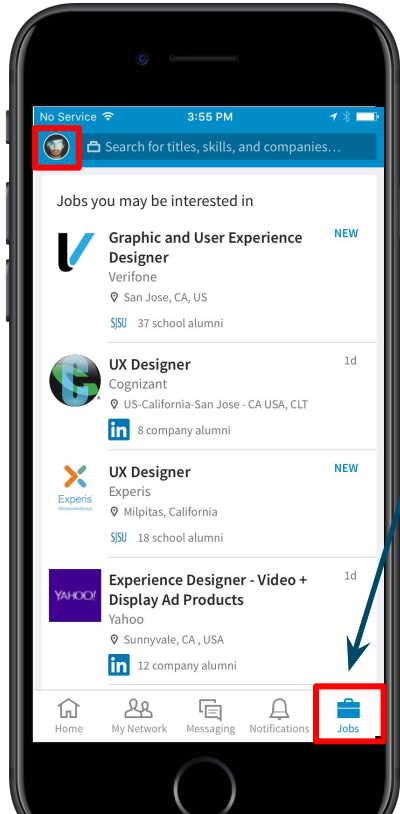


New Jobs  
Entry Point





**Invest in developing the right success metric.**



Hypothesis:

- Improving awareness of jobs in mobile app
- Building a consistent experience between Desktop and Mobile

# What to consider when testing such a change?



## A/B Testing

Hypotheses & Key metrics impacted

➤ **Overall** LinkedIn ecosystem

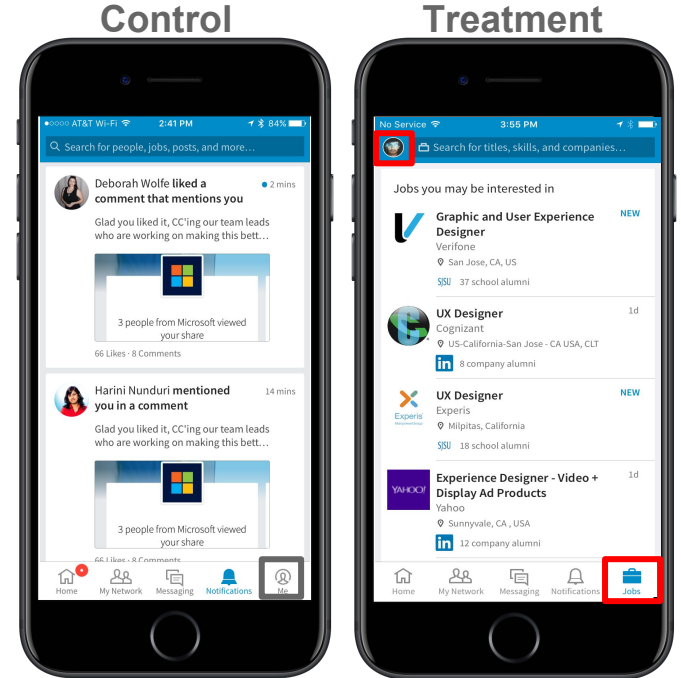
→ Metrics: UU's, sessions, revenue

➤ **“Profile”**

→ Metrics: Self profile views, edits

➤ **“Jobs”** and drive a lot of job applications

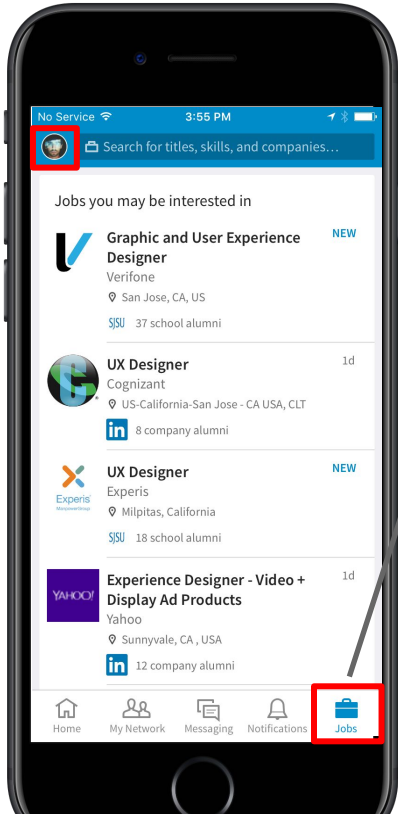
→ Metrics: Jobs UUs, job views, job applies







**Need accurate reliable standardized data logging to enable metric computation.**



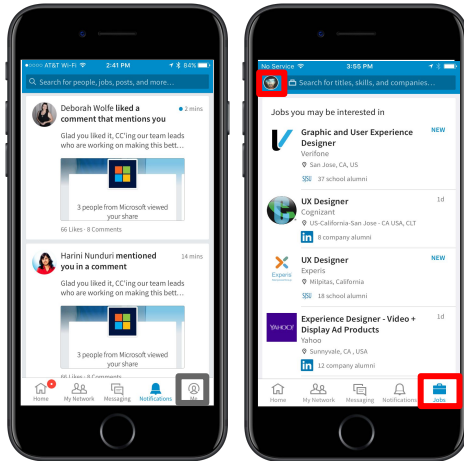
1. Collaborate with product manager to draft tracking specs
2. Align with engineers on what will be tracked and how the data will flow
3. Make sure all the needed data will be available at launch



**Rigorously set up, then identify whether the feature increased the success metric.**

How can we go fast while controlling risk and improving decision quality?

1. Launch to a small portion of members to mitigate risks
2. Reach maximum statistical power to analyze the impact
3. Based on the results: Launch to 100% OR roll back



← Portion of users will have the new experience rolled out to their app



## Hypotheses verified by A/B test

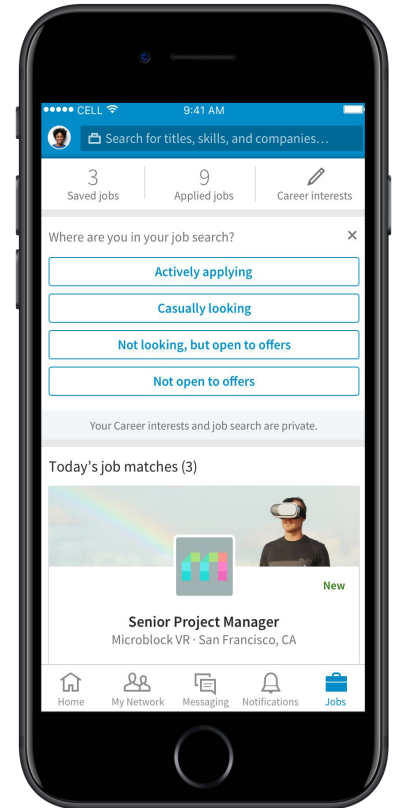
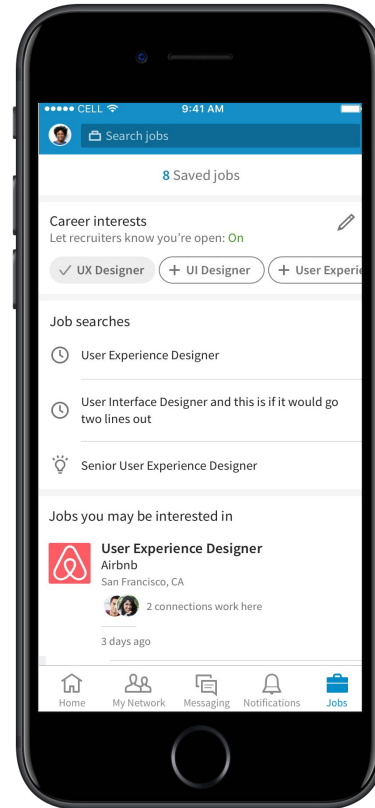
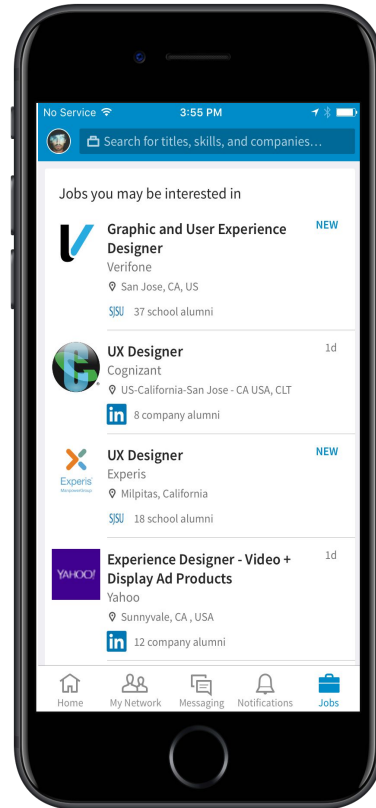
Overall LinkedIn ecosystem	→ Sessions		
“Profile”	↘ Profile Edits	↘ Self Profile Views	
“Jobs”	↗ Jobs UUs	↗ Jobs Views	↗ Jobs Applications

## Recommended next steps

- Ramp the jobs tab to 100%
- [Profile] Build an onboarding tutorial that points out the new location of the ME tab
- [Profile] Add an edit profile promo on the jobs tab
- [Jobs] Improve the tab by adding different type of modules to drive more downstream engagement

# The journey is not done! Keep on improving

*Identify opportunities to continuously improve the experience*



# Example 2:

Recommending additional content

Ask

Measure

Develop

Test

Learn

Actionable Insights  
Lead to Product Ideation

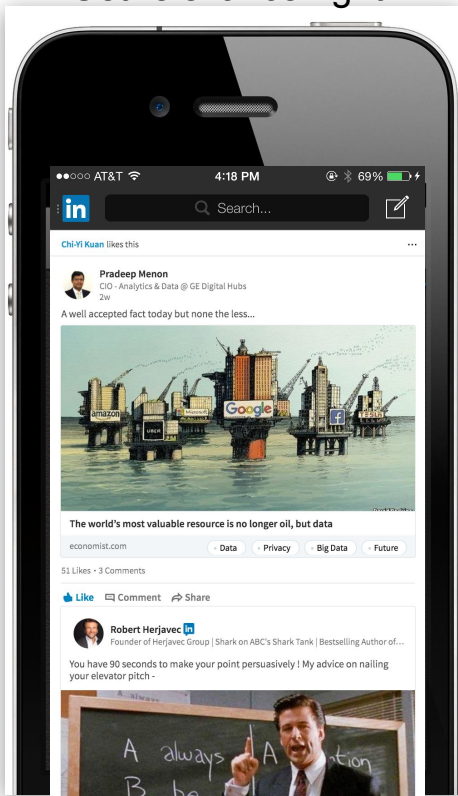
Success Metric  
Definition

Tracking  
Instrumentation  
specification

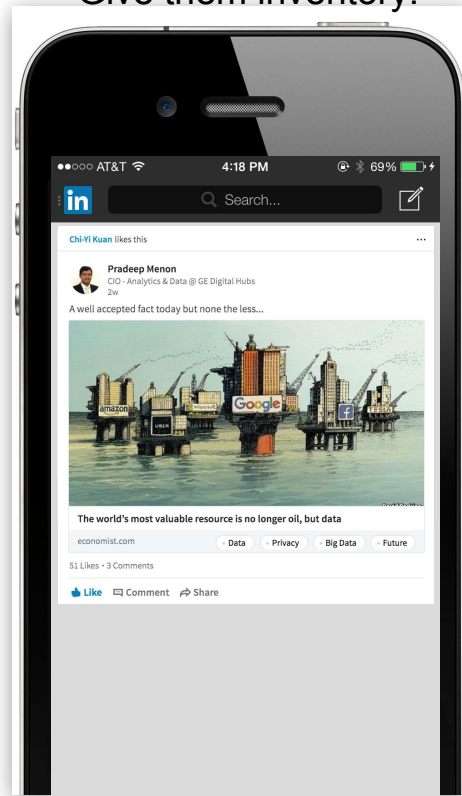
Experimentation  
and Iteration

Release &  
Post-Launch  
Insights

Well-connected.  
Get relevance right.



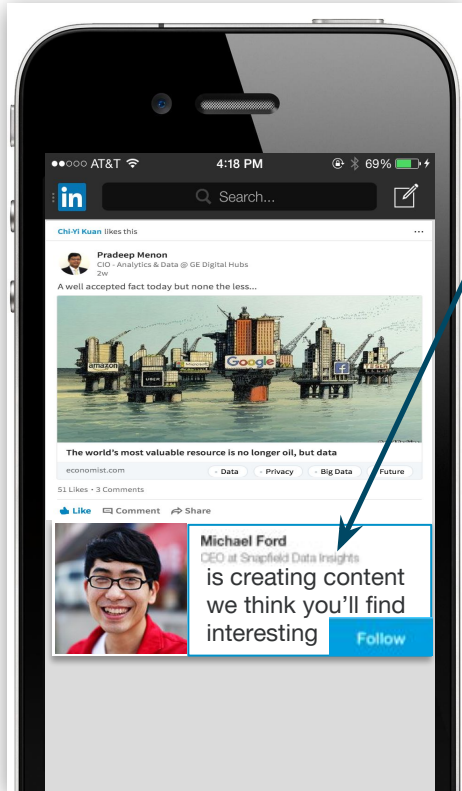
Few connections.  
Give them inventory.



1. Opportunity sizing: how big or important is the problem?
2. Use data to predict successful product initiatives:
  - Show news articles
  - Suggest new connections
  - Suggest following active content creators
  - Show sponsored ads



## Invest in developing the right success metric.



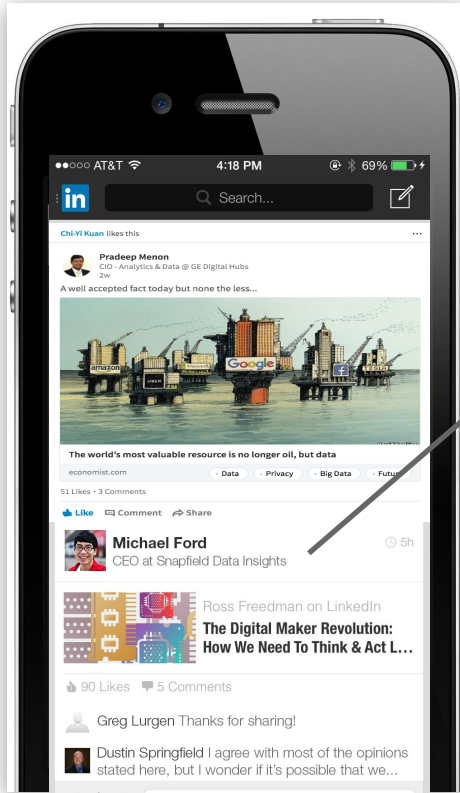
Hypothesis: Following active sources leads to improved user experience with additional Feed

Success Metric – progression of definition:

- Total clicks on Follow
- # clicks / #impressions of Follow suggestions
- % Feed Inventory created by new followees
- Downstream sustained engagement with items created by these followees
  - What is engaging? # of clicks? Time spent? # Shares?



## Need accurate reliable standardized data logging to enable metric computation.



~Metric = Downstream engagement with items created by these followees

Must enable attributing future clicks on feed items to that campaign as a source for the Follow.

```
FeedActivityClick
```

```
{  
  memberID = 77777  
  actor = 55555  
}
```

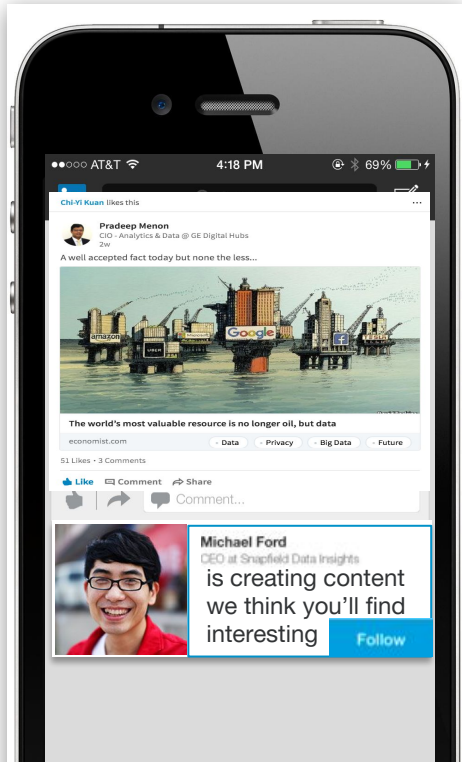
```
FollowSources
```

```
{  
  followCampaign666  
  memberID = 77777  
  followeeID = 55555  
}
```





**Running A/B tests and making decisions based on the movement of a predefined true-north metric ensures that all decisions push the product in the correct direction.**



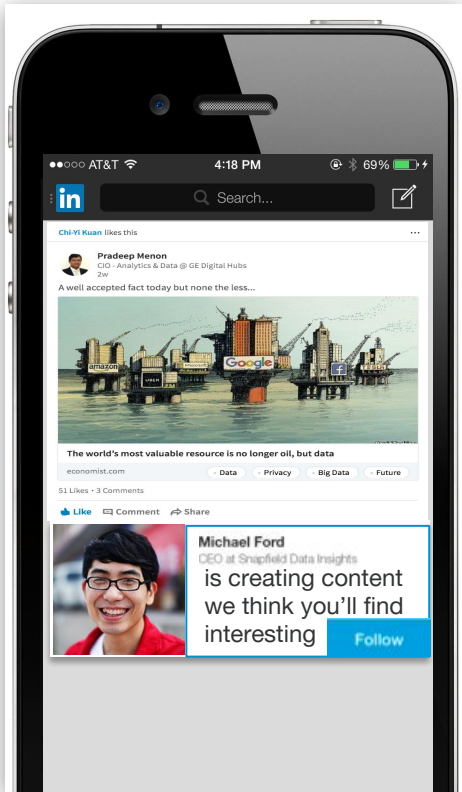
Design: How long to run experiment, on whom?

Implement: Properly set up & randomize to ensure no bias

Analyze: Go or no-go? Monitor success metric, ideally automated on company-wide platform for holistic view of impacts



**Iterate. How can we revise? How can we learn to optimize?**



Reporting, monitoring, ad hoc analysis

Long term measures of engagement/success

Analysis to inform revision of design

**The journey is not done! Keep on improving!**

# OUTLINE

---

## Introduction

- ❑ Introduction to Data Analytics and Data Science @LinkedIn

## Big Data Analytics

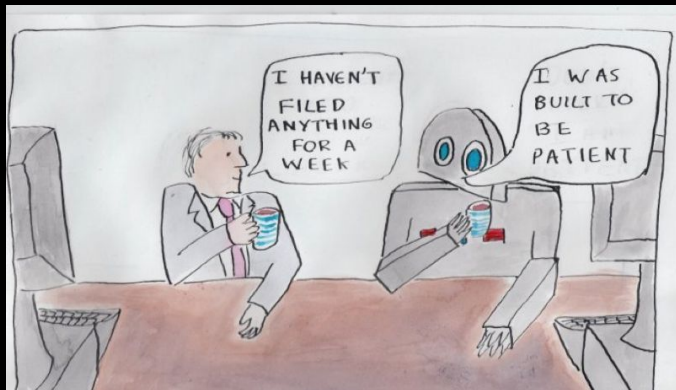
- ❑ Common practices on metric development
- ❑ How can we develop a better product experience?

## Machine Learning

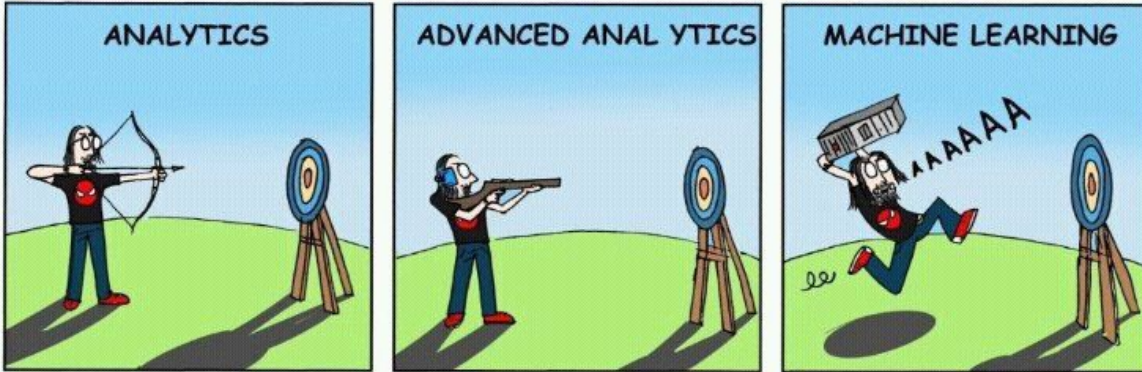
- ❑ Why Machine Learning is important?
- ❑ End-to-end walkthrough of a production modeling solution
- ❑ Common pitfalls and challenges
- ❑ Case Study - B2B Modeling

# Why Machine Learning ?

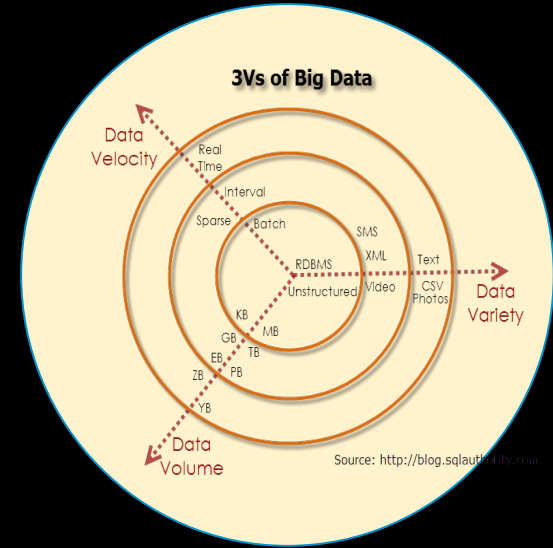
---



# Why Machine Learning ?



SM. MAR 2017.



