

# **Regressing Towards Simpler Prediction Systems**

Sibyl project, Google Inc  
(Speaker: Tushar Chandra)

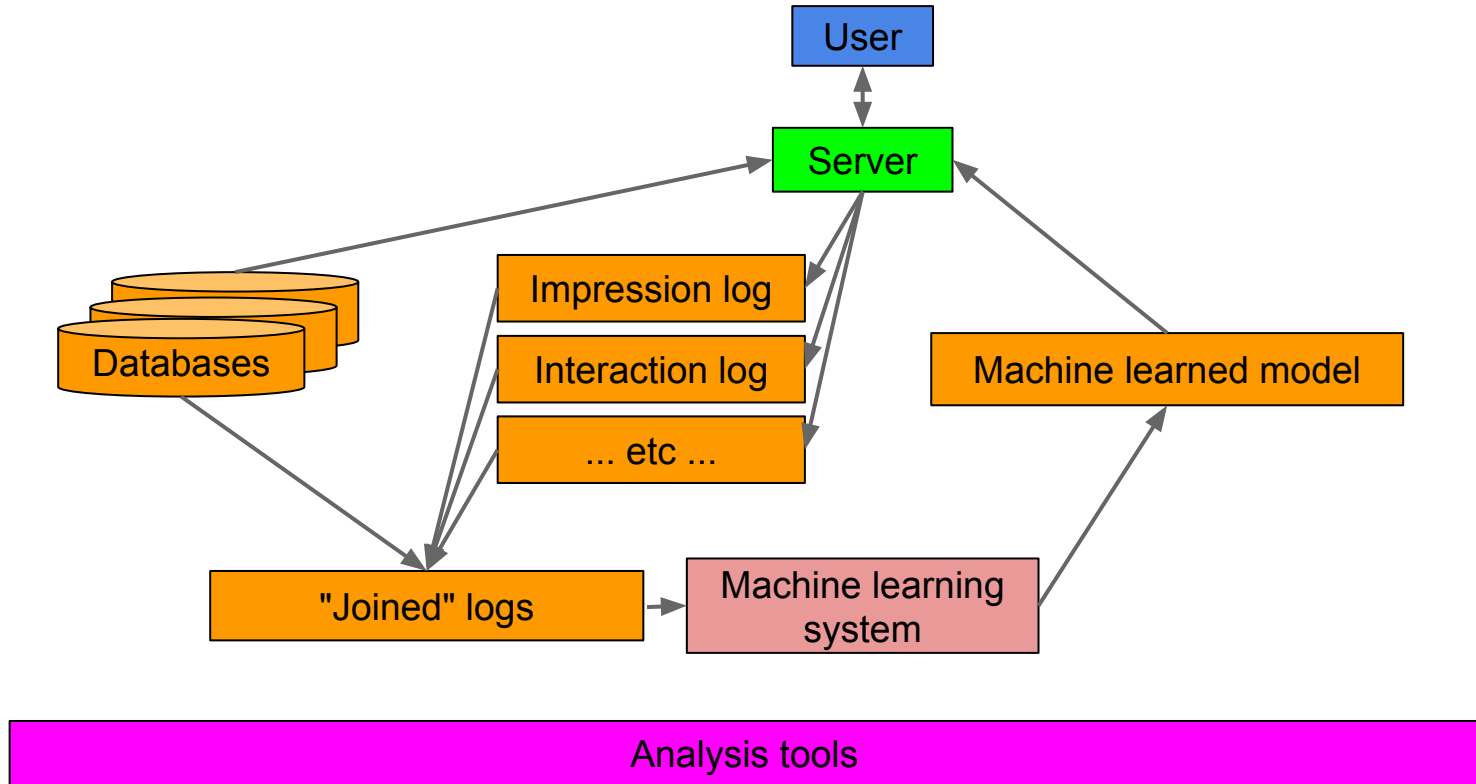
# Example: YouTube recommendations

The image shows a YouTube video player for the video '6 - Heave Ho, Thomas! - Thomas the train' by user 'onewish80'. A chart titled 'Click prob' is overlaid on the video player, showing the probability of users clicking on various recommended videos. The chart consists of a vertical bar with horizontal arrows pointing to the right, each labeled with a percentage. The recommended videos are listed on the right side of the page, enclosed in a red box. A red arrow points from the text 'Recommended Videos' to this box.

