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
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: 20.6%
- Data analytics: 28.6%
- Engineer: 11.1%
- Business Leader: 20.6%
- Data ETL: 15.9%
- Data Product Manager: 14.3%
- Engineering Executive: 19%
- Engineering Director (Management): 15.9%

- Less than 10%: 42.9%
- 10-30%: 19%
- 30-70%: 14.3%
- Over 70%: 15.9%
- I do not know any machine learning technique: 15.9%
From “Big data” to “AI”?

- Financial
- Social network (LNKD, FB, etc)
- CRM
- Weblog (GOOG, YAHOO, EBAY, etc)
- Shared economy, IoT, Instant Messaging (Uber, Didi, Airbnb, WeChat, Snapchat, etc)
- Health care? Environmental science? Education? Entertainment?
Analytics evolution: from data to impact

Intelligence (Prediction/Optimization/Automation)

Information/Knowledge
Why did it happen?

Data
What happened?

Insights
What is the best that could happen?

Change & Impact
Recommend, Transform, Implement & monitor

Custom Analyses => Scalable automated solutions (AI)
LinkedIn’s business model & why analytics is important

- Member growth and engagement
- Relevant and valuable products and services
- Critical mass of data

Technology platform
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
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 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

<table>
<thead>
<tr>
<th>Vanity vs. Actionable</th>
<th>Page Views vs. Purchases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leading indicator</td>
<td>DAU* vs. Active Purchasers</td>
</tr>
<tr>
<td>Member retention</td>
<td>DAU vs. Cohort AU</td>
</tr>
<tr>
<td>Effective channel</td>
<td>CPC** vs. CPA***</td>
</tr>
<tr>
<td>Removing seasonality</td>
<td>WoW vs. YoY</td>
</tr>
</tbody>
</table>

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:

- **Ask**: Actionable Insights Lead to Production Ideation
- **Measure**: Success Metrics Definition
- **Learn**: Release & Post-Launch Insights
- **Test**: Experimentation and Iteration
- **Develop**: Tracking Instrumentation Specification
Example 1:
Introducing Job Tab
Actionable Insights Lead to Product Ideation

Success Metric Definition

Tracking Instrumentation specification

Experimentation and Iteration

Release & Post-Launch Insights

Ask

Measure

Develop

Test

Learn

Ask

Measure

Develop

Test

Learn

Move Profile Entry Point

New Jobs Entry Point
Hypothesis:

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

Invest in developing the right success metric.
What to consider when testing such a change?

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

---

A/B Testing

**Control**

**Treatment**
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

<table>
<thead>
<tr>
<th>Overall LinkedIn ecosystem</th>
<th>Sessions</th>
</tr>
</thead>
</table>
| “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
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
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?

Invest in developing the right success metric.
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?
LinkedIn’s Social Network

500+M Members
9M Companies
29K Schools
Examples: B2B Analytics

Business Needs

Acquire New Customers

- What is the market segmentation?
- How to prioritize marketing leads?
- How much is the potential spending from a client if acquired?
- Who are more likely to become customers? Why?
- What is the best channel to acquire?

Seal the deal

Empower Existing Customers

- How are new customers onboarded?
- How are customers engaging with products based on different segments?
- Which customer is going to renew less, why and what can we do?
- Which customer has upsell potential, why and what product to buy?

Demand Generation

Onboard & Retention

Business Growth

B2B Predictive Modeling
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.

Binary Classification

Purchaser

No Purchase
Example: Web Content Analytics

Social data
- Facebook
- Twitter
- YouTube
- Instagram
- Pinterest
- LinkedIn

Customer feedback
- Customer service
- Group updates
- Network updates

Survey results

Text Classification

Relevance
- LinkedIn
- Non-LinkedIn

Products Categorization
- Home Page
- Mobile
- Message
- Subscription

Sentiments
- Happy
- Neutral
- Sad
- Angry
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”

- Problem Formulation
- Label Preparation
- Feature Engineering
- Model Learning
- Model Deployment
- Model Management
Machine Learning: The Process

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

- Label Preparation
- Feature Engineering
- Model Learning
- Model Deployment
- Model Management
Problem Formulation

Skills Clustering
- Art
- Business
- Marketing
- Engineering

Estimating Customer Spending

Predicting Churn
- Churn
- No Churn

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

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

$$P(y_i \mid \text{member}_i, \text{context})=?$$
Machine Learning: The Process

- Problem Formulation
- Label Preparation
- Feature Engineering
- Model Deployment
- Model Learning
- Model Management
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\}
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 evolvement, etc
- Approach: percentage change, T-test, etc -> aware of any **statistical assumption** restrictions

![Graph showing data trends with Labor Day marked]
# Feature Engineering - Integration