# LinkedIn's Social Network

---



**500+M**

Members

---



**9M**

Companies

---

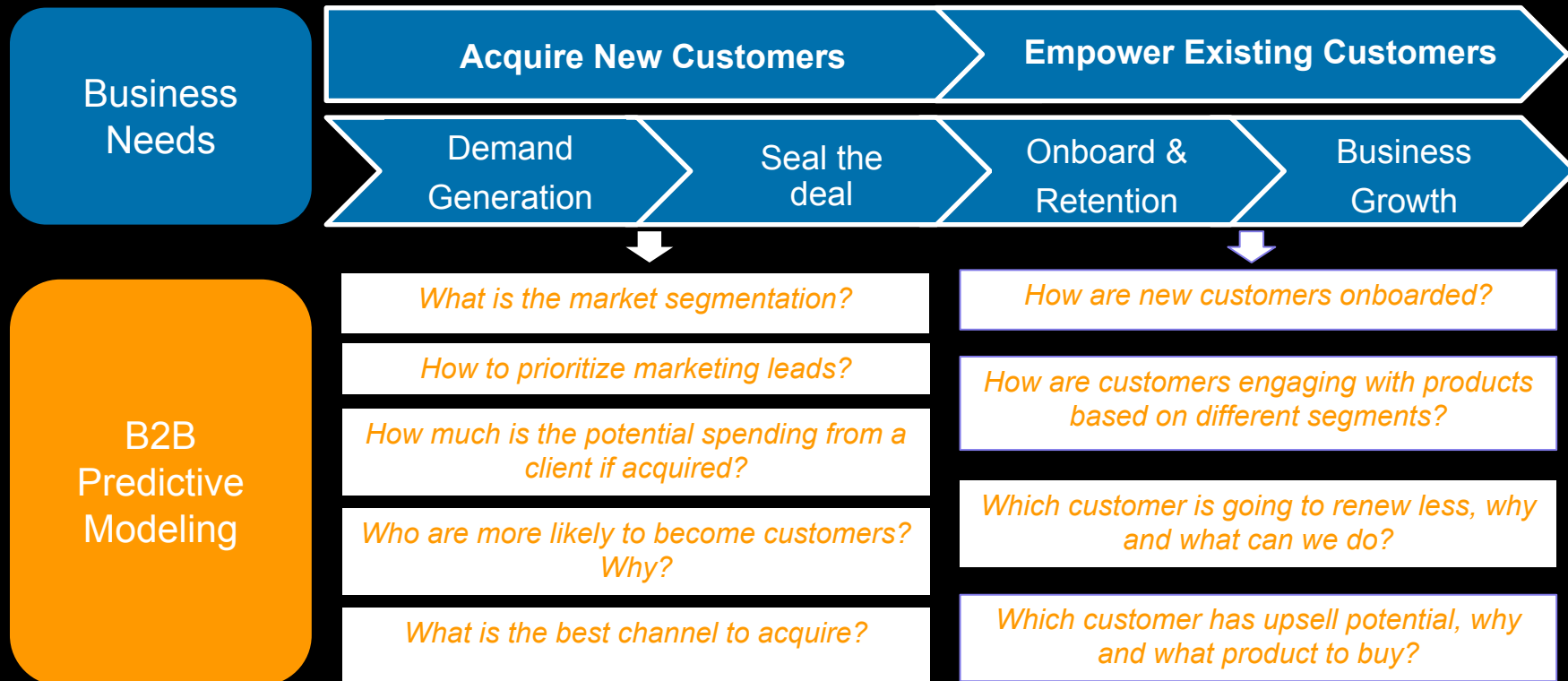


**29K**

Schools

---

# Examples: B2B Analytics





# Examples

---



Fraud/Spam Detection



Machine Translation



Speech Recognition

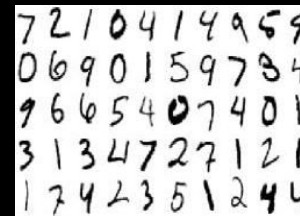
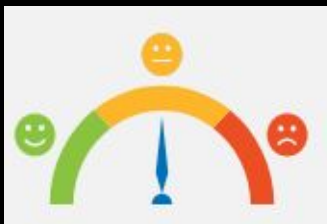


Image Recognition



Sentiment Analysis



Chatbot



Search

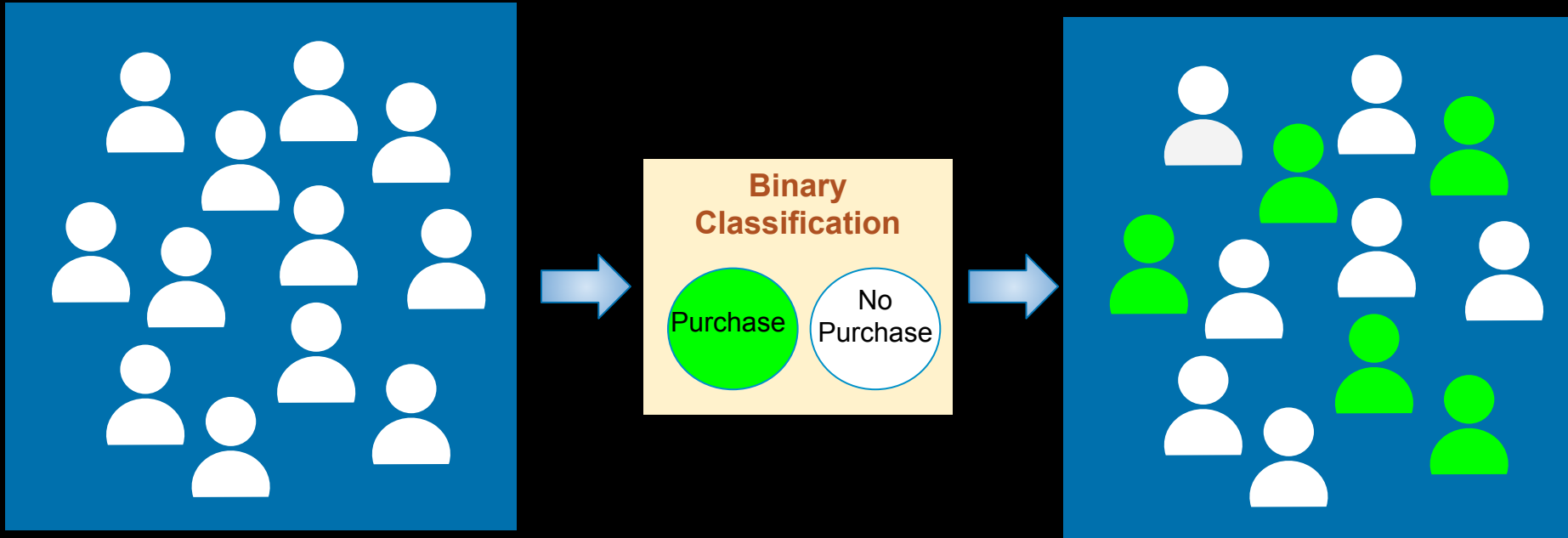


Self Driving Car




# Example: B2C Analytics

Consumer Level: Predict user's intention or action, e.g. click, purchase, churn, etc.



# Example: Web Content Analytics

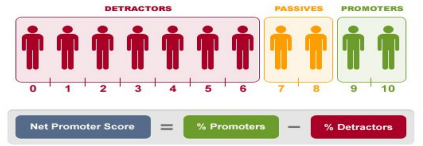
**Social data**



**Customer feedback**

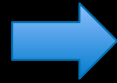
- Customer service
- Group updates
- Network updates

**Survey results**



DETRACTORS (0-6) | PASSIVES (7-8) | PROMOTERS (9-10)


Net Promoter Score = % Promoters - % Detractors




**Text Classification**




**Relevance**



**Products Categorization**



**Sentiments**



# OUTLINE

---

## Introduction

- ❑ Introduction to Data Analytics and Data Science @LinkedIn

## Big Data Analytics

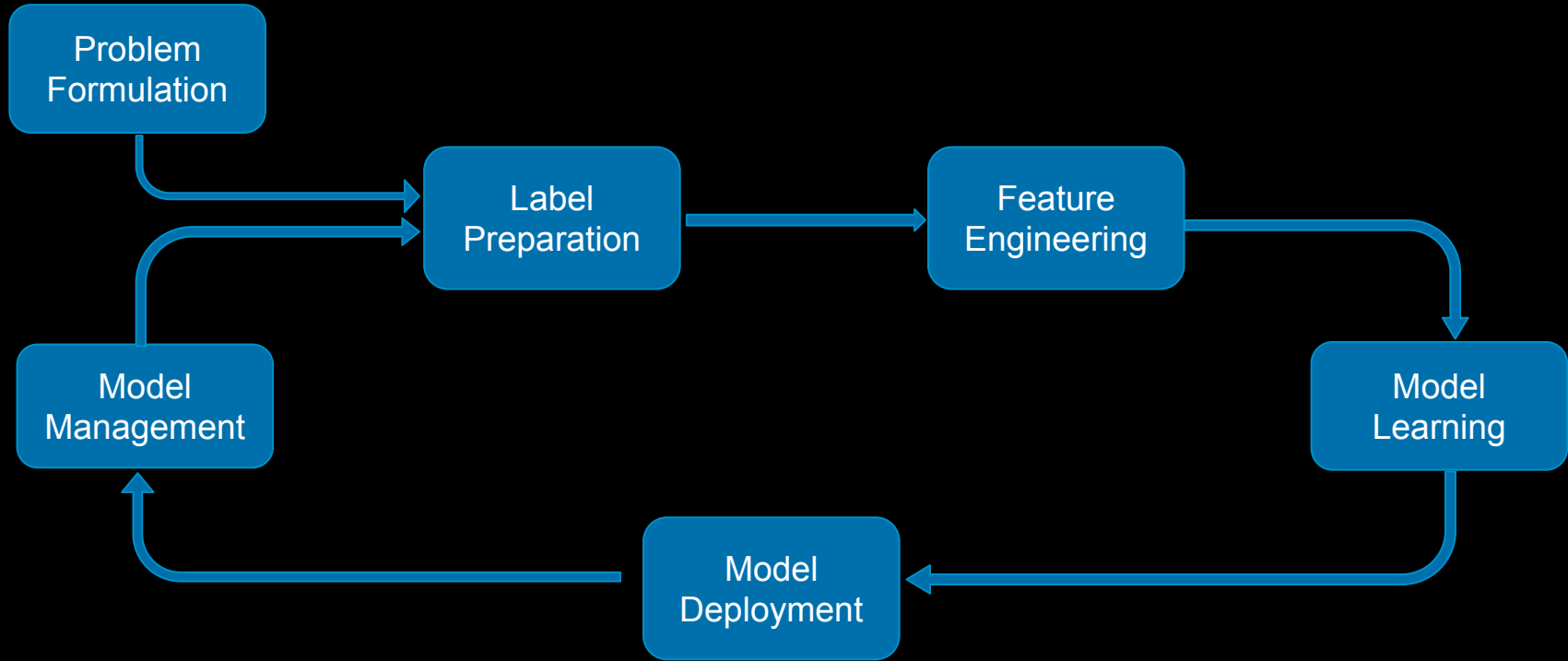
- ❑ Common practices on metric development
- ❑ How can we develop a better product experience?

## Machine Learning

- ❑ Why Machine Learning is important?
- ❑ End-to-end walkthrough of a production modeling solution
- ❑ Common pitfalls and challenges
- ❑ Case Study - B2B Modeling

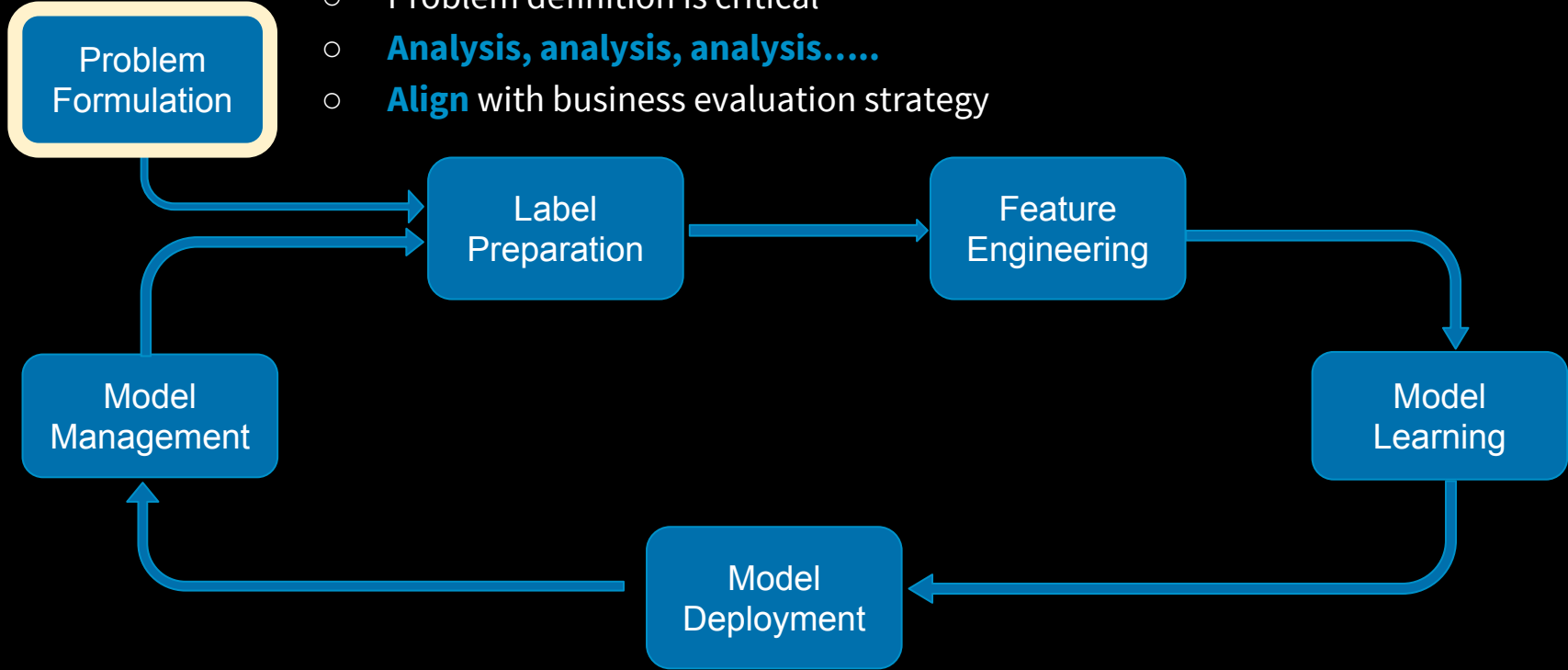
# Machine Learning is a “Process”

---

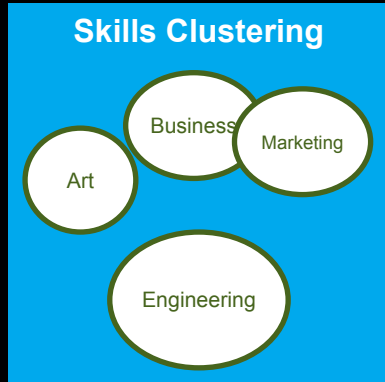


# Machine Learning: The Process

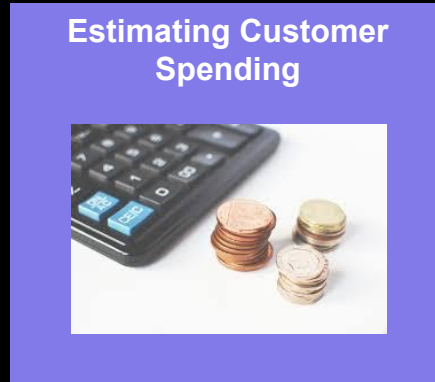
- Problem definition is critical
- **Analysis, analysis, analysis.....**
- **Align** with business evaluation strategy



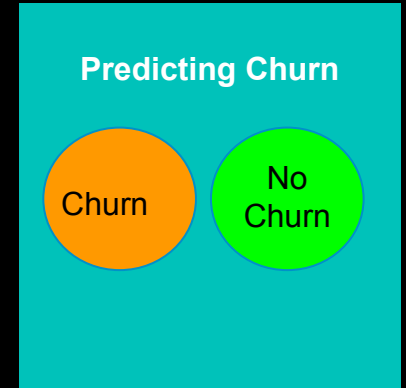
# Problem Formulation



**CLUSTERING**



**REGRESSION**



**CLASSIFICATION**

# Problem Formulation

## Example: Job Seeker Subscription Model

*Assume we periodically send marketing promotions / campaigns to LinkedIn members for job-seeker subscriptions. How do we decide who we should send these emails to?*

*Binary classification problem: let  $y_i$  represents the product subscription status of member*

$$y_i = \begin{cases} 1 & \text{: subscriber} \\ 0 & \text{: otherwise} \end{cases}$$

$$P(y_i | \text{member}, \text{context}) = ?$$

Data Scientist San Francisco Bay Area Find jobs

in Apply Save

Senior Data Scientist  
**LinkedIn**  
Mountain View, CA, US

12 connections work here  
Posted 2 days ago

Job 1 Job 2 Job 3

**Job Seeker Subscription**

Reaching out

- Who's viewed your profile
- InMail™ messages

Finding the right people

- Premium search

Job seeking

- Featured applicant
- Applicant insights
- Salary data

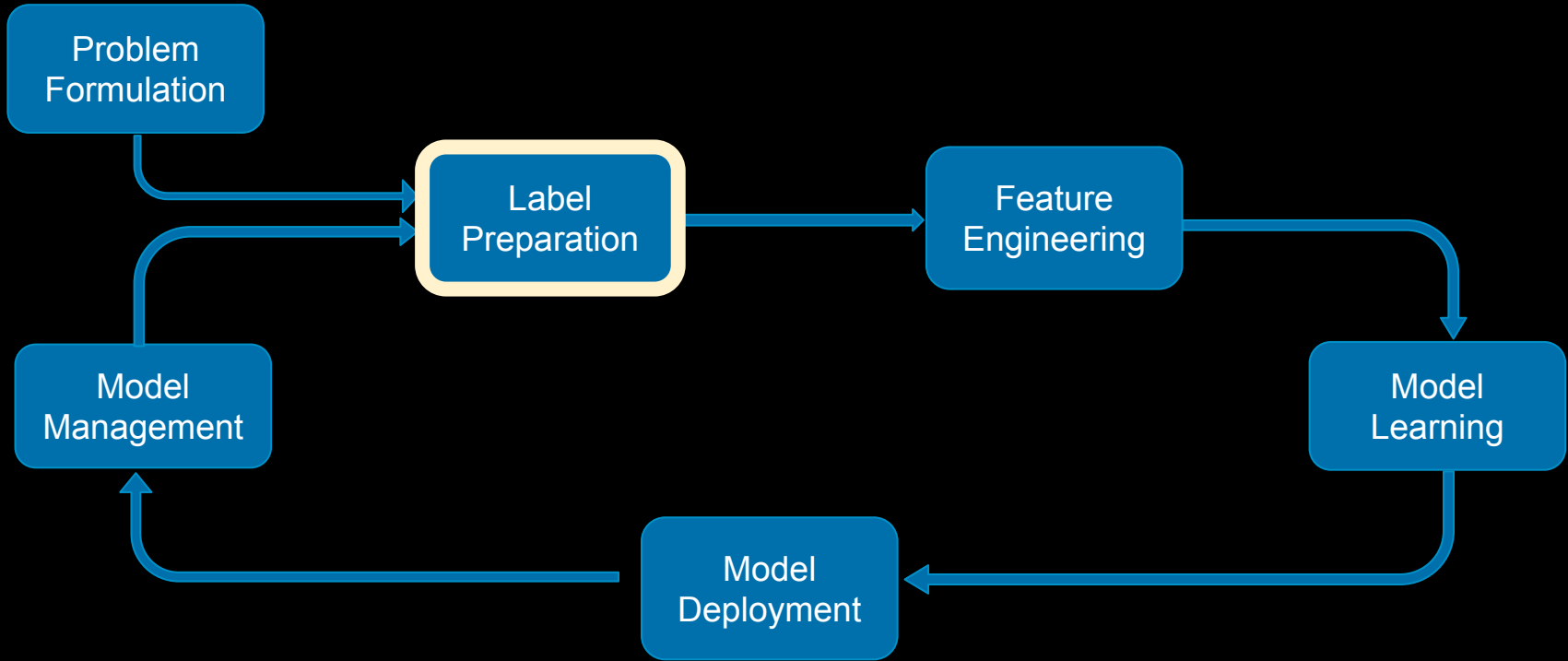
Standing out

- Premium Profile
- Larger search listings

Buy this plan

# Machine Learning: The Process

---





# Label Preparation

---

- A set of **labels** (“right answers”) need to be defined in advance
- Methods
  - Derive from data
    - Historical data
    - User preference
    - User activity
  - Domain expert
  - Scale up label collection
    - Crowdsourcing