Click Probability	Recommended Video Title	Views
9%	Thomas and Friends : Pop Goes Thomas	1,750,342
8%	Thomas And Friends: Misty Island	6,822,042
6%	Thomas And Friends : Rescue On The Rails	200,279
4%	Thomas the Tank Engine & Friends - by Vysethedetermined2	2,306,251
3%	Thomas the Train .....Percy and the	1,907,423
2%	Thomas & Friends - Trainz Special No. 4	353,570
1%	Takara Tomy Thomas And The Shaking	13,728,918

Recommended Videos

# Prediction system architecture



# What is Sibyl?

- Large-scale machine learning system
  - Classification, regression, factorized models
- Designed for the largest Google data sets
  - Process 100B+ training examples
  - Explore 100B+ unique features
  - Serve models with 1B+ features
- Uses principled, state-of-the-art algorithms

# What is Sibyl?

- Prediction platform
- Widely used across Google

# Why Do Teams Collaborate with Sibyl

- Because we help teams build simple, maintainable, understandable prediction systems that significantly improve their product
- Not because of our machine learning algorithms

# Wins from ML

Manual System

↓ +50%

Add Machine Learning

↓ +10%

Good Machine Learning

↓ +2%

Exceptional Machine Learning

# Wins from Features

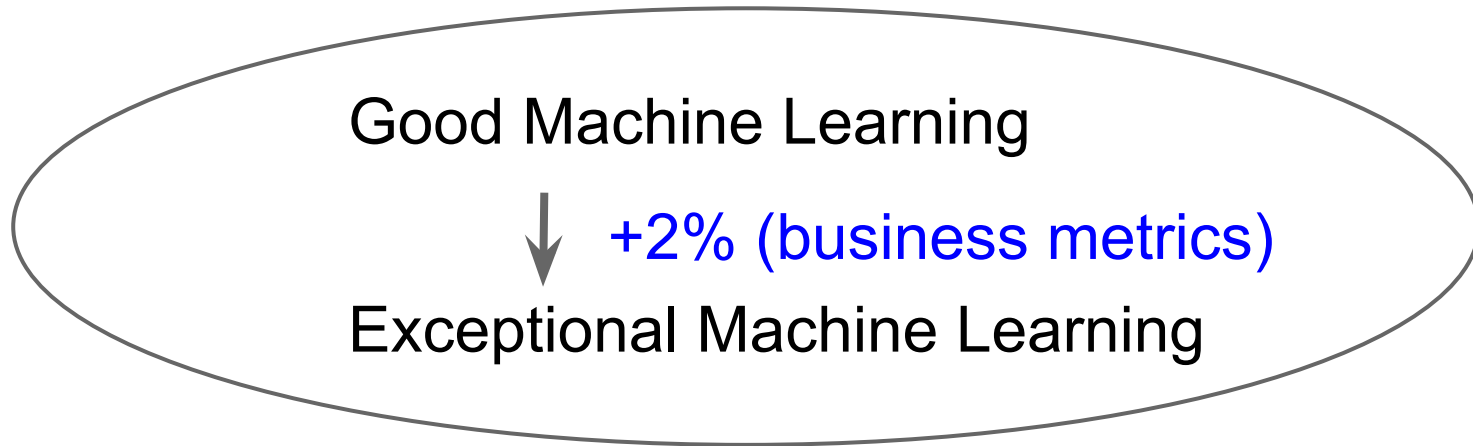
Document Features +5%

Query Features +5%