## Label set

<table>
<thead>
<tr>
<th>user_id</th>
<th>timestamp</th>
<th>Label</th>
<th>pageviews</th>
<th>searches</th>
<th>Tenure</th>
<th>Is employed</th>
<th>connections</th>
<th>Connect in</th>
</tr>
</thead>
<tbody>
<tr>
<td>xx1</td>
<td>2007-01-01</td>
<td>0</td>
<td>97</td>
<td>5</td>
<td>3876</td>
<td>1</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>xx2</td>
<td>2007-08-01</td>
<td>1</td>
<td>27</td>
<td>2</td>
<td>60</td>
<td>0</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>xx3</td>
<td>2007-06-31</td>
<td>0</td>
<td>null</td>
<td>null</td>
<td>2700</td>
<td>1</td>
<td>120</td>
<td>30</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

## Feature set 1

<table>
<thead>
<tr>
<th>user_id</th>
<th>pageviews</th>
<th>searches</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>xx1</td>
<td>97</td>
<td>5</td>
<td>...</td>
</tr>
<tr>
<td>xx2</td>
<td>27</td>
<td>2</td>
<td>...</td>
</tr>
<tr>
<td>xx4</td>
<td>58</td>
<td>4</td>
<td>...</td>
</tr>
</tbody>
</table>

## Feature set 2

<table>
<thead>
<tr>
<th>user_id</th>
<th>Tenure</th>
<th>Is employed</th>
</tr>
</thead>
<tbody>
<tr>
<td>xx1</td>
<td>3876</td>
<td>1</td>
</tr>
<tr>
<td>xx2</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>xx3</td>
<td>2700</td>
<td>1</td>
</tr>
</tbody>
</table>

## Feature set 3

<table>
<thead>
<tr>
<th>user_id</th>
<th>connections</th>
<th>Connect in</th>
</tr>
</thead>
<tbody>
<tr>
<td>xx1</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>xx2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>xx3</td>
<td>120</td>
<td>30</td>
</tr>
</tbody>
</table>

...
Feature Engineering - Static & Dynamic

Training data with timestamp

mid & Y & X
---
xx1 & 0 & Snapshot features & Aggregated features
xx2 & 1 & ... & ...
xx3 & 0 & ... & ...

trans & desc
---
self & x
sign & sign(x)
log & sign(x)*log(abs(x)+1))
isna & 1:x==null; 0:x!=null

Feature Mart
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 data

- **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.
    - \{skills_{member}\} X \{skills_{job}\}
    - \{skills_{member}\} X \{industry_{job}\}

- **NULL**
  - Dummy variables indicator
Feature Engineering - Feature Representation Learning
Machine Learning: The Process

1. Problem Formulation
2. Label Preparation
3. Feature Engineering
4. Model Learning
5. Model Deployment
6. Model Management
Model Learning

Solvers
- Logistic regression
- Decision Tree
- Random Forest
- Gradient Boosting Trees
- SVM

Data Partitioning
- Labeled Data Set
- Training Set
- Validation Set
- Testing Set

Model Training
- Model Validation
- Model Selection
- Model Testing

Best Model
Model Learning

Solvers
- Logistic regression
- Decision Tree
- Random Forest
- Gradient Boosting Trees
- SVM

Data Partitioning
- Labeled Data Set
- Training Set
- Validation Set
- Testing Set

Model Training
- Model Selection
- Model Validation
- Model Testing

Best Model
Model Learning - Data Partitioning

Basics & What to pay attention

1. **Split Ratio** may depend on the size of the data, typically train/validation=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

Logistic regression
Decision Tree
Random Forest
Gradient Boosting Trees
SVM

Solvers

Labeled Data Set
Data Partitioning
Training Set
Validation Set
Testing Set

Model Training
Model Validation
Model Selection
Model Testing

Best Model
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.
### Performance Evaluation

<table>
<thead>
<tr>
<th>ID</th>
<th>Label</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Truth (Label):**
  - **True Positive:**
  - **False Positive:**
  - **False Negative:**
  - **True Negative:**

- Consider “1” as positive, “0” as negative.
Performance Evaluation

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 - 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

- Problem Formulation
- Label Preparation
- Feature Engineering
- Model Learning
- Model Deployment
- Model Management
Model Deployment

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

<table>
<thead>
<tr>
<th>id</th>
<th>score</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxx1</td>
<td>0.99</td>
<td>1</td>
</tr>
<tr>
<td>xxx2</td>
<td>0.87</td>
<td>2</td>
</tr>
<tr>
<td>xxx3</td>
<td>0.72</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

marketing campaigns
Machine Learning: The Process

1. Problem Formulation
2. Label Preparation
3. Feature Engineering
4. Model Learning
5. Model Deployment
6. Model Management
Model Management

Is Your Fruit Rotting?