Example 1: Job seeker subscription model

Label: whether subscribed {0:no, 1:yes}

Example 2: Churn prediction

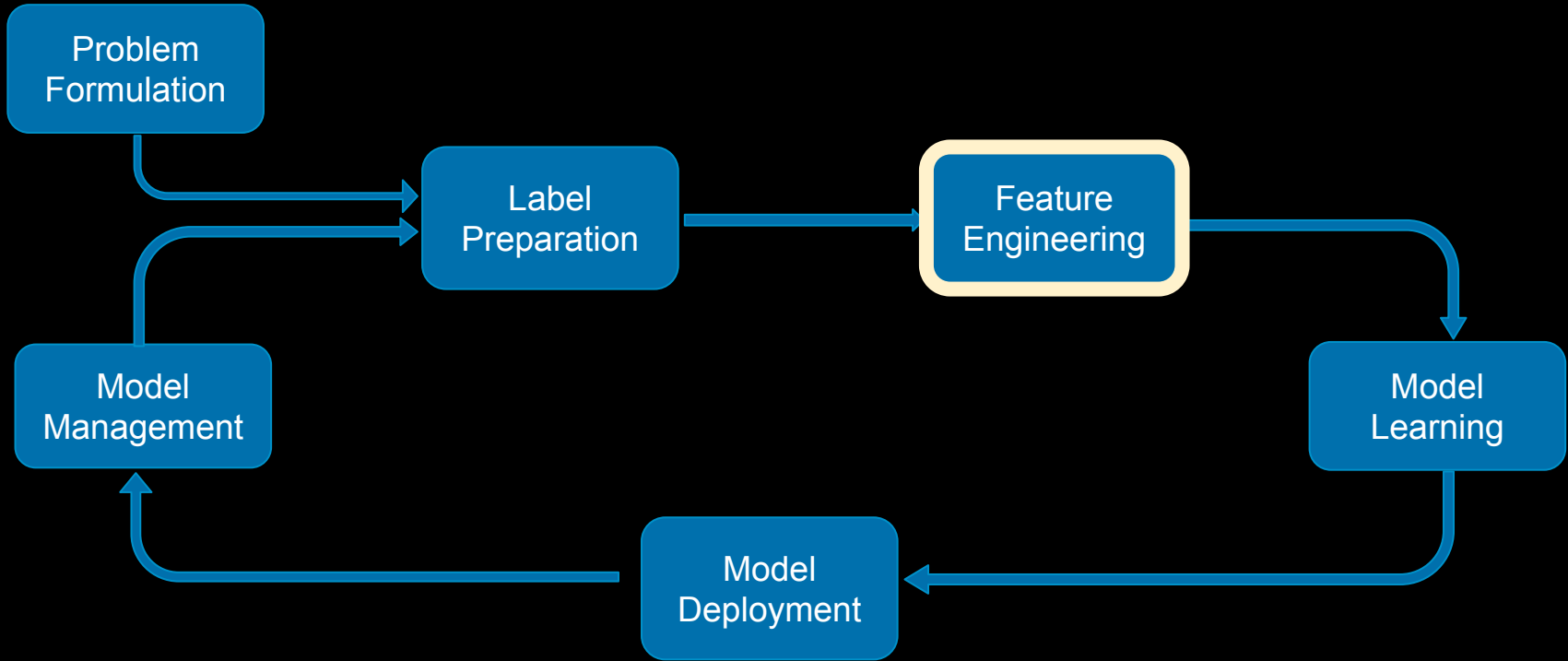
Label: close/renew

Example 3: Sentiment analysis

Label: sentiment types {strong negative, negative, neutral, positive, strong positive}

# Machine Learning: The Process

---



# Feature Collecting

---

- **Feature** is an individual measurable property or characteristic of a phenomenon being observed



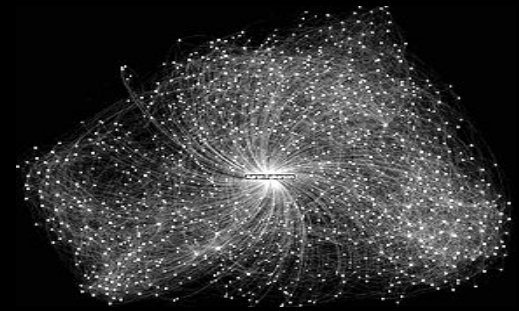
## Identity Features

- *Demographics*
- *Personal and professional interest*
- ...



## Behavioral Features

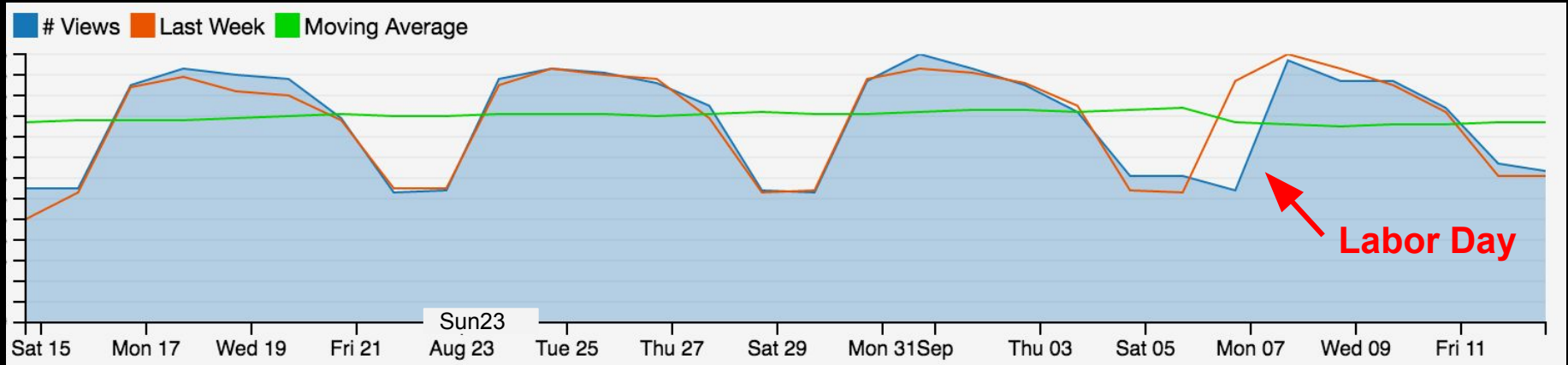
- *pageviews*
- *searches*
- *activities on external sites*
- ...



## Social Features

- *Social network identity and behaviors*
- ...

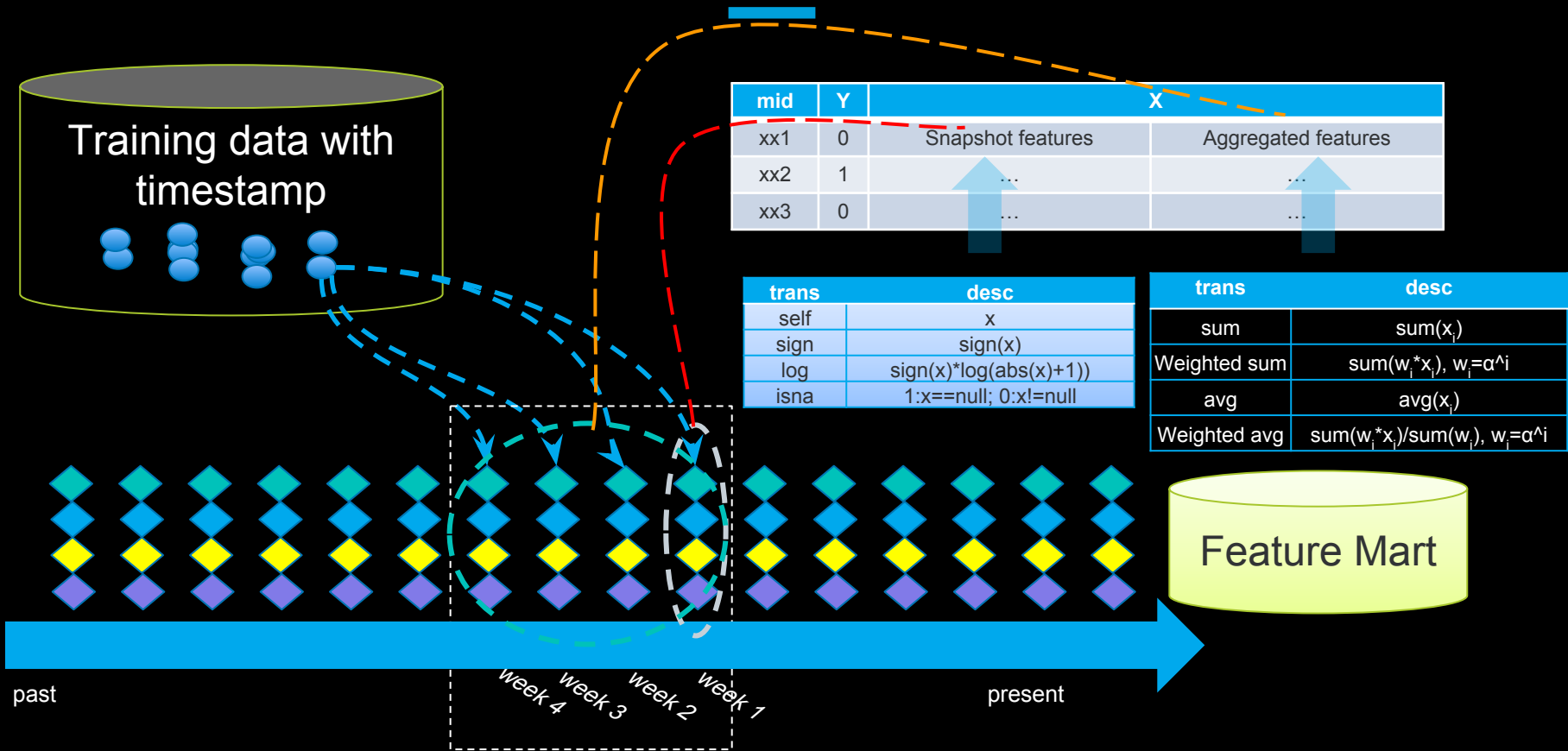
# Feature Engineering - Quality Monitoring



- Compute **basic statistics** such as: sum/avg/coverage/percentiles
- Understand the **intrinsic characteristics** of the feature: dynamic/volatile in nature or static
- Define **anomaly with context**: seasonal, product evolution, etc
- Approach: percentage change, T-test, etc -> aware of any **statistical assumption** restrictions



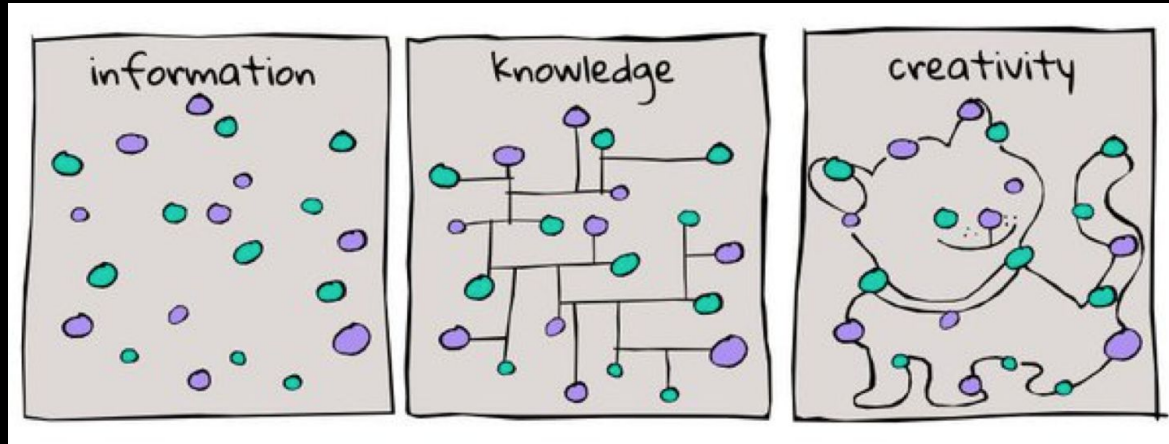
# Feature Engineering - Static & Dynamic



# Feature Engineering - Clean up & Transformation

---

- **Clean-up:** remove outliers, check NULLs
- **Transform:** transform raw data into features that better represent the underlying problem to the predictive models: important for linear models



# Feature Engineering - Transformation

---

- **Numeric** values:
  - Separate data into clusters
  - Reduce dimension & keep most information: PAC
  - Convert distributions to satisfy algorithm assumption: de-mean, unit variance, log
  - Non-linear to linear
  - Continuous to discrete: buckets based on histograms of dat
- **Categorical** values
  - Convert to numbers
    - one-hot-encoding: binary indicator for each categorical value
    - ordered categorical (ordinal) 1-10-> 5, 20-30-> 25
  - Combine levels: when levels are skewed
- **Interactions**
  - cross-products of feature types, e.g.
    - $\{\text{skills}_{\text{member}}\} \times \{\text{skills}_{\text{job}}\}$
    - $\{\text{skills}_{\text{member}}\} \times \{\text{industry}_{\text{job}}\}$
- **NULL**
  - Dummy variables indicator