UI Features +5%

*O(10) such* +5% each

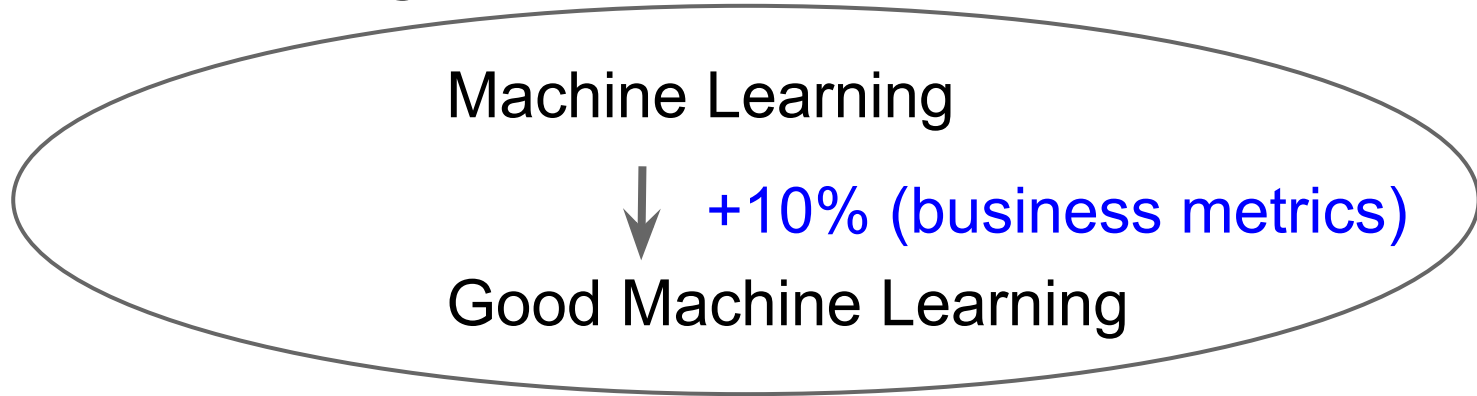
# How Sibyl Started



- Competed with existing system
  - +1.5%, cost of switching was too high

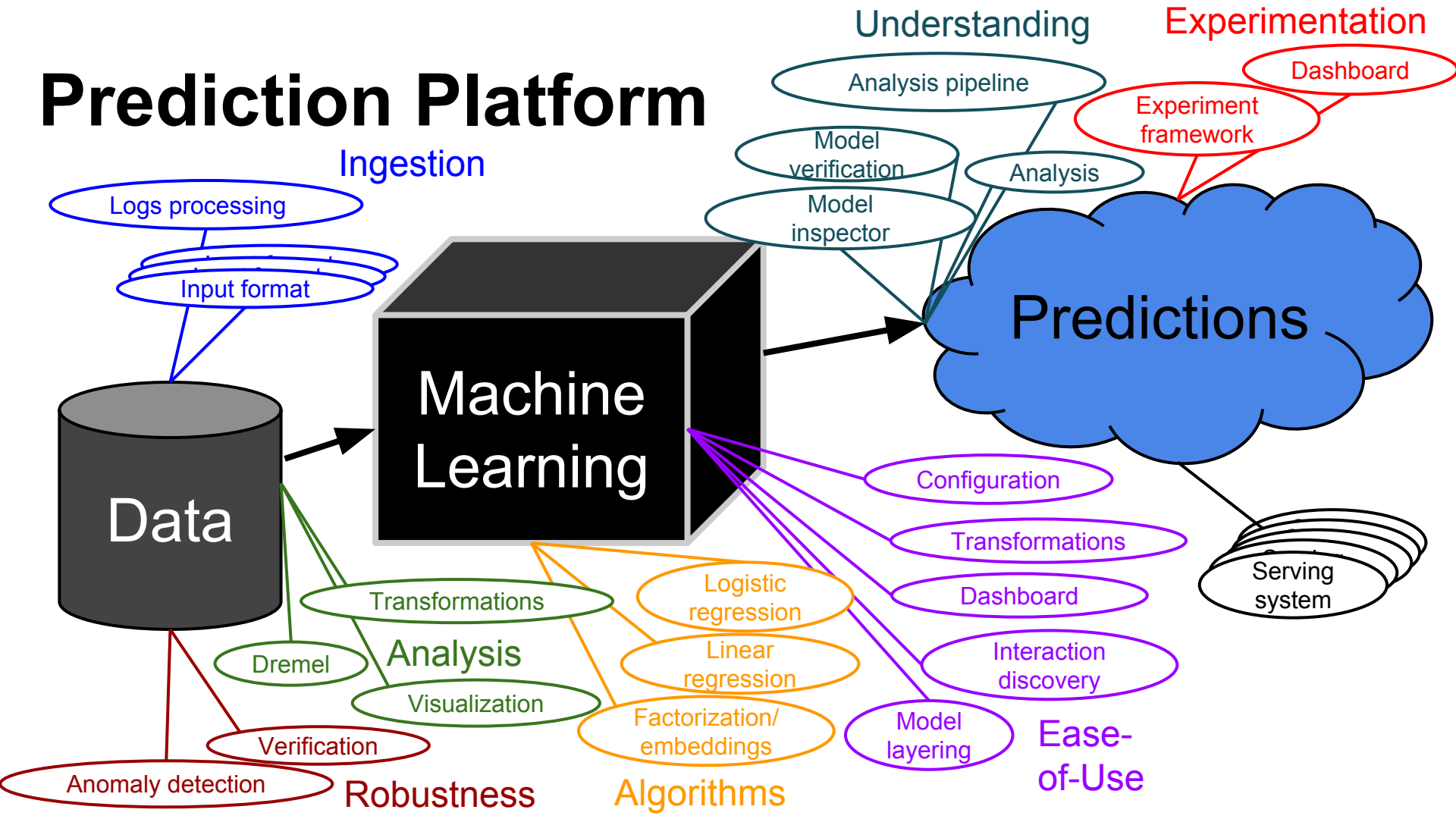


# Case Study: Small YouTube property



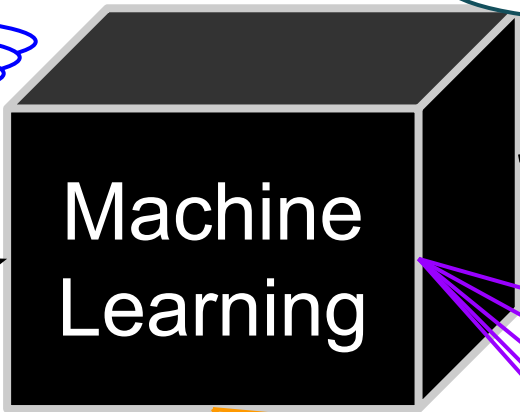
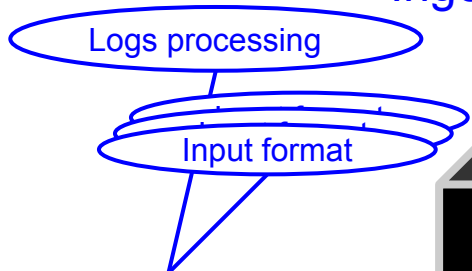
- Only two feature types (Country, Video ID)
- 1 month to implement linear regression
- 9 months to launch
- 20% improvement

# Prediction Platform

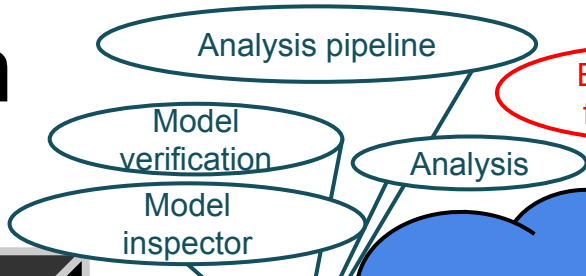


# Prediction Platform

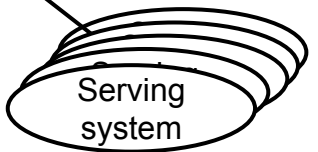
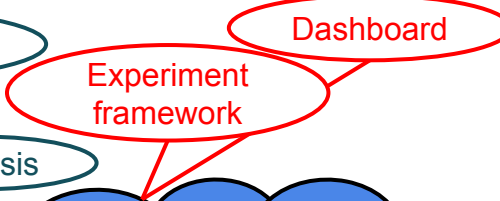
## Ingestion