Day 1

Day 30

Photo credit: asmfoto Marcell Mizik, photo license with Depositphotos File Purchase Agreement #41549281
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

- Feed in new training data to generate new model periodically
- Assign versions to models built over time
- Monitor changes over time
- Ensemble historical models as one of the candidate models
Performance Measurement via A/B Test

Algorithm a
80%

Algorithm b
20%

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

- raw data
- feature engineering
- raw features
- crowd/internal judgments
- label preparation
- log data
- a/b test reports
- validation data
- training data
- model training
- model evaluation & model selection
- offline scoring and indexing
- best model
- scoring features
- offline systems
- online/offline systems
- online A/B test
- raw features
- feature integration
- features with label
- data partitioning
- features integration
- label
- log data
- model performance
- compute offline evaluation metrics
- testing data
Machine Learning as a Service
Examples of Machine Learning Platforms

- Amazon Web Services + Amazon ML
- Amazon Web Services + Apache Spark
- Microsoft Azure ML + Hadoop/Spark
- Amazon Web Services + Turi
- Amazon Web Services + Databricks
- Amazon Web Services + H2O.ai
- Google ML
## Comparison of Machine Learning Platforms

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

Application Layer
- B2B Analytics Modeling
- B2C Analytics Modeling
- Web Analytics Modeling

Intelligence Layer
- Feature Preparation
- Model Building
- Model Ensembling & Deployment
- Model Management
- Model Reasoning

Data Layer
- Feature Mart
- Feature management
- Label management

Tools and Technologies:
- Tableau
- TensorFlow
- Highcharts
- H2O.ai
- Apache Spark
- dmlc
- XGBoost
- MLLib
- Azkaban
- Hadoop
- Avro
- Teradata
- Mahout
Consider **complexity** and **unification** of Feature mart/metastore to **enable flexibility** on feature integration and feature engineering.

**Data Layer - Basics: what to build**

- **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 ad hoc 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**
Simplicity is the ultimate sophistication.

*Leonardo da Vinci*
Common Pitfalls and Challenges

- Label Quality/Noise
- Class Imbalance

- Data Leakage
- Data Quality
- Categorical Data
- Missing Data
- Outliers

- High Dimensionality
- Overfitting
- Scalability, speed, fast iteration
- Model Interpretation

- Model Degradation
- Feature quality monitoring

- A/B testing
- Dependencies

- Problem Formulation
- Model Management
- Model Deployment
- Label Preparation
- Feature Engineering
- Model Learning
- Dependencies
Survey Results

- **Model Interpretation**: 38 (61.3%)
- **Data Quality**: 31 (50%)
- **Categorical Data**: 17 (27.4%)
- **Data Leakage**: 12 (19.4%)
- **Missing Data**: 17 (27.4%)
- **Outliers**: 14 (22.6%)
- **Class Imbalance**: 19 (30.6%)
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? **NO!**
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
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
- Transform a categorical feature into multiple binary ones
  - Introduce an additional ‘others’ feature to represent all the new categories in the testing set
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 vs. 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

### Training set

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

- Leaking of information from the future into the past

<table>
<thead>
<tr>
<th>price_1</th>
<th>price_2</th>
<th>price_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yesterday</td>
<td>Today</td>
<td>Tomorrow</td>
</tr>
</tbody>
</table>

feature_i = (price_1 + price_2 + price_3) / 3

Feature = “Rep id”  Target: customer churn

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

Target = future price?
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

Business Needs

Acquire New Customers
Demand Generation
Seal the deal

Empower Existing Customers
Onboard & Retention
Business Growth

B2B Predictive Modeling

Who are more likely to become customers? Why?
Case Study - B2B Modeling

- Problem Formulation
- Label Preparation
- Feature Engineering
- Model Learning
- Model Deployment
- Model Management
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

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

$$P(y_i \mid \text{account}_i) = ?$$
Case Study - B2B Modeling

Problem Formulation

Label Preparation

Feature Engineering

Model Learning

Model Deployment

Model Management
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

re-touch
Case Study - B2B Modeling

- Problem Formulation
- Label Preparation
- Feature Engineering
- Model Learning
- Model Deployment
- Model Management
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 $\rightarrow$ $18.4, 11.5, 9.2$

- **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), $\Rightarrow$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 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
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.

January          February          March

Company A, label = 1  Company A, label = 1
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:

![WIN RATE graph](image)

- 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.
Fast growth reflects potential demands, find out how our product can help the customer further grow and increase social selling.

### Account Propensity Score

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>92</td>
<td>65</td>
<td>30</td>
</tr>
<tr>
<td>Growth</td>
<td>Top Industry</td>
<td>Fast Growth</td>
<td>Decline</td>
</tr>
<tr>
<td>Profile</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Spending</td>
<td>High</td>
<td>Medium low</td>
<td>Low</td>
</tr>
<tr>
<td>Social Selling</td>
<td>High</td>
<td>Medium low</td>
<td>Medium low</td>
</tr>
</tbody>
</table>
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
Thank you