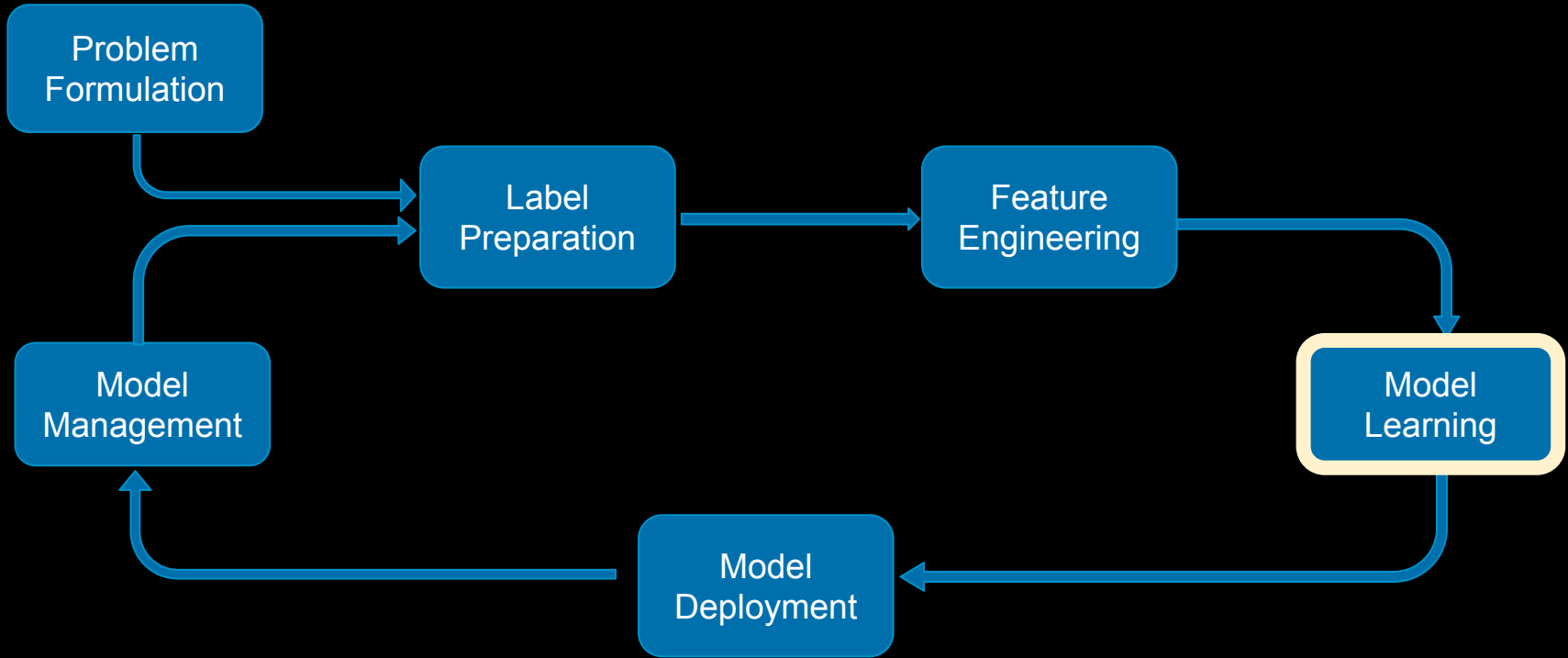


# Feature Engineering - Feature Representation Learning

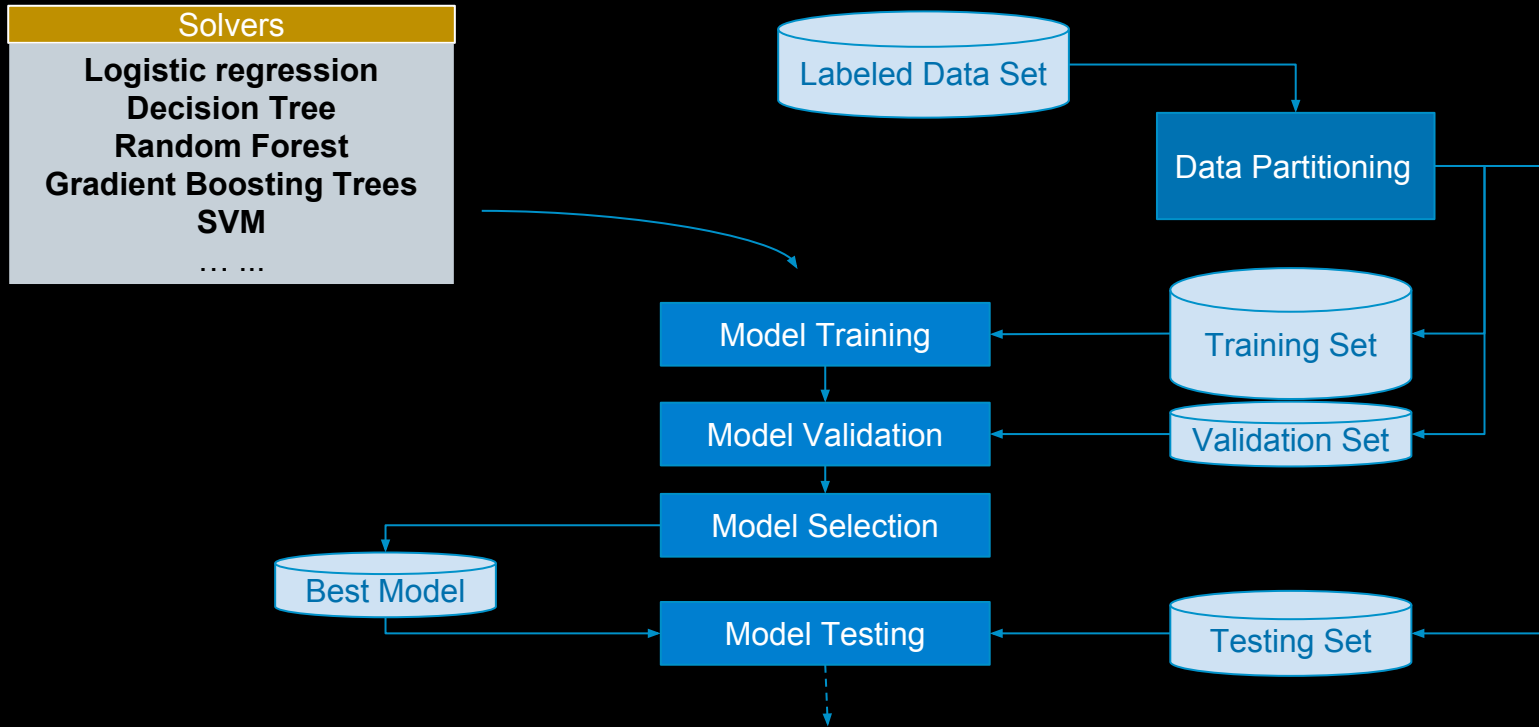
---

# Machine Learning: The Process

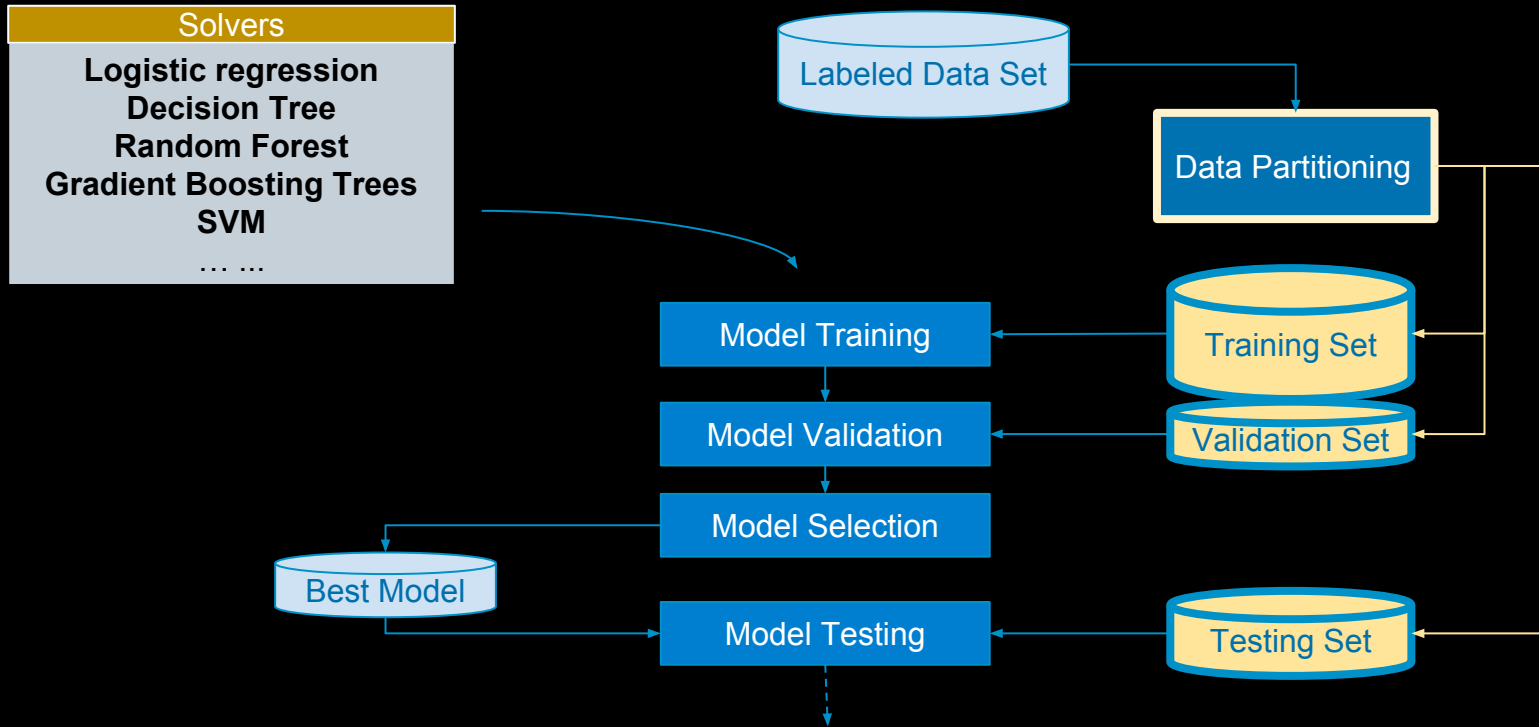
---



# Model Learning



# Model Learning



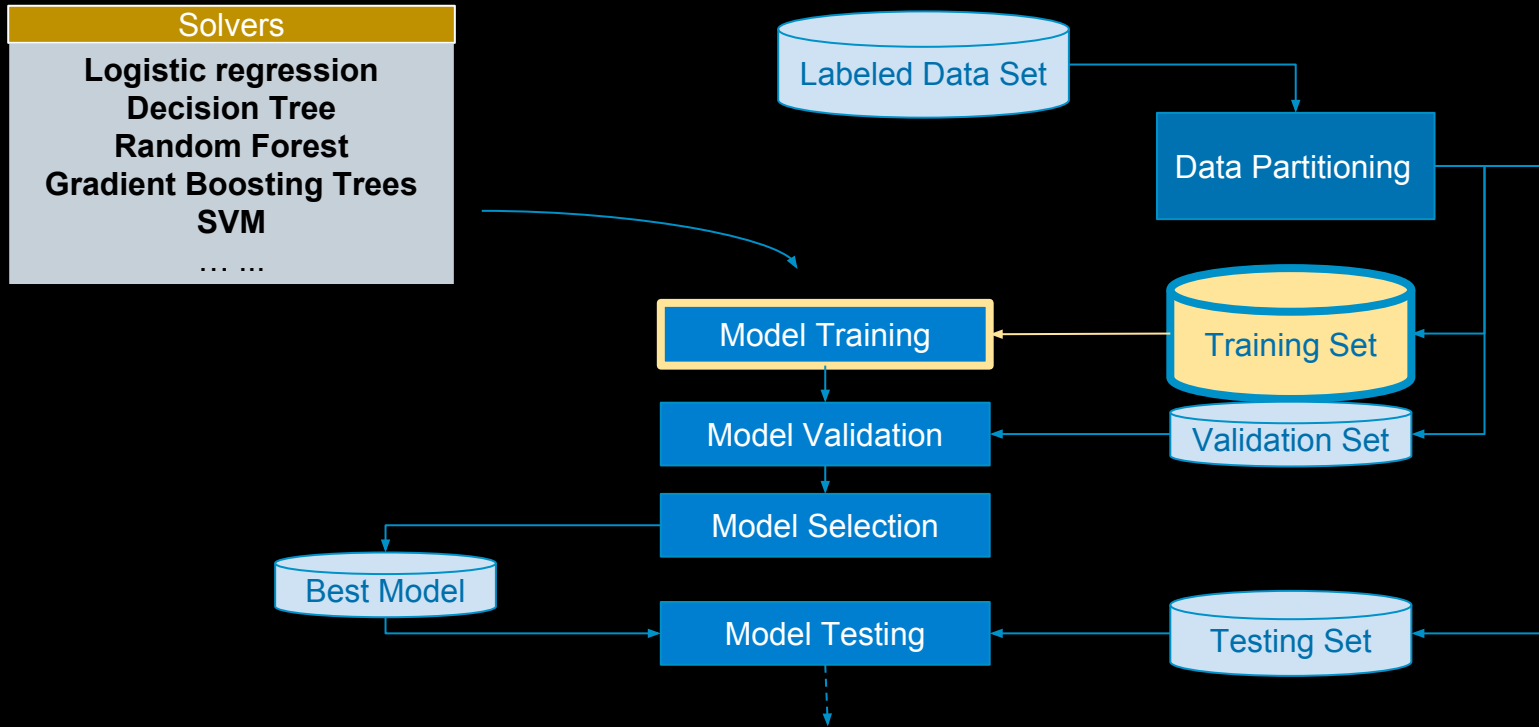
# Model Learning - Data Partitioning

---

## Basics & What to pay attention

1. **Split Ratio** may depend on the size of the data, typically train/validation=test=6:4/7:3
2. **Test data** should be independent
3. Fixing the test set and trying different combination of training and validation set
4. If there is timestamp attached to the data set, **considering the timeline** of the data set
  - a. Forecasting problem: training/validation/test on different time range
5. **Skewed Data**: stratified random sampling balancing over certain features
  - a. Strata: population is partitioned into non-overlapping groups
  - b. When population density varies greatly -> ensure smaller group has the representation
  - c. Often applies to categorical type of dimension: age, gender, geo location, etc
6. Carefully check the data, **avoid data leakage**
  - a. dedup
  - b. e.g. Two opportunities from same company (same time stamp)

# Model Learning



# Model Learning - Solvers



HOW TO CHOOSE A SOLVER:

- TYPE OF PROBLEM
- SYSTEM REQUIREMENT
- PERFORMANCE vs INTERPRETATION

## EASY TO INTERPRET

- Logistic Regression
- Decision Tree

## FAST

- XGBoost
- Logistic Regression
- Linear Regression
- Linear SVM

## PERFORMANCE

- Logistic Regression
- Decision Tree
- Random Forest
- Gradient Boosted Trees
- XGBoost

# Model Learning - Hyperparameter Search

---

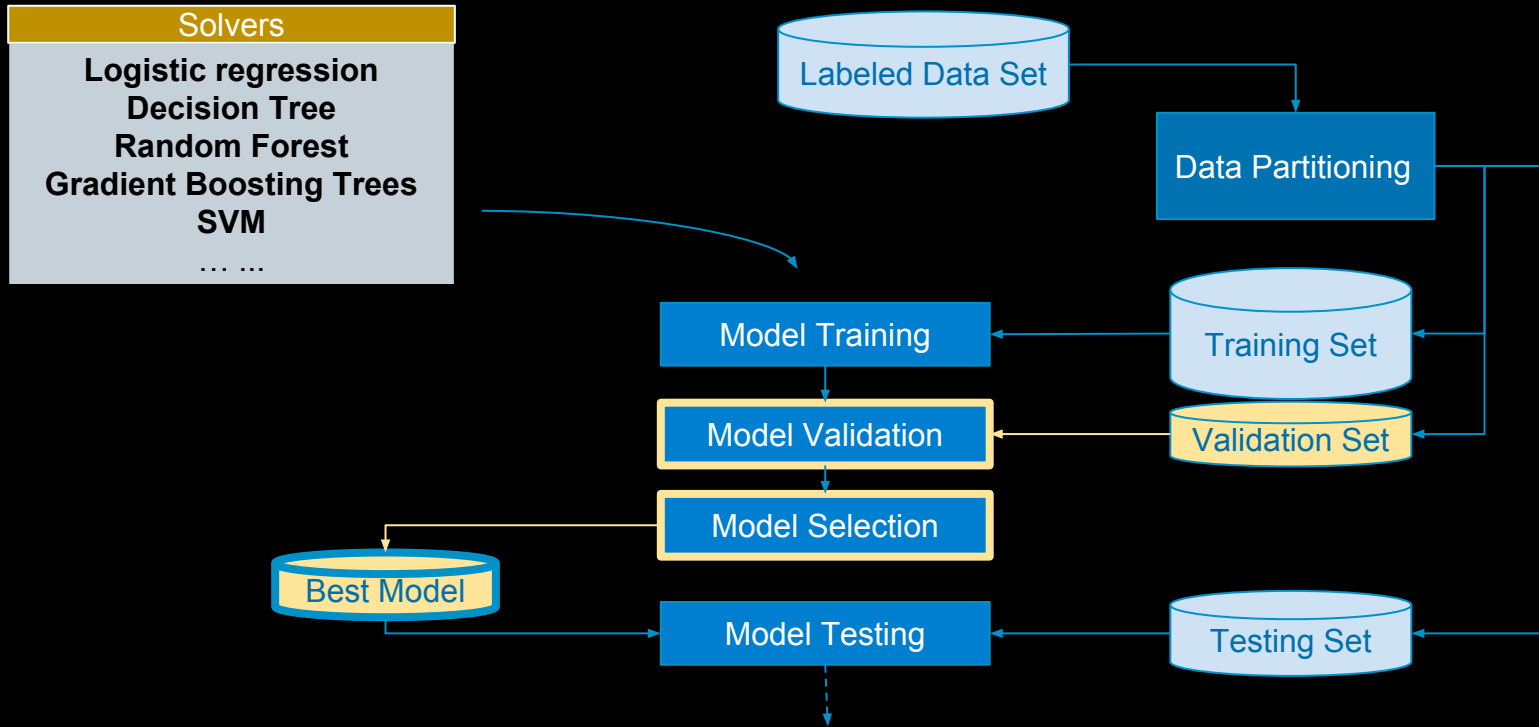
- Goal: optimize performance
- Methods:
  - **Grid search**: most expensive: combination x #cv
  - **Randomized search**:
    - random select combinations on the same space of parameters
    - Faster, may not guaranteed the best, but often good
- Caveat
  - **Overfit**: Avoid using training set, check learning curves, cross-validation
- Tips
  - Not common for grid search on every parameter -> **pick most important ones**
  - Strike the **balance** between the best parameter vs. training time
  - Leverage existing tools/packages, think about parallel, provide wrappers to ease hyperparameter search in early problem exploration
  - Typically don't want to rerun hyperparameter search for each retrain, do only periodically when there is a major change in data volume, features, etc.



[Pic Credit](#)

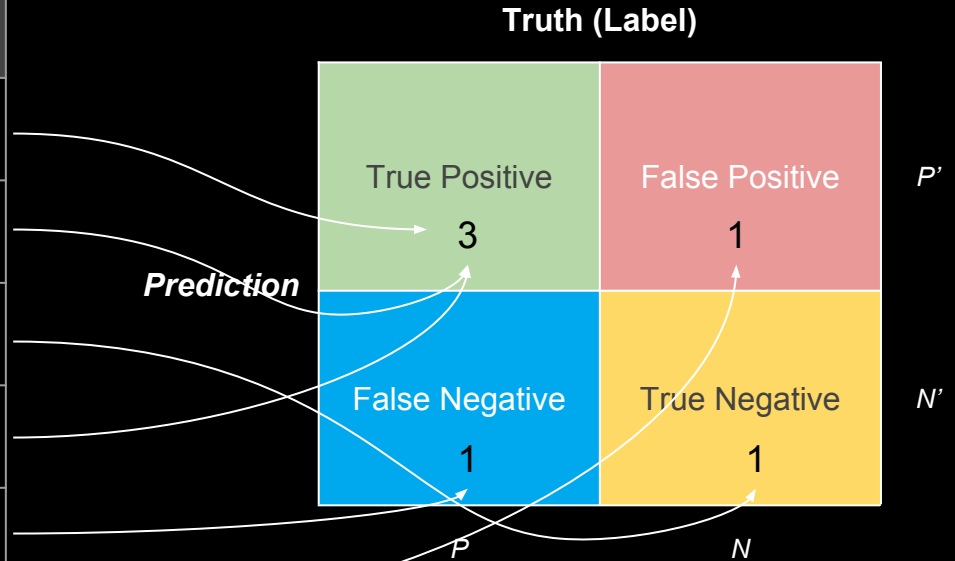


# Model Learning



# Performance Evaluation

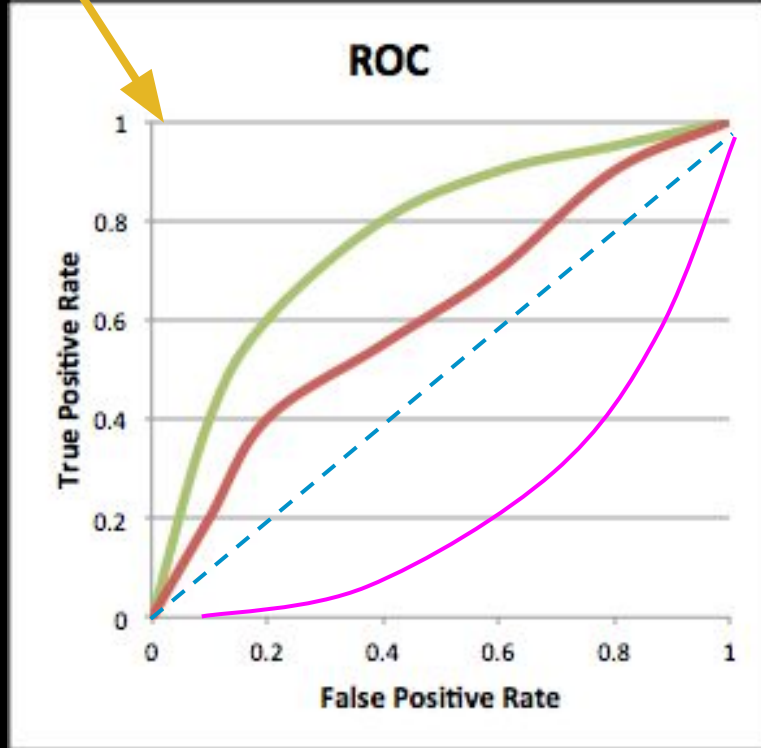
ID	Label	Prediction
1	1	1
2	1	1
3	0	0
4	1	1
5	1	0
6	0	1



consider "1" as positive, "0" as negative

# Performance Evaluation

*perfect prediction*



		Truth (Label)	
Prediction	True Positive	True Positive	False Positive
	False Negative	False Negative	True Negative

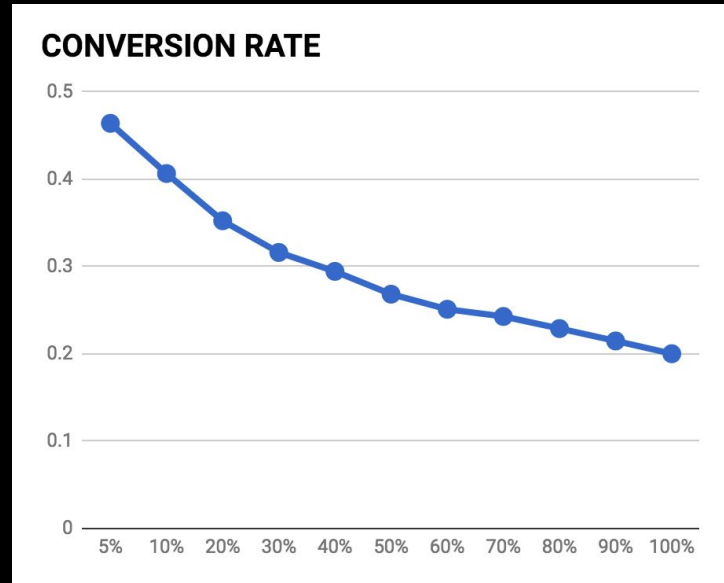
## AUROC:

- Diagonal line: random guess
- Above diagonal line
  - normal prediction
  - Curves close to the *perfect prediction* have a better performance level than the ones close to the baseline.
- Below diagonal line
  - pool prediction
- TPR: true positive rate =  $TP / (TP + FN)$
- FPR: false positive rate =  $FP / (FP + TN)$

# Performance Evaluation



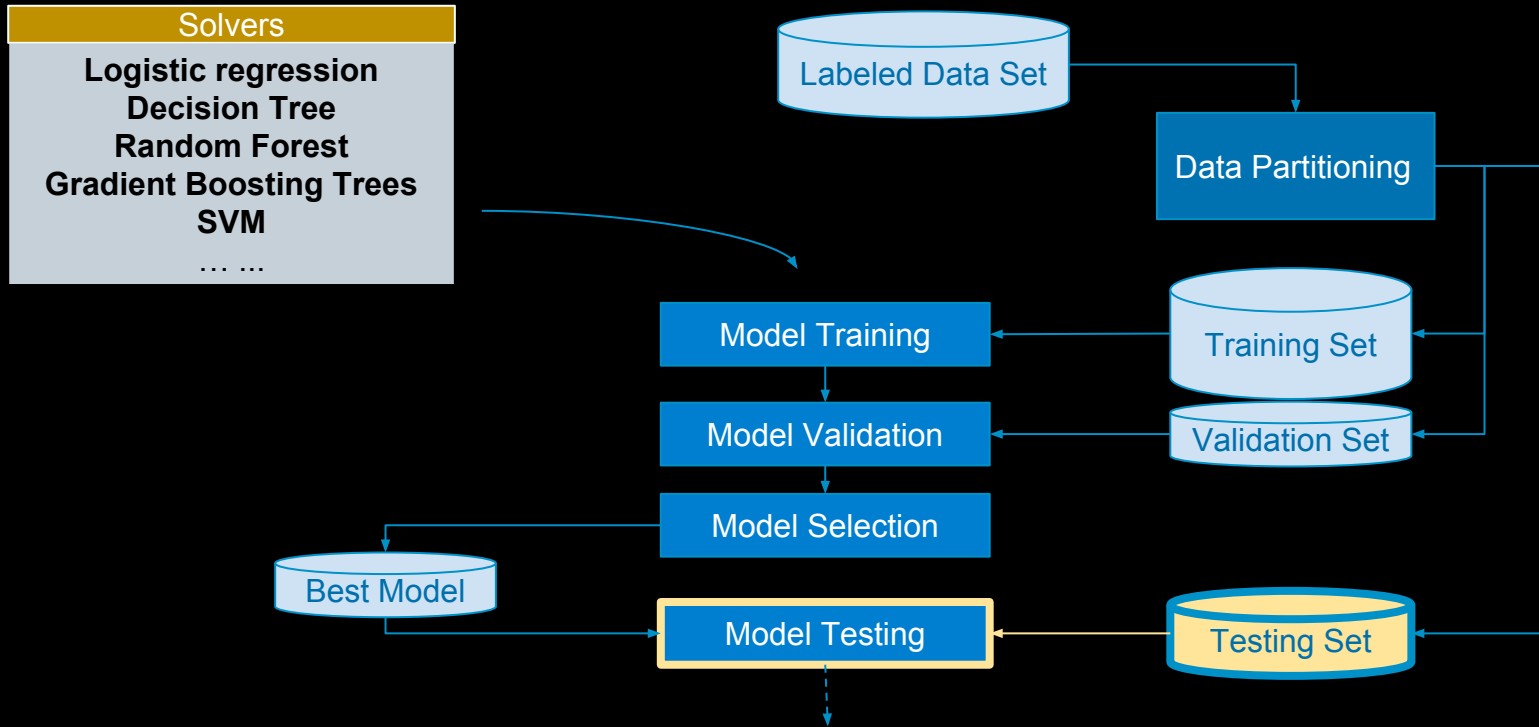
**Rule-of thumb:** Evaluation should always consider actual **Business Metric**



Conversion Rate:

- Rank entities with respect to the probability that they are positive in descending order
- For top N percentage calculate the conversion rates.