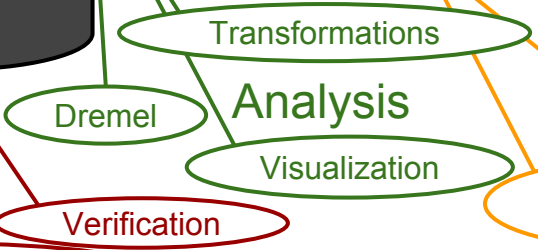
## Understanding



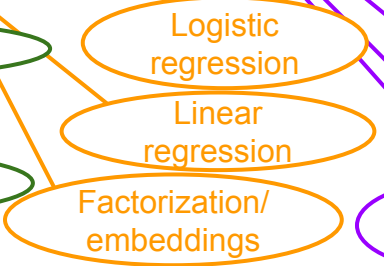
## Experimentation



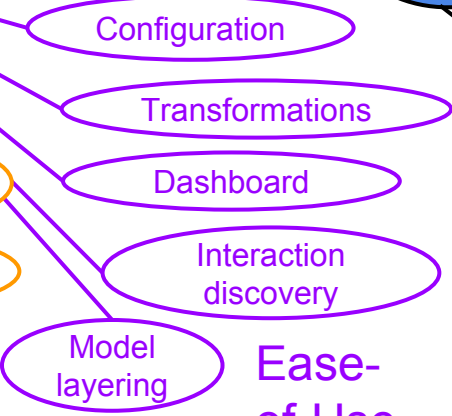
## Robustness



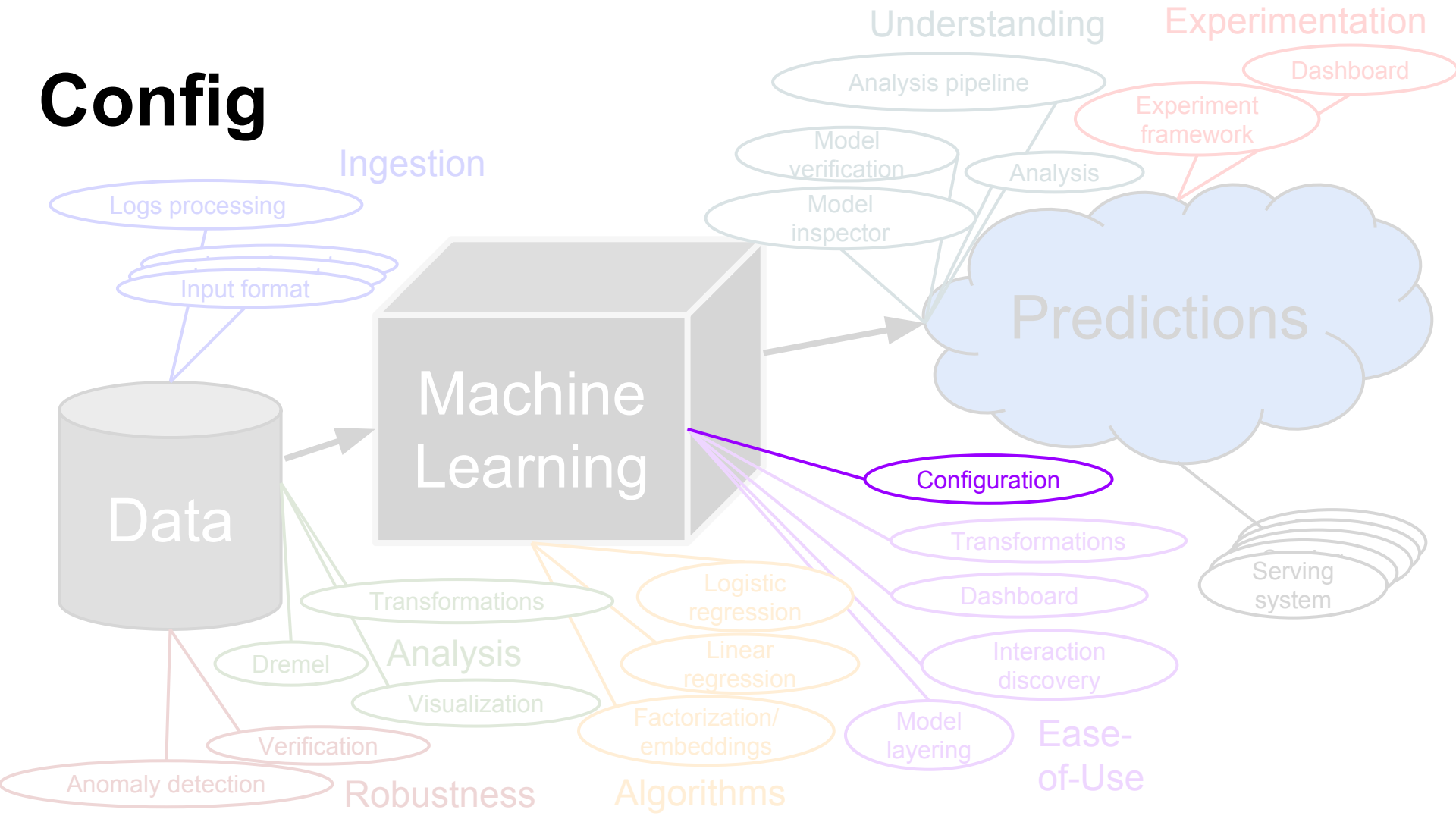
## Algorithms



## Ease-of-Use



# Config



# Config

How users use the system

Started out as an after-thought

Now

- On version 4.5
- >40K lines of code in the core
- ~100K lines of user code
- 2 person dev team and growing

Several other machine learning platform teams have had similar experiences

# Config challenges

Orchestrate all components in the system, e.g.,

```
c.Monitoring.SetAlertRecipients("tusharchandra@google.com")
```