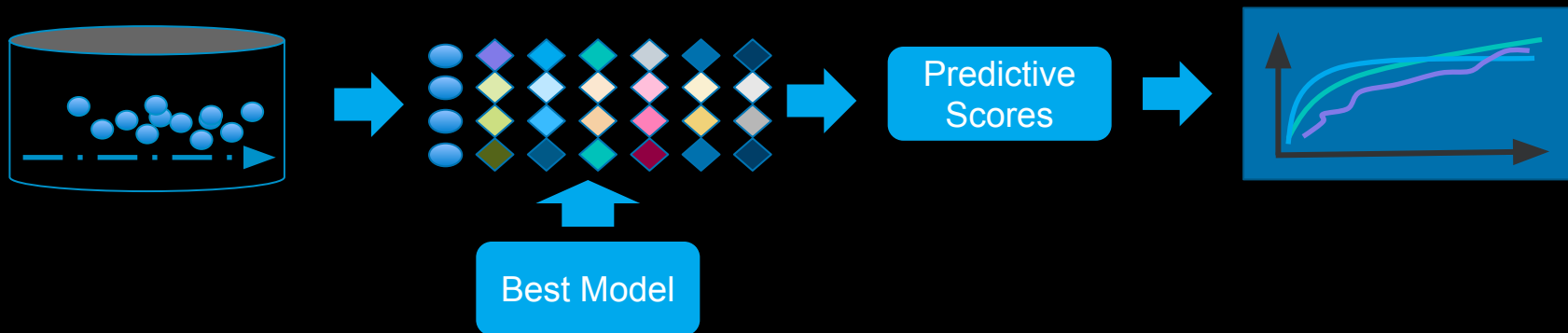
# Model Learning



# Model Learning - Test

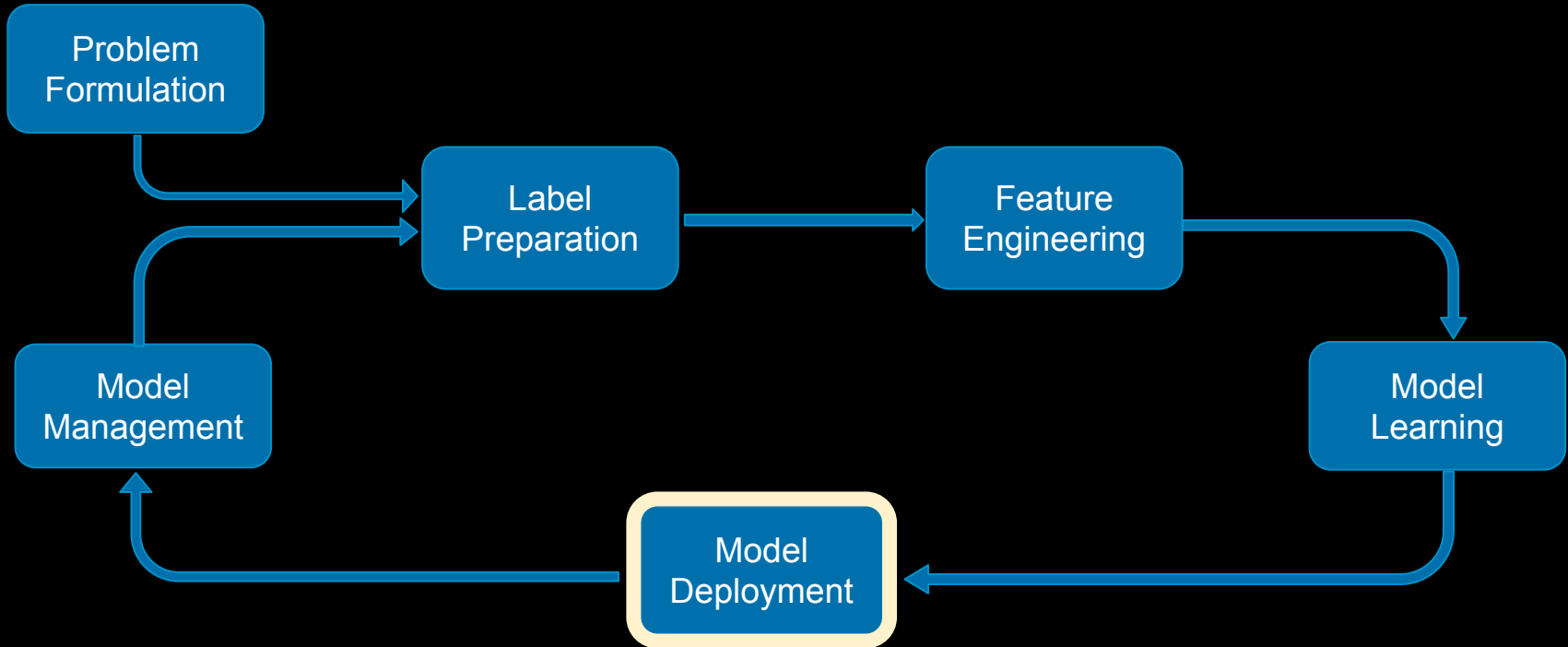
Getting the actual expectation of your model performance

- Test data NOT used in model selection or training.
- Use the best model (chosen by validation set!)
- Performance evaluation/comparison.



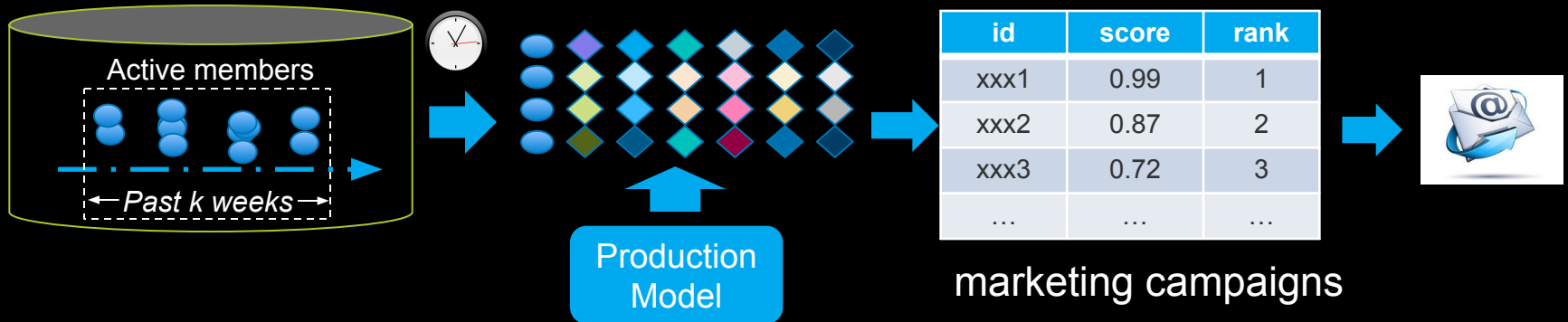
# Machine Learning: The Process

---



# Model Deployment

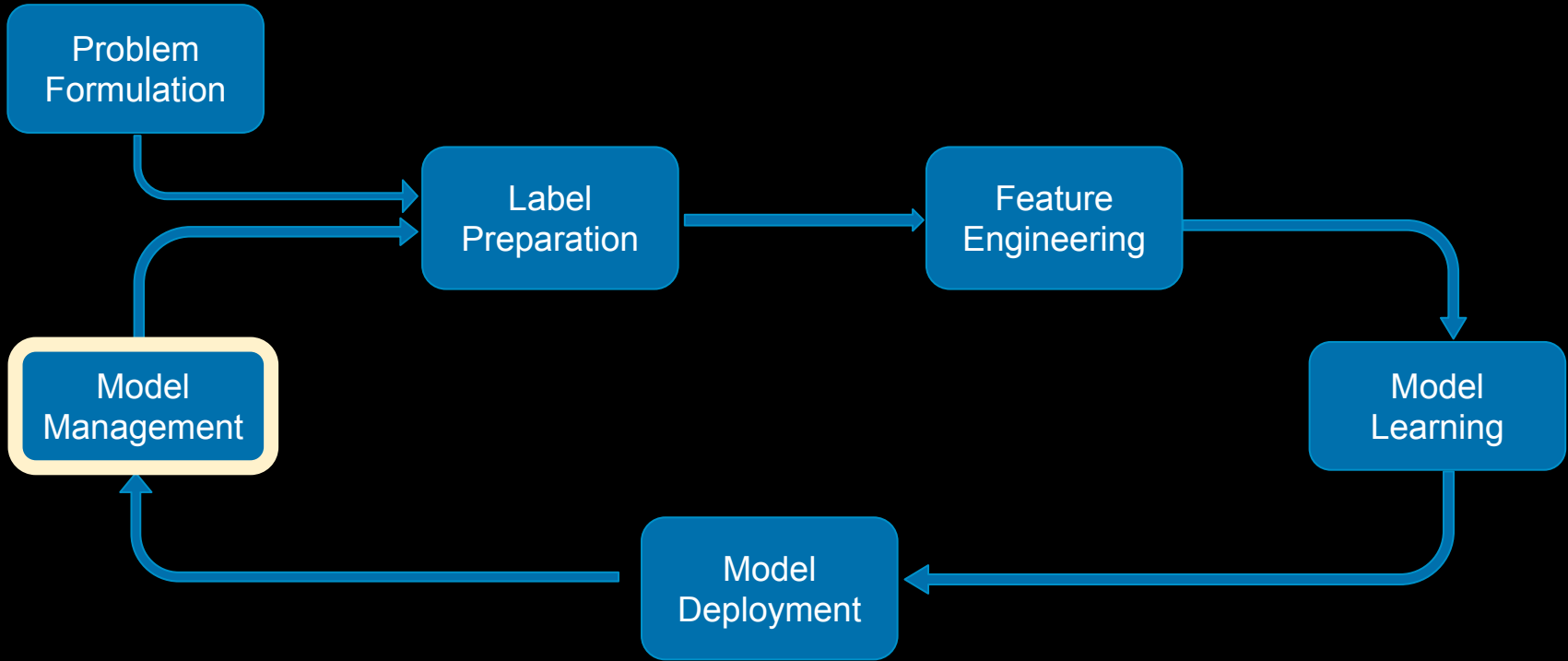
- Schedule and run the scoring pipeline regularly
  - Feature integration and transformation
  - Scoring using the selected model





# Machine Learning: The Process

---



# Model Management

---

## Is Your Fruit Rotting?



# Model Management - Monitoring

- Business customers evolve dynamically
- Products update periodically

- Centralized model repo with standard format
- Monitor both **feature/model** performance changes over time
- Feed in new training data to generate “challenger models” to compete

inherent temporal nature

Monitoring

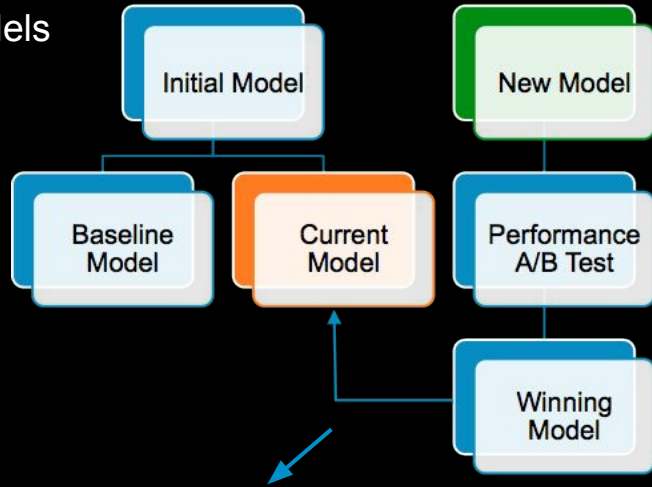


- Performance degradation
- Failure/Outlier examples
- Feature statistics over time:
  - *non-null count, sum, medium*
  - *coefficient of variation for volatility evaluation*

- Model refresh
- Feature diagnosis

# Model Management - Refresh

Ensemble historical models as one of the candidate models



Feed in new training data to generate new model periodically

Monitor changes over time

Assign versions to models built over time

# Performance Measurement via A/B Test



**Hired, Inc.** shared: Sponsored  
Follow

On Hired, companies come to you for your coding skills. Submit 1 app and receive salary/equity offers from multiple companies!

One Application = Many Job Offers  
hired.com

Like • Comment • Share Hiring? Post a job

Check out the latest news about **Coursera** Following 1h

**TOPICA** Two edtech giants team up: Topica announces partnership with Coursera  
e27.co Through this partnership, Vnh University in Vietnam, will recognise credits for blended onlin...

Like • Comment • Share View Company Page

Add a comment...

**Carter Lin** Sr. Associate, Business Analytics at LinkedIn, PhD - We're hiring! 21h

'Drone' hits British Airways plane approaching Heathrow Airport - BBC News  
bbc.co.uk A British Airways plane approaching Heathrow Airport is believed to have been hit by a ...

Like • Comment • Share • Message

Algorithm a  
80%

Algorithm b  
20%

**Geneysys** shared: Sponsored  
Follow

Does your contact center have the 8 critical capabilities recommended by Gartner?  
<https://lnkd.in/bCzEIXg>

Delivering Next Generation Customer Experiences  
GARTNER REPORT: 8 Critical Capabilities for Contact Center Infrastructure  
geneysys.com Find out what they are and which of the top 16 vendors leads the pack!

Like • Comment • Share 3

**Kevin Chang** is now following: 3h

**Sir Alex Ferguson** fluencer  
Director, Ambassador, Visiting Fellow, Author  
Follow

If You Want to Be Successful, You Have to Know What Success Means  
Sir Alex Ferguson on LinkedIn Part of the way you develop excellence in an organisation is to be careful abo...

**Vik Singh** was mentioned in the news 1d

Infer's Self-Service Profiling Revolutionizes Predictive Sales and Marketing  
businesswire.com Infer Inc., a leading predictive sales and marketing platform, today announced updates to its ...

Message Vik • Like • Comment • Share • 5

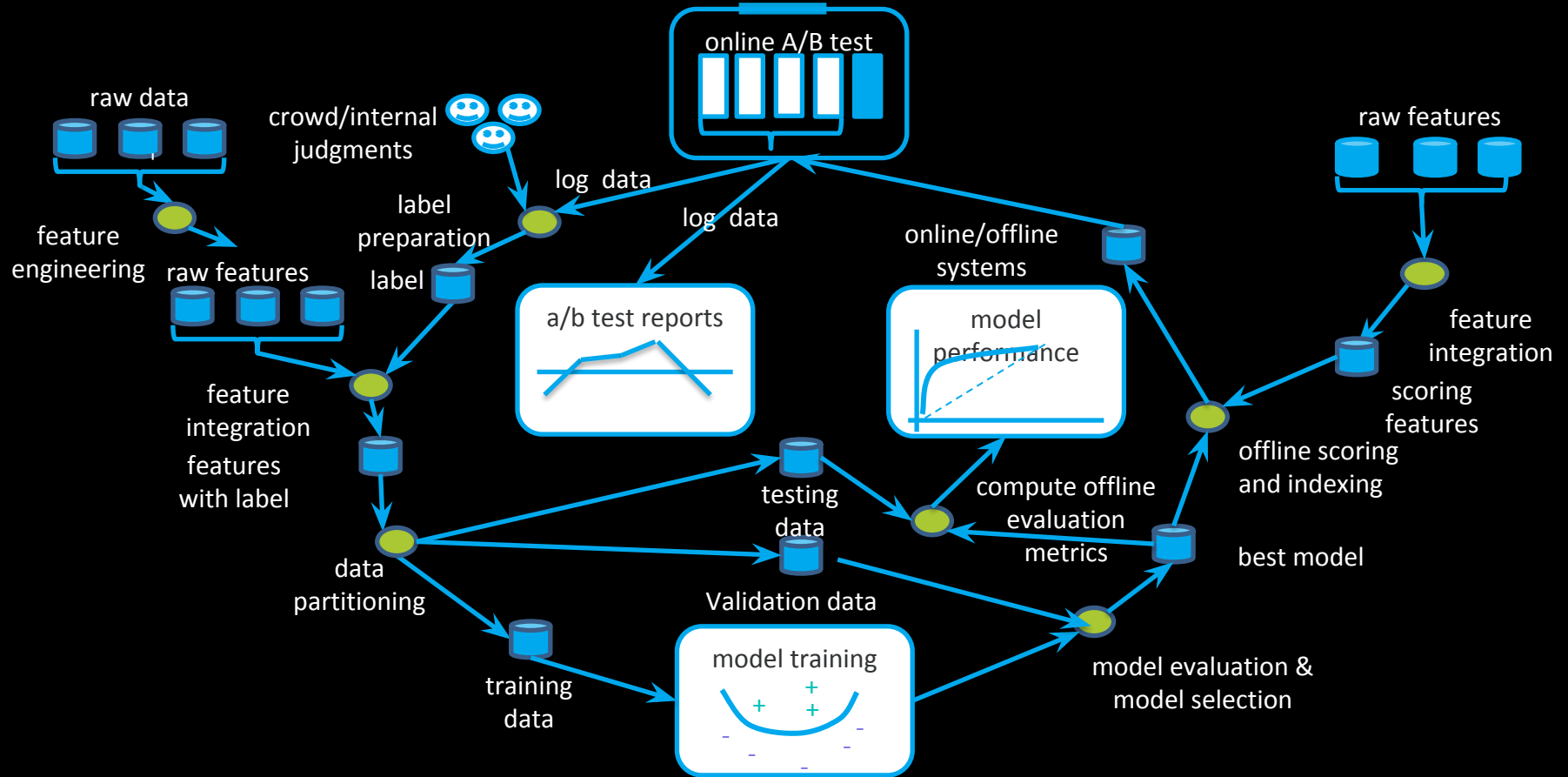
Collect results to determine which one is better

# Best Practices for Running A/B Test

- Start testing on a **small portion** of users
- **Measure one change** at a time
- Be aware of **potential biases** (time, targeted population etc.)
- Avoid coupling a marketing campaign with an A/B test
- Use a **simple rule** of thumb to address **multiple testing problems**
  - 0.05 p-value cutoff for metrics that are expected to be impacted
  - a smaller cutoff, say 0.001, for metrics that are not



# Machine Learning: The Process



# Machine Learning as a Service

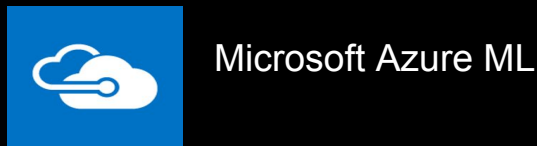
---





# Examples of Machine Learning Platforms

---



# Comparison of Machine Learning Platforms

	Microsoft Azure Machine Learning	Google Cloud Machine Learning Platform (alpha)	Amazon Machine Learning	Databricks	H2O	Turi
<b>Interfaces</b>	API, Notebooks, GUI Workflows, R, python	API, Notebooks, Programmatic Workflows, Java, python (alpha)	API, console	API, Notebooks, Programmatic Workflows, Java, Scala, python, R	API, Notebook, R, Java, Scala, python	Notebooks, C++, python
<b>Cloud</b>	Microsoft Azure	Google Cloud Compute Engine	Amazon AWS	Amazon AWS	Amazon AWS, Microsoft Azure*, Google Cloud*, your data center	Amazon AWS, Microsoft Azure*, Google Cloud*, your data center
<b>Data Sources</b>	Hive, Azure blob storage, Azure table, Azure SQL OData feed, bulk upload, URL download	text (including JSON), Google Cloud Bigtable, Google Cloud Datastore, byo	Amazon S3, Amazon Redshift, Amazon RDS (SQL)	Amazon S3, Amazon Redshift, mongoDB, mySQL, Shark, Hive, HDFS, byo, ...	HDFS, file, URL download, Amazon S3	HDFS, Amazon S3, file, ODBC, Avro, SparkRDD, ...
<b>Modeling Techniques</b>	extensive regression & classification, k-means, Vowpal Wabbit	TensorFlow: flexible deep learning, regression, ...	binary, multiclass classification, regression	MLlib: linear, logistic, tree (RF, GBDT), MF, survival regression, multiclass, k-means, LDA	extensive regression & classification, GBDT, ensembles, deep learning	extensive regression & classification, recommenders (factorization machines, MF, ...), k-means, LDA, deep learning
<b>Limitations</b>	10GB training datasets <a href="#">full info</a>	ML in limited preview Prediciton API: 2.5GB training datasets	100GB training datasets 1k input features <a href="#">full info</a>			
<b>Extensibility</b>	byo R, python, package imports	TensorFlow SDK, Google Cloud Platform	byo AWS services	pipelines, byo Spark	byo on your platform	python, C++, byo platform
<b>Pretrained Models</b>	vision, speech, sentiment, ...	speech, vision, translation, ...				
<b>Scoring</b>	API, batch	API, batch, planned export	API, batch	API, batch, streaming, PMML export	API, JSON, POJO export	API, Batch

# Machine Learning Platform - Best for YOU?



- **Understand your problem:** scale, scope, technique needed, complexity, etc
- **Evaluate time/resources/cost:** tight timeline, limited budget, start-up?
- **Evaluate candidate framework:** techniques, support, limits, etc
- **Integration with your own tech stack:** data formats or sources, model deployment

# Leverage Vendor Solutions

---

- Leverage your network
  - Who is using it? in production? Is it working for them?
  - What are its limits? scalability, stability, ease of use, ....
- Open source
  - How active is the development of the project?
    - sometimes means lots of changes or instability
    - usually a good sign
  - From what companies?
    - Diverse companies suggests broad adoption
    - Big companies suggests scale, maturity of engineering processes
    - Academic only may indicate weak points in scale or robustness (especially with respect to scoring solutions)
- Purchasing solutions or support
  - Consider size of the company, amount of funding raised, your tolerance to risk
  - Pedigree of the ML experts and engineers at the company
- Anticipate evolvement

# Anticipate Evolving Needs

---

- May start with buying a solution
- But later need to integrate more tightly and move toward assembling
- Or later build major components yourself
- Consider whether the solutions you buy are built on open source and can help you transition to assembling solutions



# Vendor Solution vs Inhouse Platform

---

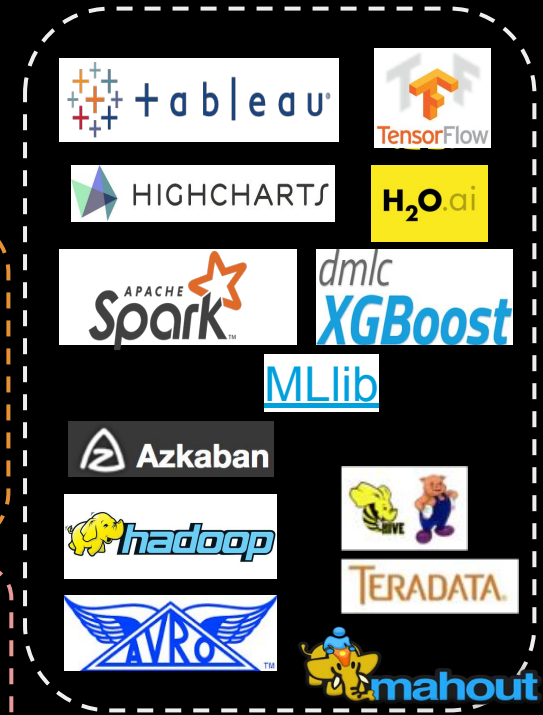
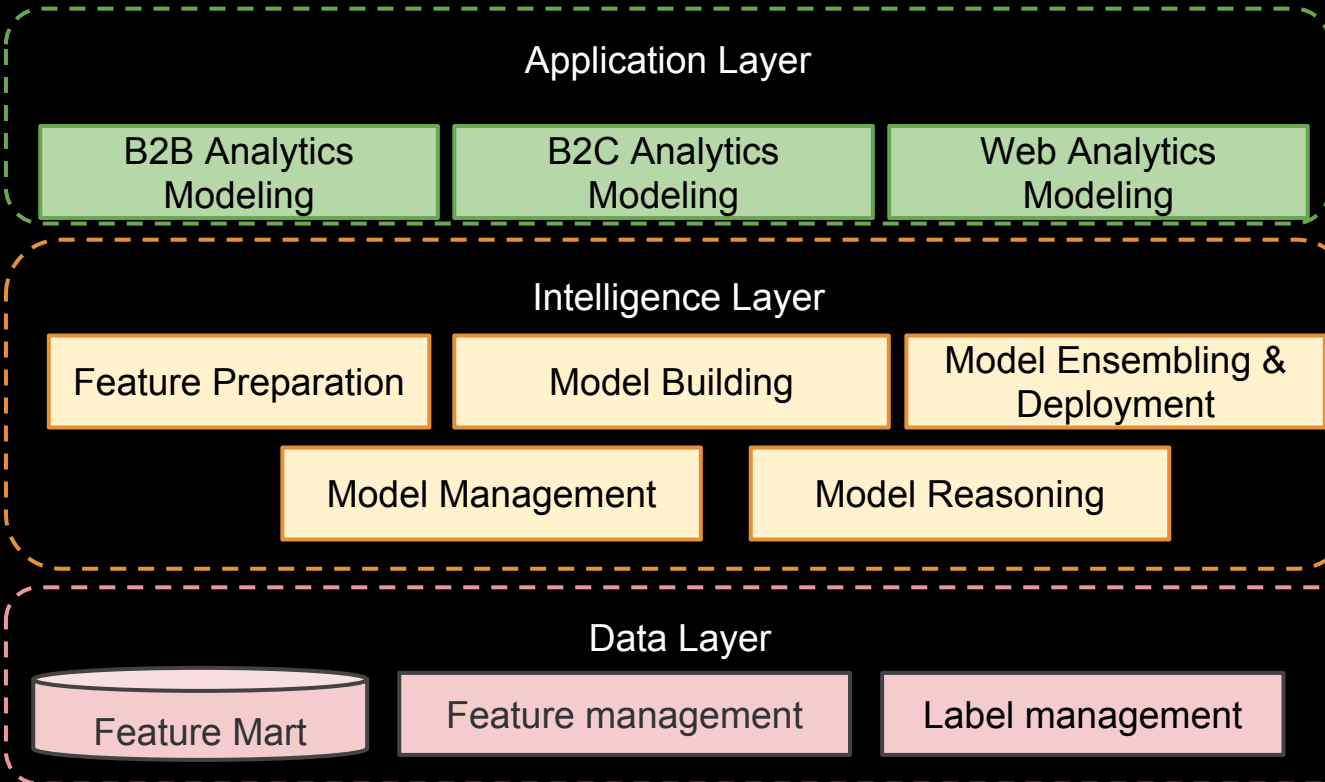
## Vendor Solution

- Might be **costly**
- Deploy “models”, as services, not an end to end system
- **Optimizing just algorithms**
- **Latency** in network
- **Data security**
- Limited in terms of techniques
- Need to **fill in gaps** that platform doesn't address
- How much will the integrated layer delegate?

## Inhouse Platform (Build or Assemble)

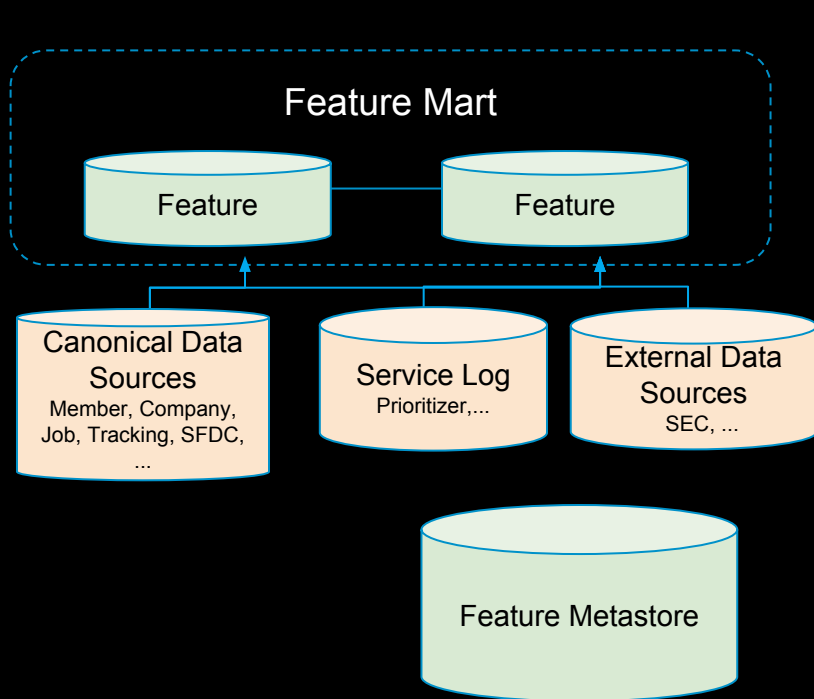
- Deploy **end-to-end system** and have **full control** on it
- **Optimizing the whole pipeline**, not just algorithms
- **Flexibility** of the techniques
- Scalable and fast iteration
- Allows **specialization** and **innovation**
- Control and develop deep expertise in the whole stack
- **Security**

# Example of Building/Assembling a Platform



# Data Layer - Basics: what to build

Consider complexity and unification of Feature mart/metastore to enable flexibility on feature integration and feature engineering



Feature mart contains data:

- from different sources
- structured vs. unstructured
- with different entities
- as snapshot or aggregational
- with different granularity
- with different privacy
- with different quality

Feature metastore:

- support feature governance and feature application
- has feature search and feature profiling



# Data Layer - Manage: what to provide

---

As a user **searching** for features, he can..

- Search: use faceted/generic search to narrow down the space
- View: detailed view for the features
- Discover/Recommend: can discover similar/useful features

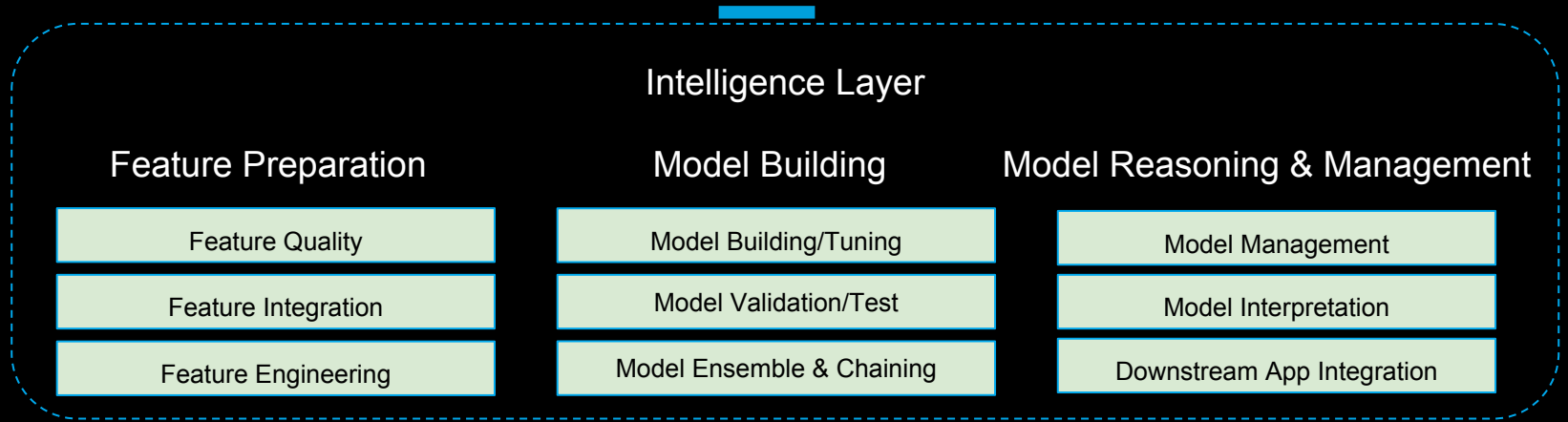
As a user trying to **understand a feature**, he can...

- Know the basic: owners, feature logic, lineage, tier of feature (gold, silver, experimental), other meta information
- Know the quality: health, coverage, change log
- Know the value: whether it has shown values in other similar type of modeling

As a user **consuming a feature**, he can..

- How: standardize the format for easy consume
- Monitored: get email communication from feature owner
- Contribute: add comments/tag for the feature to share with community

# Intelligence Layer - One stop shopping



- **On Feature:**
  - **Adapt feature variation:** different **granularities**, user **adhoc** feature, etc
  - **Support basics:** various common feature transform and UDF capability
- **On Learning:**
  - **Coverage:** general learning problems, strategies, best practice
  - **Flexibility:** user can choose and configure settings
  - **Simple:** Be “client oriented”, hide complexity, provide simplicity
  - **Specialty:** modules specific to business needs, e.g. model interpretation

# Application Layer

---

- This is **REAL TEST**
- For User:
  - Provide template configurations
  - Provide example solutions
  - Map **“model output” -> actionable insights**
- As Owner:
  - **Communicate** results & follow-up
  - Dive deep and **aggregate knowledge** cross aspects
  - Collect **feedback** & analysis usage
  - Create **ecosystem**/ user community
  - Explore & **push the limit**

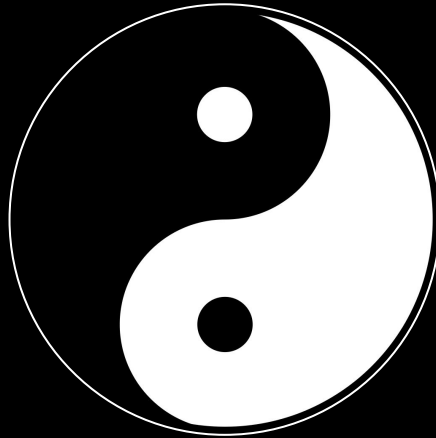


# Machine Learning Platforms - Take Away

---

*Simplicity is the ultimate sophistication.*

*Leonardo da Vinci*



# OUTLINE

---

## Introduction

- ❑ Introduction to Data Analytics and Data Science @LinkedIn

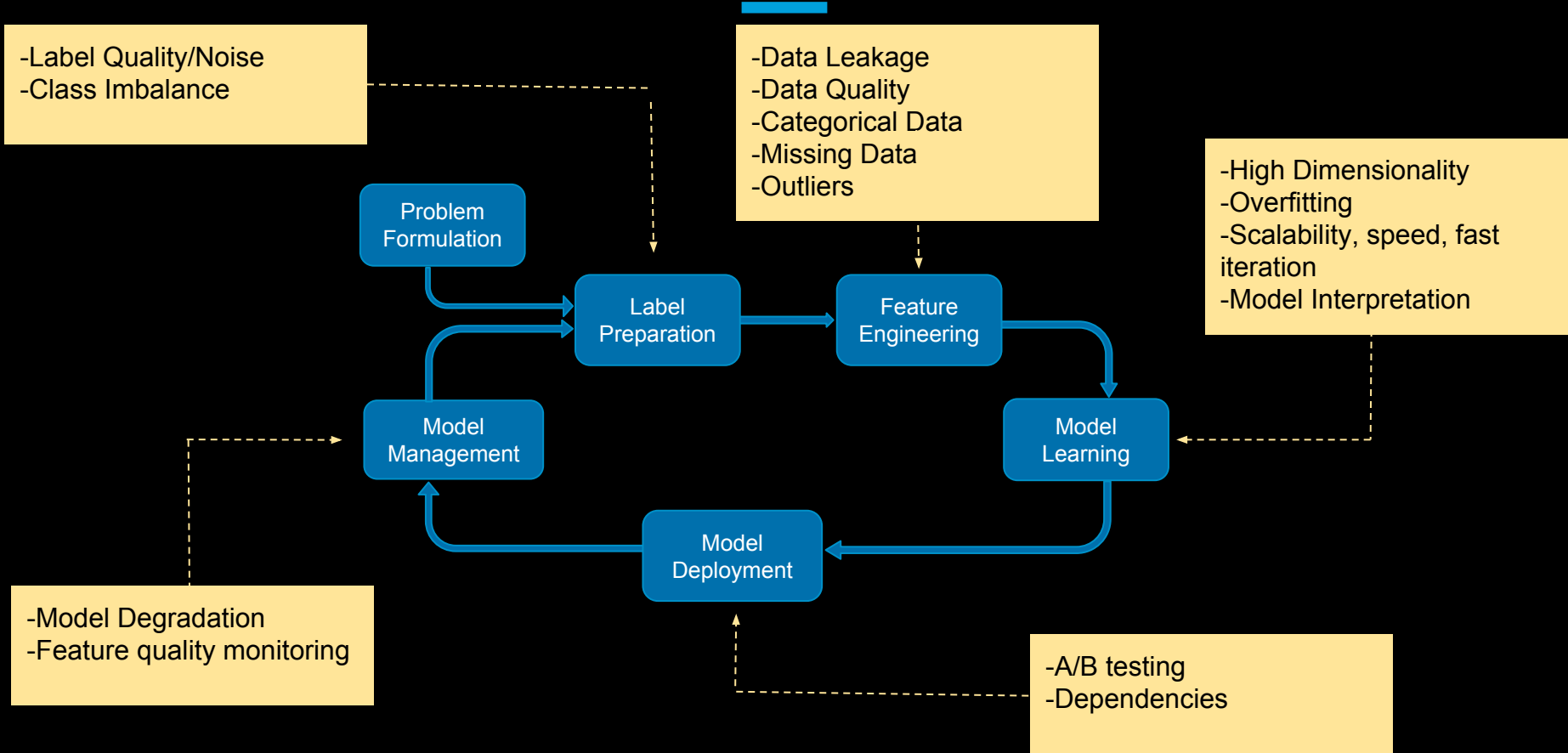
## Big Data Analytics

- ❑ Common practices on metric development
- ❑ How can we develop a better product experience?

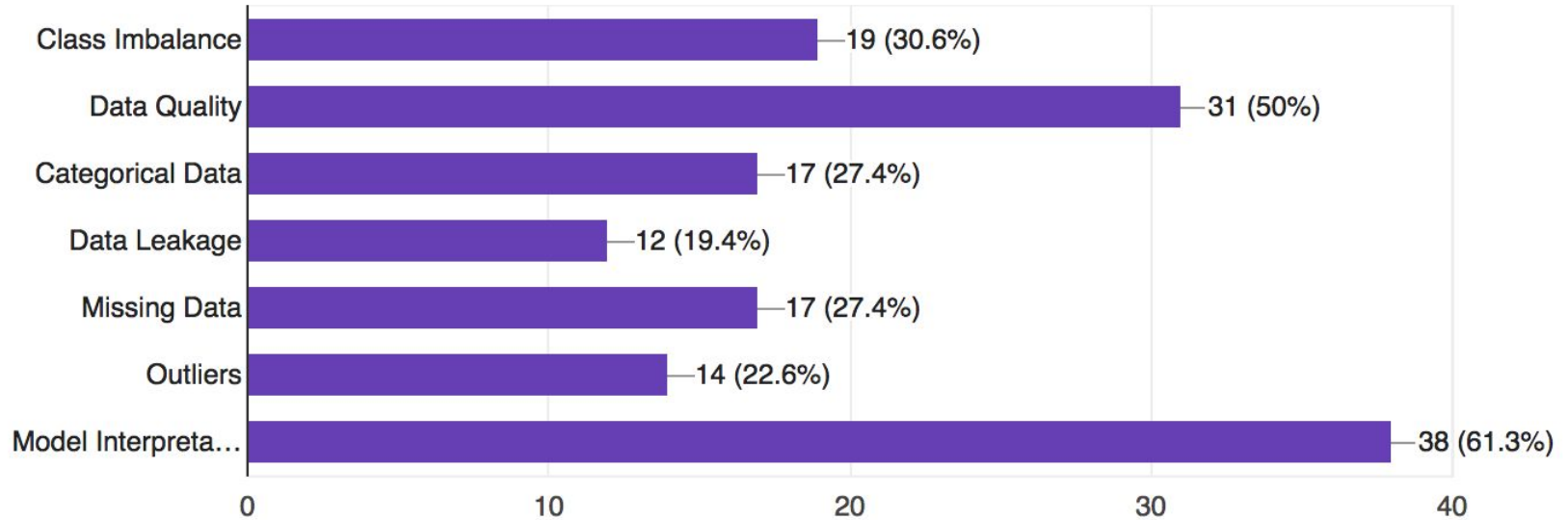
## Machine Learning

- ❑ Why Machine Learning is important?
- ❑ End-to-end walkthrough of a production modeling solution
- ❑ **Common pitfalls and challenges**
- ❑ Case Study - B2B Modeling

# Common Pitfalls and Challenges



# Survey Results



# Common Pitfalls and Challenges

---

## MODEL INTERPRETATION





# Model Interpretation

---

- Why do we need model interpretation?
  - Debug, diagnose, generate new hypotheses
  - Inevitable questions about why a prediction was made from your business counterparts
  - For presenting reasons to users - may be output of a ML model itself

# Model Interpretation

---

- Why is model interpretation a challenge?
  - **Univariate Feature Interpretation:** Feature importance coming from the ML model.

## Pros:

- Get a sense of importance for each feature
- Many available algorithms: Random Forest, Regularized linear models, various feature selection algorithms

## Cons:

- Bias, e.g. impurity evaluation of RF is biased towards preferring variables with more categories
- Difficulty of interpreting the ranking of correlated variables
- Single feature may contain lots of noise.

Can we trust solely on feature importance of model?



# Model Interpretation

---

## Solution:

- **Group-wise feature interpretation:** cluster features into buckets which have semantic meaning, then build models using only the subset of the features within each bucket.

## Pros:

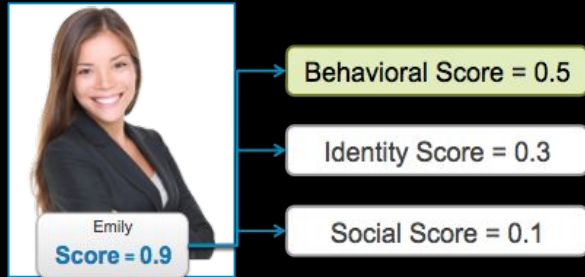
- Easy to capture the overall look by grouping massive features into a few buckets
- Strong semantic meanings
- Inter-group correlation are less
- Impact from noise is reduced by analysing multiple features at the same time

## Cons:

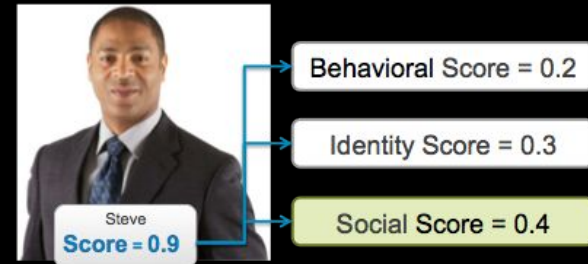
- Domain knowledge required

# Model Interpretation

Group-wise feature interpretation:



...With premium account, you *get more search result* and access to...



... Do you know *5 of your connections* have started to use premium account ...

# Common Pitfalls and Challenges

---

## DATA QUALITY



# Data Quality Issues

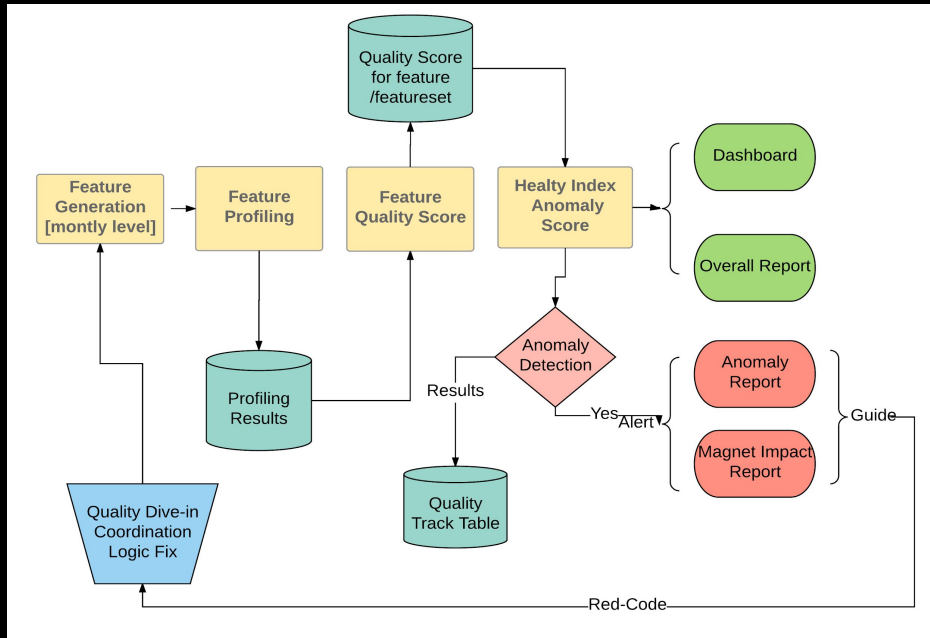
---

Why might we have poor quality data?