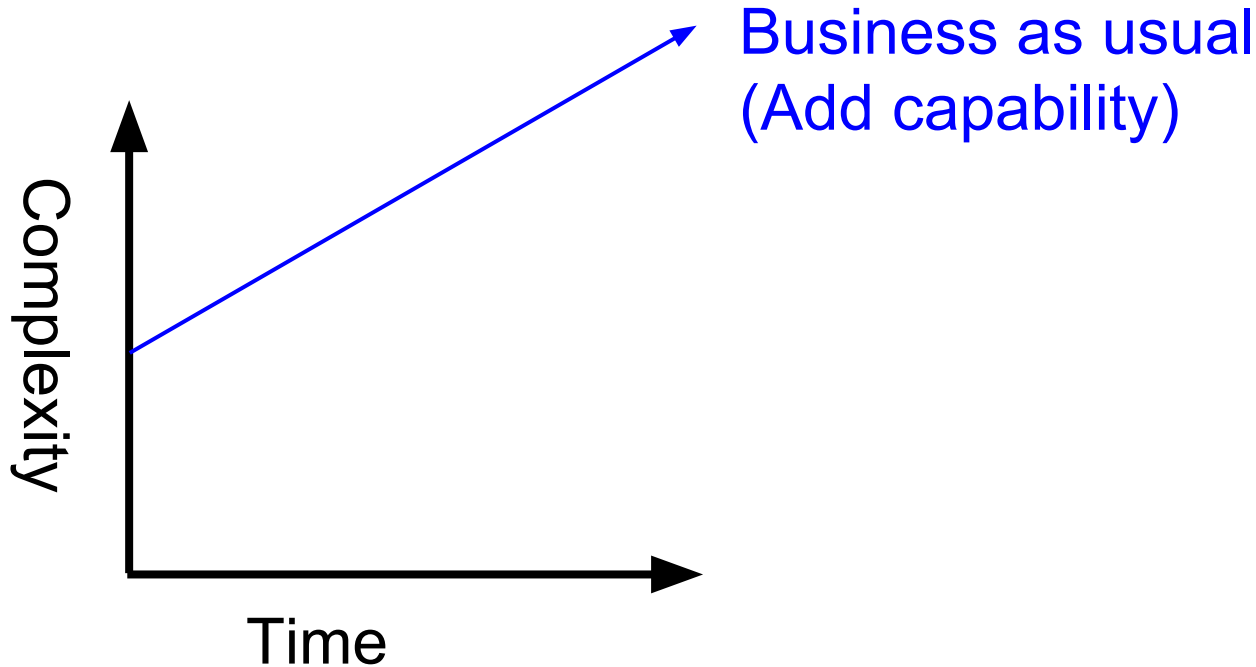
Setup models for training

- Transform data (e.g., discretize and train on bucket-id)
- Specify features, loss function, hyperparameters, etc.