- Incomplete, sparse, noisy and dynamic over time
- Missing historical data
- Lack of centralized data covering various needs
- Unclear source of truth
- Manual entry of data



# Data Quality Issues



## Solution:

- Quality Monitoring flow that generates insights for the data quality for each week/month/year
  - Discover potential issues ahead of time
- Revisit data logic periodically
- Alert for downstream users for modeling impact



# Common Pitfalls and Challenges

---

## CLASS IMBALANCE

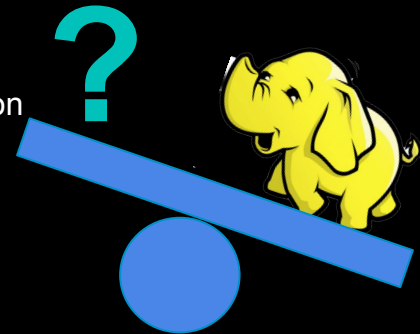




# Class Imbalance

---

- A dataset is said to be **imbalanced** when the binomial or multinomial response variable has one or more classes that are underrepresented in the training data, with respect to the other classes
  - *“I have a binary classification problem and the label is distributed in 1:100 ratio in my training set. My results are overfit to majority class.”*
- The class imbalance problem is pervasive and ubiquitous
  - e.g job recommendation, ads CTR, fraud detection
- Misclassify the minority class usually with high cost
  - rejecting a valid credit card transaction VS. approving a large fraudulent transaction



# Class Imbalance

---

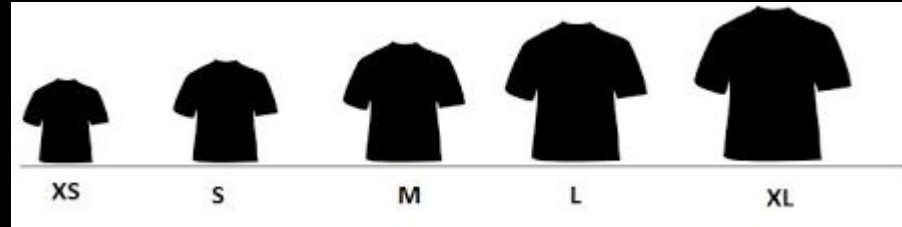
## Solutions:

- Can You Collect More Data?
- Consider different evaluation metrics
  - “Accuracy” might be misleading for imbalanced training data
  - Confusion Matrix, Precision, Recall, F1, ROC
- Re-sampling data set
  - Up-sampling (Over-sampling)
  - Downsampling
  - Synthetic Minority Oversampling Technique (SMOTE)
- Cost-Sensitive Training

# Common Pitfalls and Challenges

---

## CATEGORICAL DATA



# Categorical Data

---

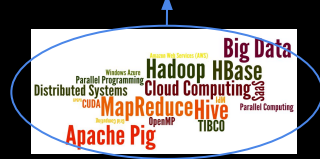
- Categorical feature
  - A variable that can take on one of a limited, and usually fixed, number of possible values, thus assigning each individual to a particular group or "category" - [The Practice of Statistics , 2003](#)
    - Gender = {male, female}
- High cardinality categorical features are common in the data
  - E.g. Industry, country, city
- Too many levels
  - Not all the levels (distinct values of the categorical feature) got enough support. Some are less useful
  - Many Machine Learning tools can only handle certain amount of levels
    - E.g. Random Forest implementation in R has a hard limit of 32-levels for a categorical feature

# Categorical Data

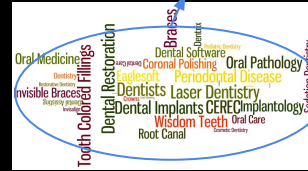
- Solutions

- Reduce number of levels by grouping categories into higher-level ones

Cloud and Distributed Computing



Dental



- Transform a categorical feature into multiple binary ones
  - Introduce an additional 'others' feature to represent all the new categories in the testing set



Hadoop	Spark	Hive	Yarn	...	others
1	1	0	0	...	0

# Common Pitfalls and Challenges

---

## MISSING DATA



# Missing Data

---



- Missing data scenarios
  - Missing Completely at Random (MCAR)
    - Is not related to other variables AND is not related to value of missing variable
    - *E.g. Computer crash*
  - Missing at Random (MAR)
    - Is related to other known variables. BUT, is not related to value of missing variable(s) once we take the above relation(s) into account
    - *E.g. male participants are more likely to refuse to fill out the depression survey, but it does not depend on the level of their depression*
  - Missing Not at Random (MNAR)
    - Is related to what the value of the missing data would have been even if we take into consideration other variables
    - *E.g. People with low high school GPA decline to report it*

# Missing Data

---



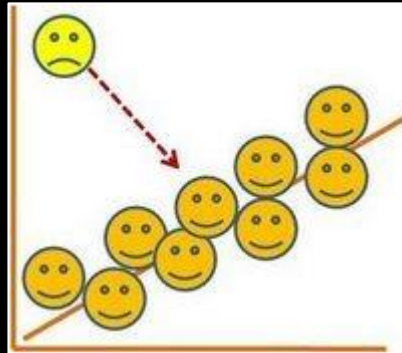
- Solutions:
  - Remove observations with missing values
    - When values are missing at random and you have enough data
  - Missing data imputation
    - Common imputation strategies
      - Categorical: Choose the category with the most support
      - Numerical: Median, mean, or simply set to 0
    - Predict missing data using a model
  - Introduce a corresponding dummy feature to indicate its availability
  - Use a model that is robust to missing data, e.g. tree-based model.



# Common Pitfalls and Challenges

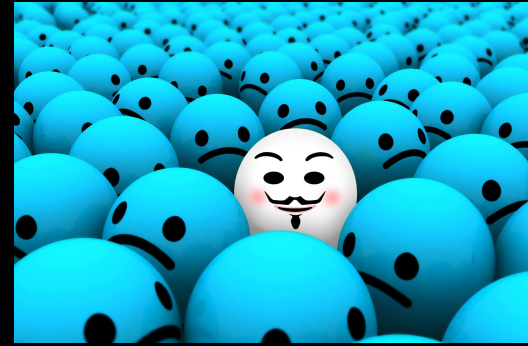
---

## OUTLIERS



# Outliers

---



- Outliers can be introduced in response or predictors
  - Rare event (valid)
  - Erroneous metrics (invalid)
- Impact of outliers
  - Outlier values can have a disproportionate weight on the model.
  - MSE will focus on handling outlier observations more to reduce square error
  - Boosting will spend considerable modeling effort fitting these observations

# Outliers

---

- Solutions:
  - Whether the outlier value is valid or invalid?
  - Remove observation with outlier feature values
  - Apply transformation to reduce impact:
    - log
    - square root
    - binning (e.g. based on distribution)
  - Impose a constraint on data range (cap values)
  - Choose a more robust loss function (e.g. MAE v.s. MSE)
  - Use a model that is robust to missing data, e.g. tree-based model

# Common Pitfalls and Challenges

---

## DATA LEAKAGE

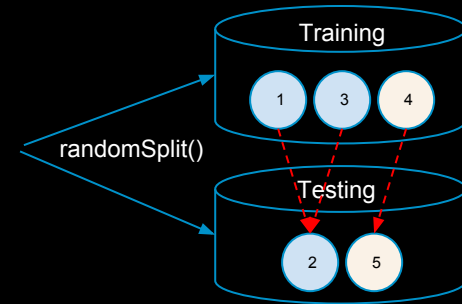


# Data Leakage

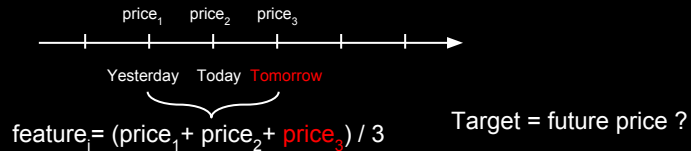
Common leakage scenarios:

- Leaking data from the training set into the testing set

ID	Entity	Time	Label	Usage 1	Usage 2	Region	Size	Spending	Activity 1	Activity 2	...
1	Company 1	01/01	1	10	2	SF	118	3,810	60	4	...
2	Company 1	02/01	1	5	0	SF	117	3,810	70	5	...
3	Company 1	03/01	1	0	3	SF	119	3,810	120	30	...
4	Company 2	01/01	0	100	30	NYC	50	2,000	23	0	...
5	Company 2	04/01	0	90	43	NYC	50	2,000	23	0	...
...	...	...	...	...	...	...	...	...	...	...	...



- Leaking of information from the future into the past



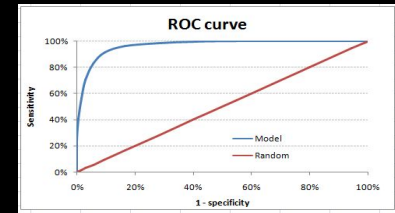
Feature = "Rep id"    Target: customer churn

A specific sales representative was assigned to take over an churned account

# Data Leakage

Main solution is to identify the data leakage:

- Exploratory data analysis (EDA)
  - An approach to analyzing data sets to summarize their main characteristics, often with visual methods
- Is model performance too good to be true?
- Early in-the-field testing



# OUTLINE

---

## Introduction

- ❑ Introduction to Data Analytics and Data Science @LinkedIn

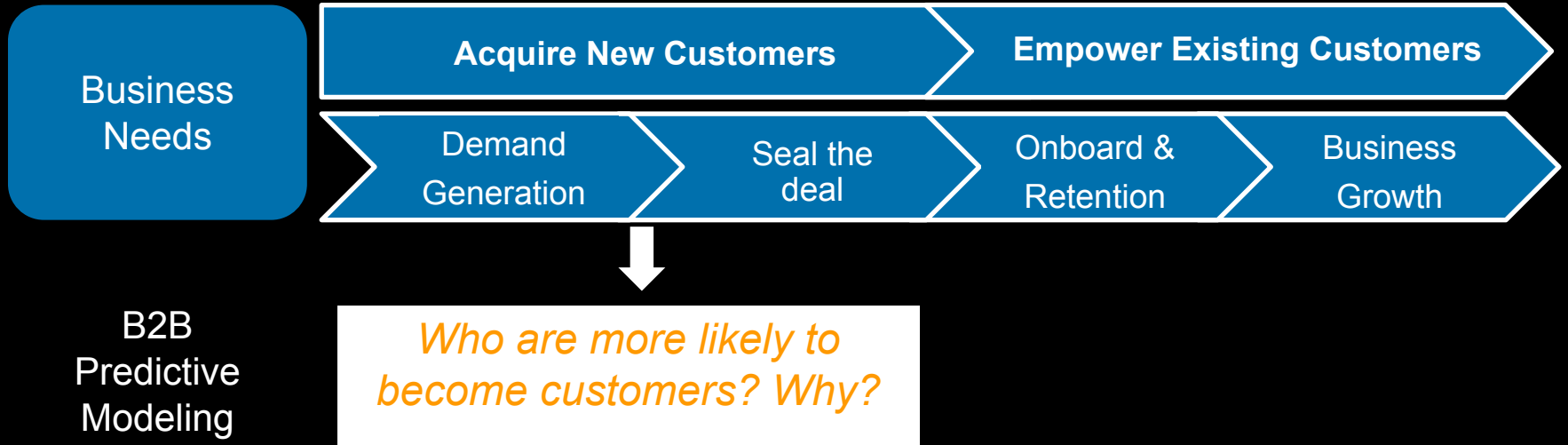
## Big Data Analytics

- ❑ Common practices on metric development
- ❑ How can we develop a better product experience?

## Machine Learning

- ❑ Why Machine Learning is important?
- ❑ End-to-end walkthrough of a production modeling solution
- ❑ Common pitfalls and challenges
- ❑ Case Study - B2B Modeling

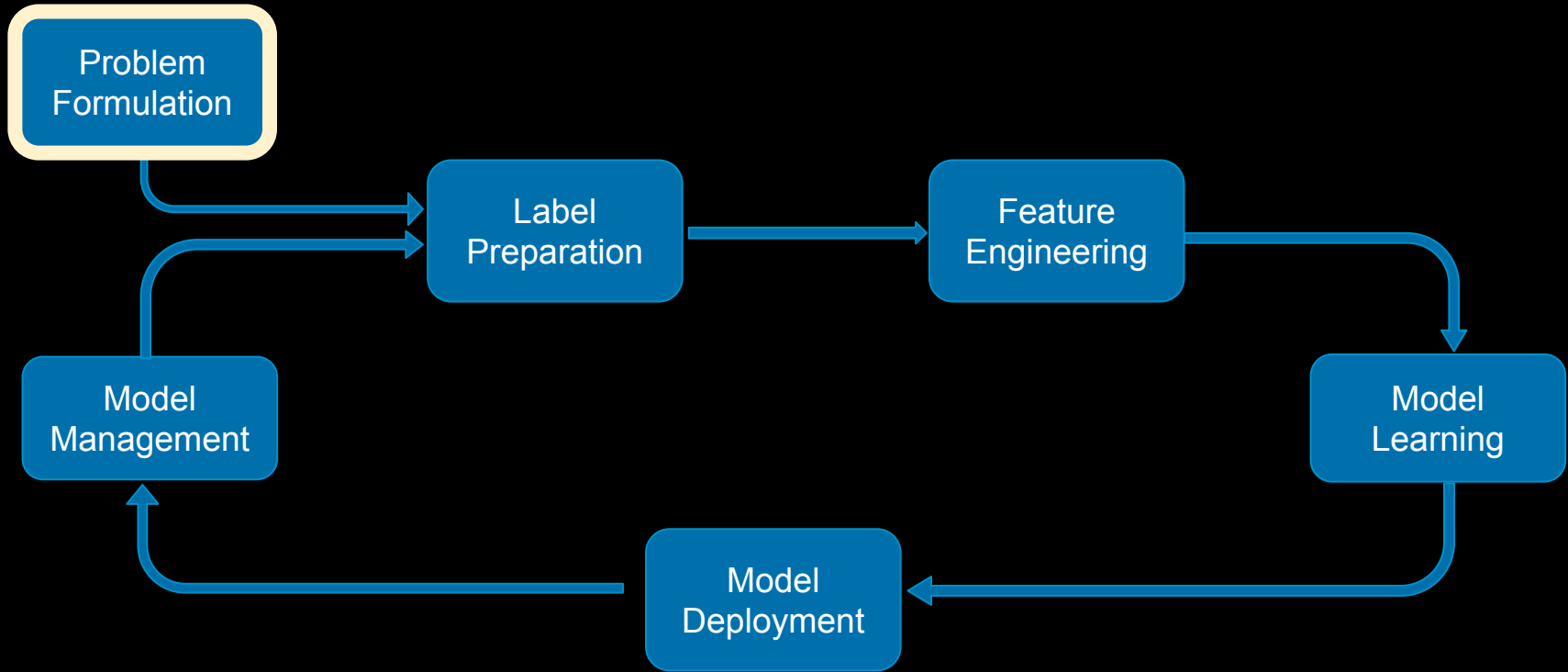
# Case Study - B2B Modeling



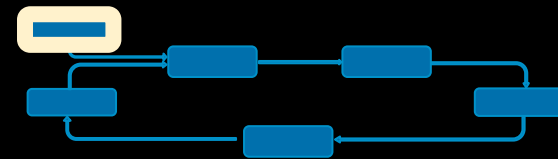


# Case Study - B2B Modeling

---



# Problem Formulation

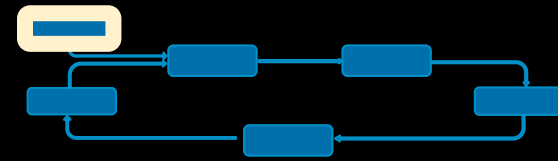


Which enterprise accounts are most likely to buy the product & why?

## Challenges of This Problem:

- Business varies significantly across region.
- Region level, regions that are very small need to borrow information from other regions.
- Data evolves dynamically, time series events.
- Data is sparse and noisy.
- Score accuracy is important for the whole spectrum.

# Problem Formulation



Which enterprise accounts are most likely to buy the product & why?

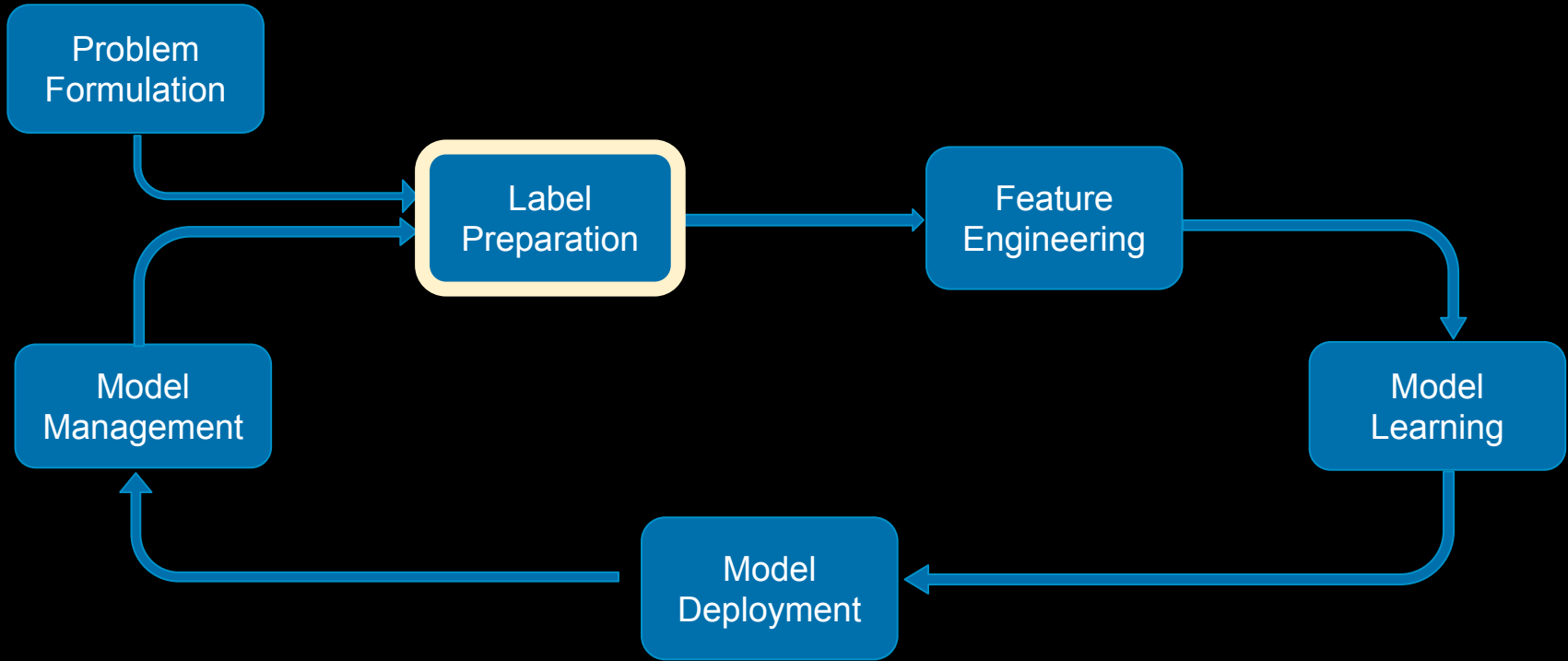
**Binary classification problem:** Let  $y_i$  represents the status of the enterprise account

$$y_i = \begin{cases} 1 & \text{closed won opportunity} \\ 0 & \text{closed disengaged opportunity} \end{cases}$$

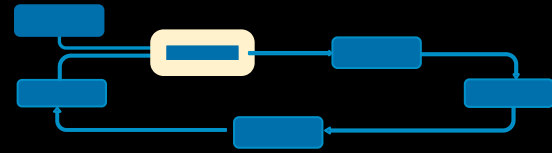
$$P(y_i | \text{account}_i) = ?$$

# Case Study - B2B Modeling

---



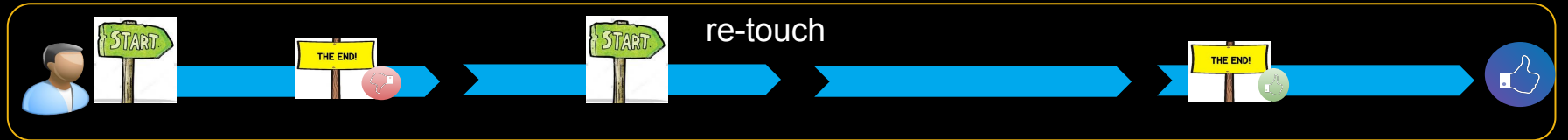
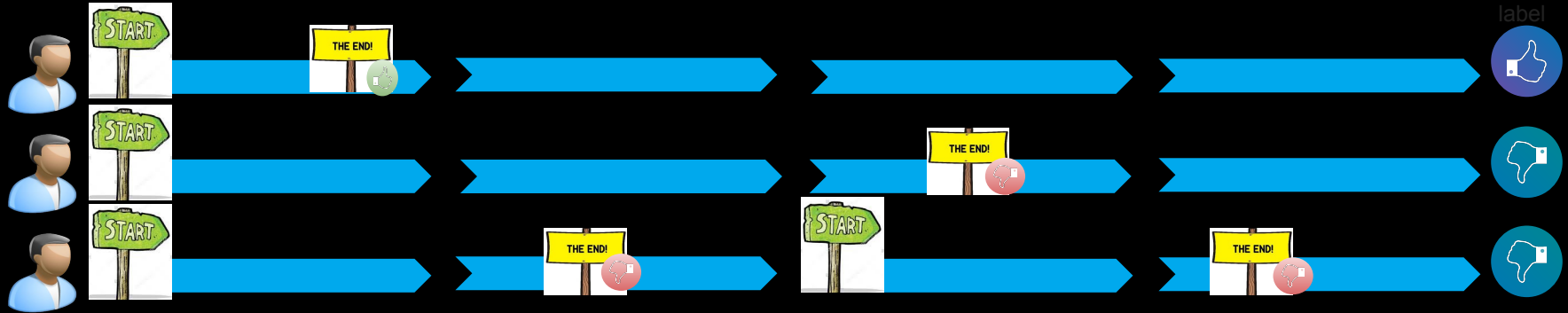
# Label Preparation



Label is defined at (account + region) level

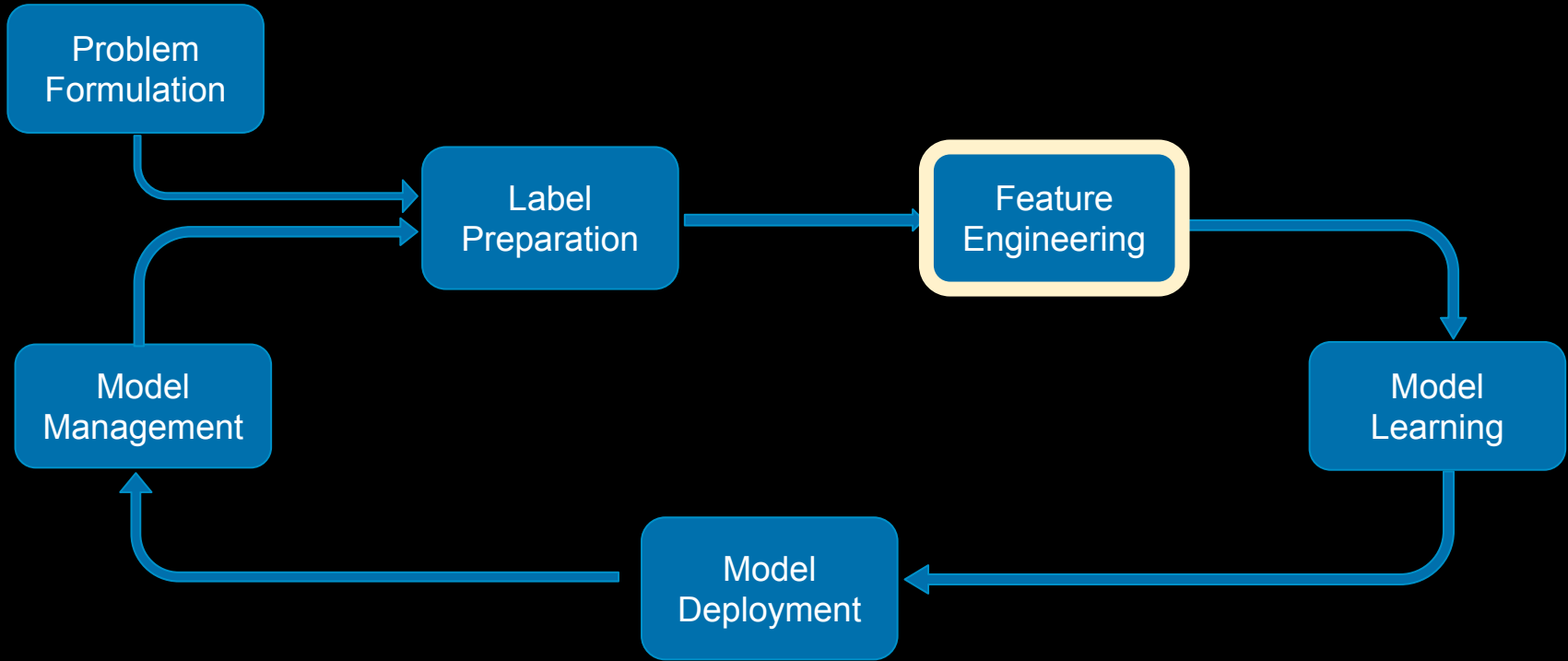
- **Positive – 1**: closed won opportunity.
- **Negative – 0**: closed disengaged opportunity

- find explicit negatives
- try until won: flip opportunities

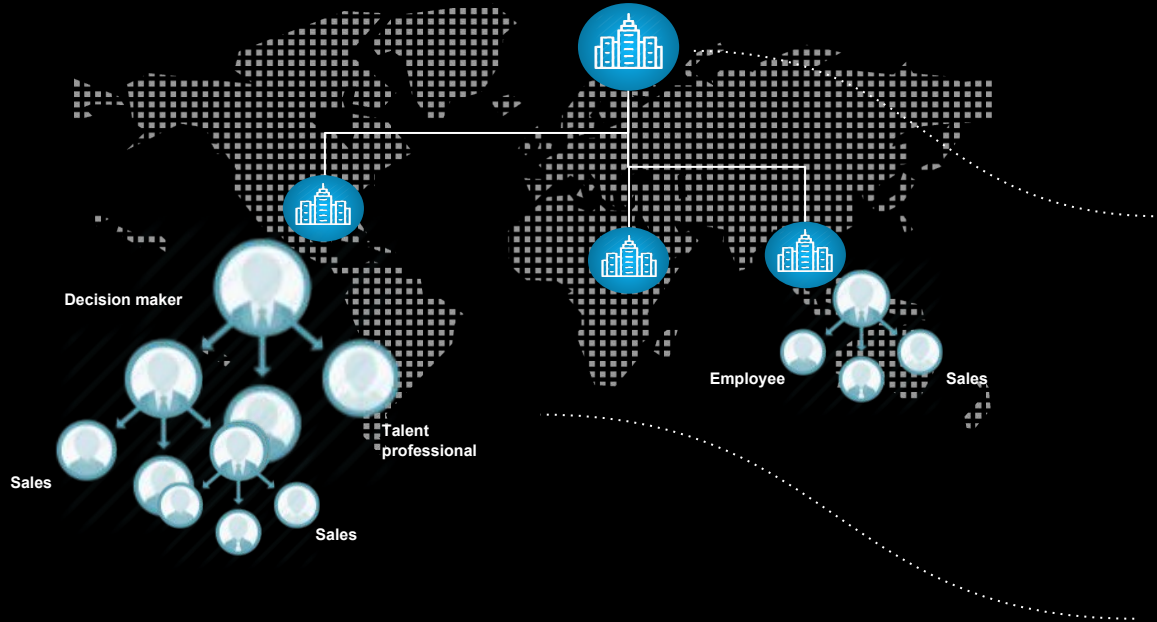


# Case Study - B2B Modeling

---



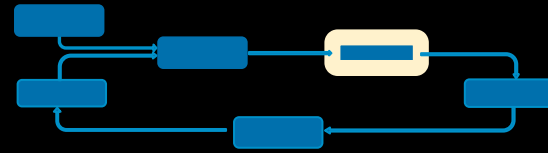
# Feature Engineering - Derive Features From IN Graph



-  Company Growth
-  Product Booking
-  Product Usage
-  Company Profile
-  Product Performance
-  Product Whitespace

-  Member Profile
-  Social Network
-  Activity

# Feature Engineering



## POTENTIAL FEATURES:

### Company & Growth

- Industry
- Region
- Sop
- Number of employees

### Linkedin Affinity

- Log-ins
- Direct ads impressions
- Followers

### Linkedin Spending

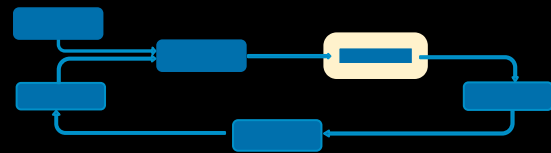
- Spending on other business lines
- Previous opportunities

### Product Related

- Sales Navigator: social selling
- Recruiter: social recruiter



# Feature Engineering

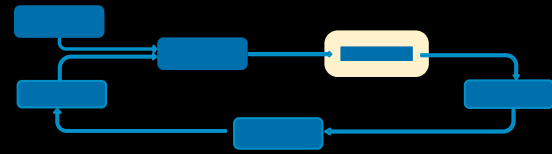


## FEATURE TRANSFORMATION:

- **Outlier** in “spending” data: e.g. \$100M, \$100K, \$10K  
Log Transformation: \$100M, \$100K, \$10K → \$18.4, \$11.5, \$9.2
- **Outlier** in “number of employees” data.  
Bucketize numerical feature as a set of categorical features  
e.g. number of employees: [1-10), [10-50), [50-500), [500-2000), =>2000
- Too many levels in “country” data which is a **categorical data**.  
Add binary variable on “country” data  
e.g. country →
  - country\_is\_us : {0,1}
  - country\_is\_canada: {0,1}
  - country\_is\_germany: {0,1}

# Feature Engineering

---



## FEATURE PROFILING:

For **numerical** data: Max, min, average, correlation with label.

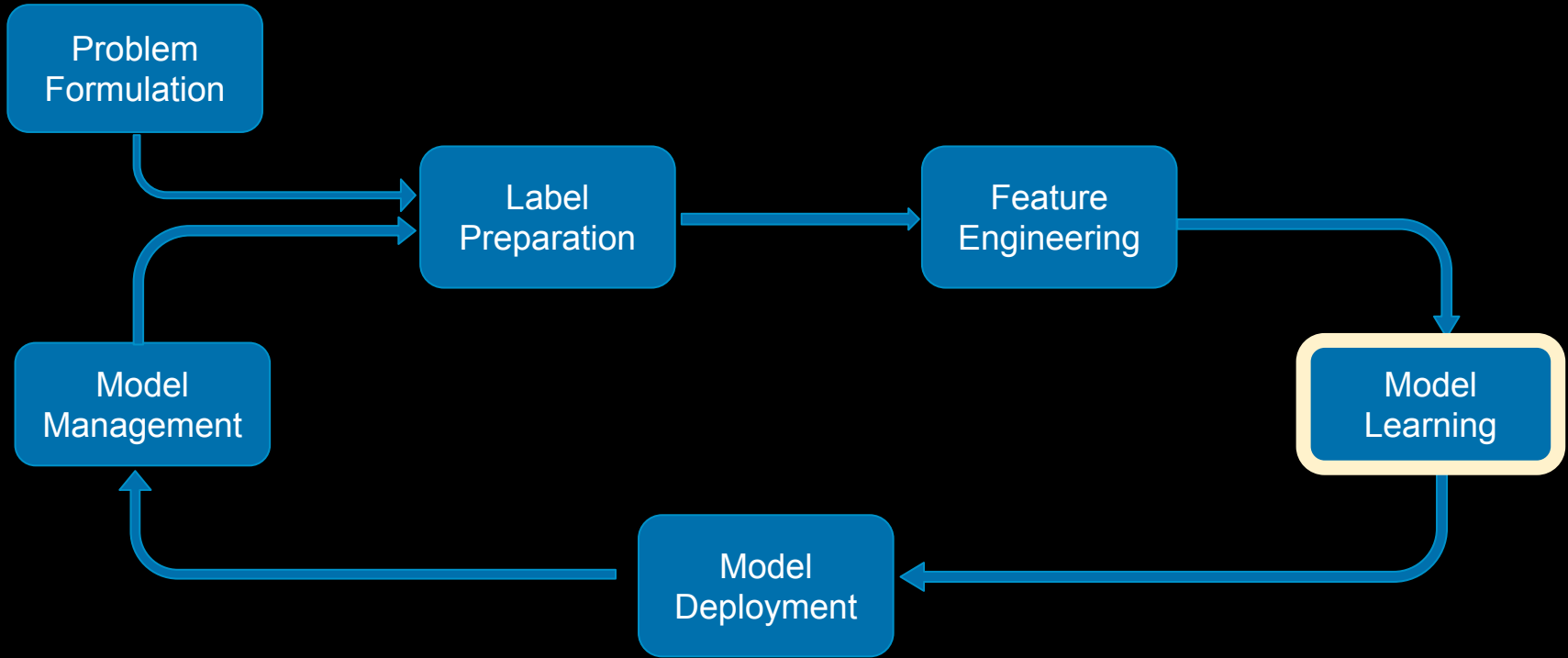
For **categorical** data: Distribution of values, non-null values.

Helps us see:

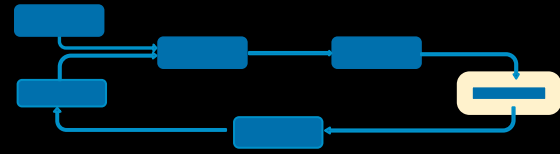
- Outliers
- Missing Data
- Data Quality Issues
- Data Leakage

Note: High Correlation with label + Sparse Data does not necessarily mean data leakage!

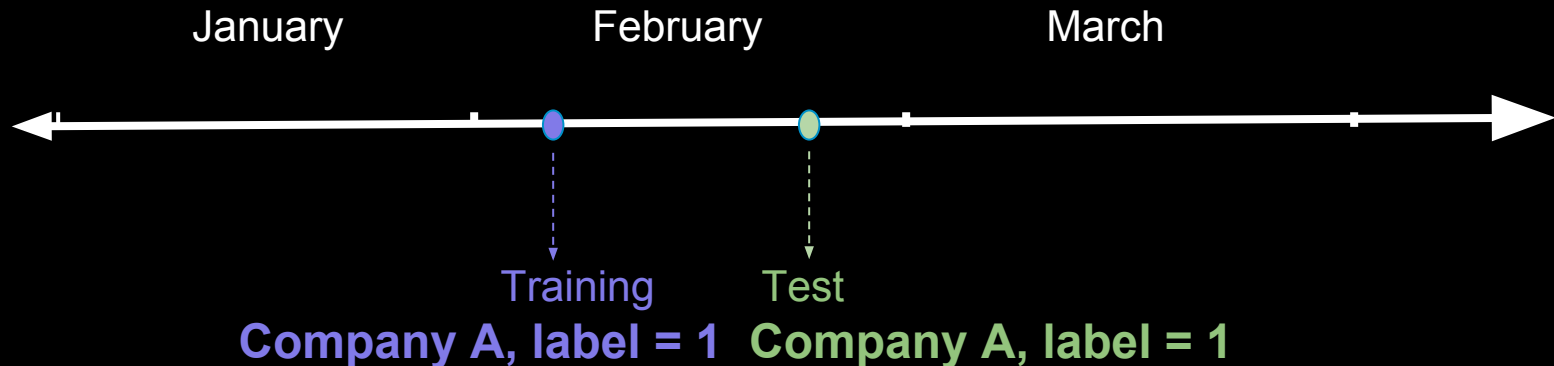
# Case Study - B2B Modeling



# Model Learning - Data Partitioning

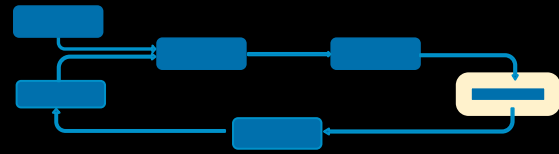


- We'll split the data into training, validation and testing using the ratio 60%-20%-20%.
  - We can have training and validation in the same time range and testing in different time range (preferably more recent time range).
  - Need to pay attention to companies with multiple entities.
- Suppose we are doing monthly aggregation.



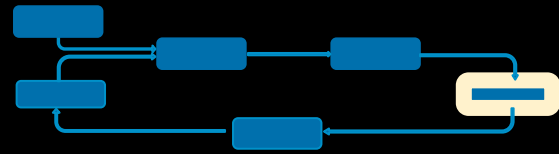
# Model Learning

---



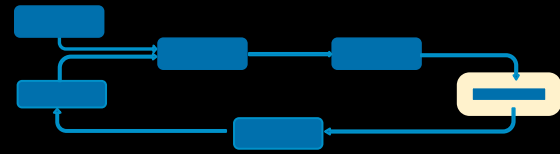
- Need a stable model, so we choose Random Forest Classification.
- Some tips for choosing hyperparameters:
  - Number of trees:
    - If the data is large, need many trees.
    - Too many features, need many trees.
    - More trees will result better accuracy and reduce bias, but also mean more computational cost and after certain number, the improvement is negligible.
  - Tree depth:
    - Deeper trees reduce the bias.
    - Limit the depth if dealing with noisy data
- Use validation set, to choose which hyperparameter to use.
- Compare error rate in the training set and in the validation set, to catch possible overfitting/bias.

# Model Learning

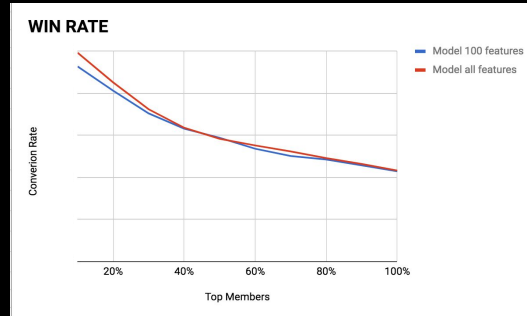


- Need a stable model, so we choose Random Forest Classification.
- Some tips for choosing hyperparameters:
  - Number of trees:
    - If the data is large, need many trees.
    - Too many features, need many trees.
    - More trees will result better accuracy and reduce bias, but also mean more computational cost and after certain number, the improvement is negligible.
  - Tree depth:
    - Deeper trees reduce the bias.
    - Limit the depth if dealing with noisy data
- Use validation set, to choose which hyperparameter to use.
- Compare error rate in the training set and in the validation set, to catch possible overfitting/bias.

# Model Learning - Validation

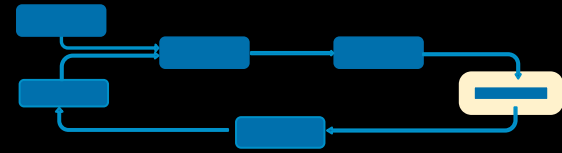


- Standard ROC: 0.64
- Check top performer features if they make sense or not
- Conversion/Win rate comparison between model with all features and model with only top 100 features:



- Review top false positives and false negatives, any bias?
- Field validation - set up review session with business partners in each region to collect feedback and suggestions.

# Model Learning - Interpretation



## Master Model



Master Model

## Component Models



Company Profile Component Model



Linkedin Spending Component Model



Social Selling Component Model

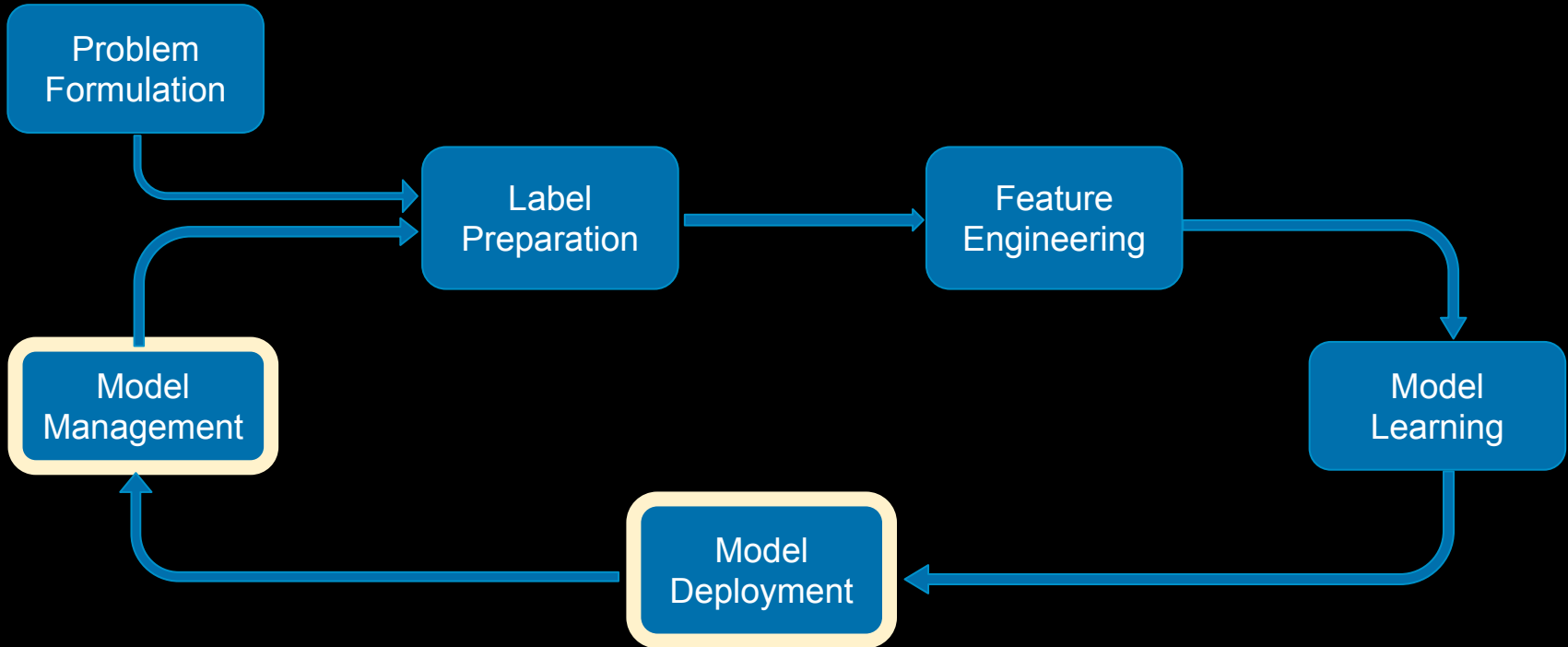


Company Growth Component Model



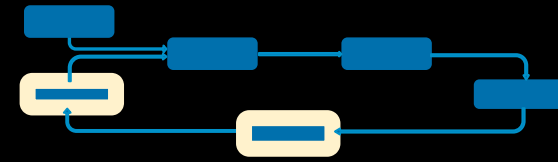


# Case Study - B2B Modeling



# Model Deployment & Management

---

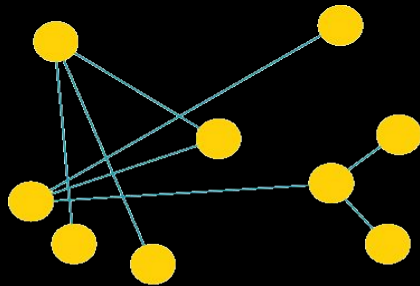


- Schedule and run the scoring monthly.
- Need to score customer accounts as well, not just prospects for completeness. Do customers score higher?
- After each scoring do some sniff test, e.g. are fortune 500 accounts lining as expected?
- Field validation
  
- Monitor model/feature performance.
- Refresh model as needed.
- Weekly review new wins/loses by segment.

# Key Takeaways

---

- Introduced the lifecycle of big data analytics and data science
- Leverage advances in big data analytics and deliver a better product
- End-to-end walkthrough of a production model with pitfalls and challenges
- Had a concrete review of how to leverage machine learning techniques to turn the chaos of data into straightforward and powerful end-to-end solution.



TURNING THIS...



INTO THIS

We are a world class team and need more talents to join us!



We are Hiring!

Please contact [nasingh@linkedin.com](mailto:nasingh@linkedin.com)

Thank you