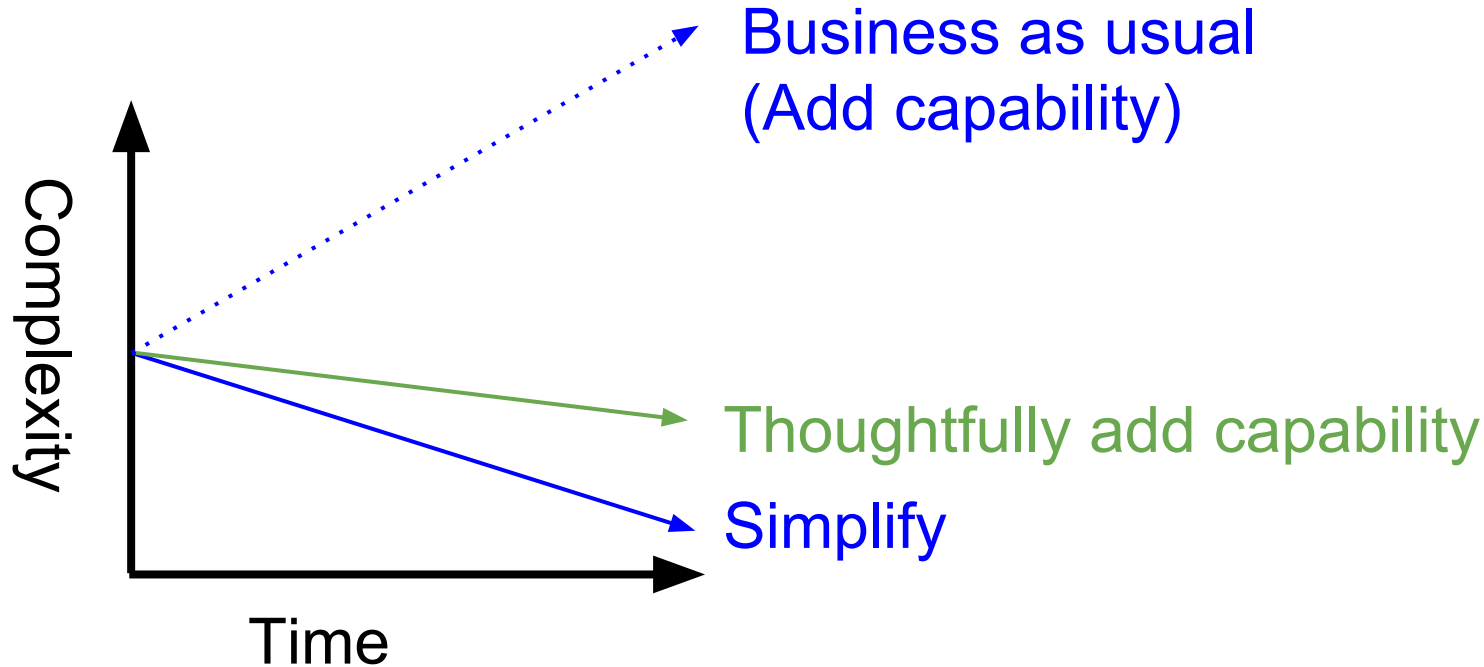
Used Go as our config language

- Better type system than Python

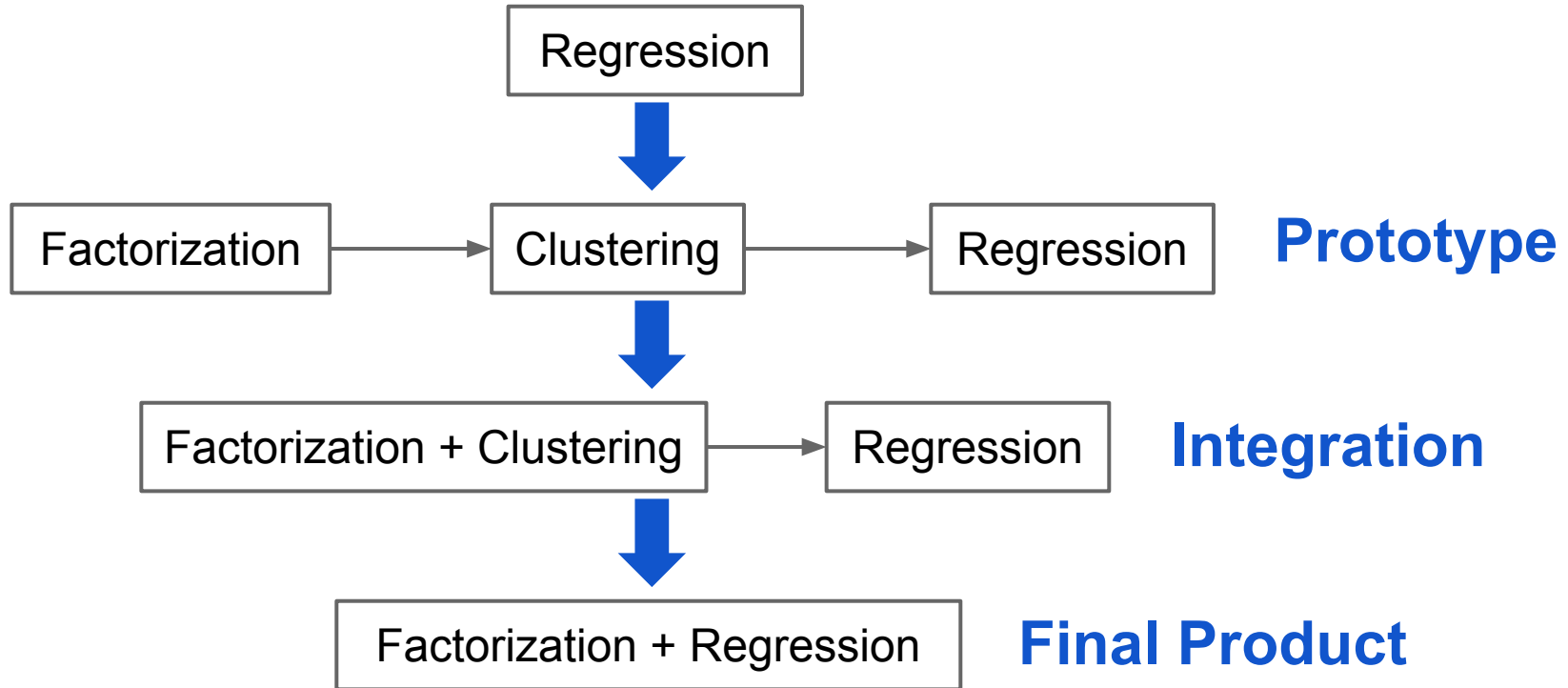
# Research instinct: add capability



# Actively manage complexity



# Thoughtfully Adding Factorization





# Future directions

New deployments

Mature deployments

Make it much easier

- System refinement
- Model understanding
- Education
- 10x scale

Feature engineering

Manual feature engineering has run its course

Need new algorithms, e.g.,

- Neural nets
- Ranking
- Embeddings
- Hidden state

# A note on Sibyl's performance

- Based on MapReduce and Google's cluster file system
- Iterative batch learner
- Spends most of its time on ML computation
  - Less than 50% of its time on MR+GFS+I/O

Conclusion: MR + cluster file system is a reasonable abstraction (from a performance perspective)

- See invited talk at DSN 2014 (on YouTube)