# Why Predictive Analysis is Slow, and How to Fix it

Art Munson
amunson@contextrelevant.com

∝ context relevant™

Chesapeake Large Scale Analytics Conference, 2015 context relevant™

Revolutionize the way people make decisions.

# What is predictive analytics?

- Training set of labeled cases:

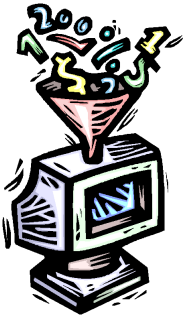
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 0.25 | 0.16 | 0.68 | $\rightarrow 0$ |
| 0 | 1 | 0 | 1 | 0 | 0.20 | 0.09 | 0.77 | $\rightarrow 0$ |
| 1 | 0 | 1 | 1 | 1 | 0.42 | 0.31 | 0.54 | $\rightarrow 1$ |
| 0 | 0 | 1 | 1 | 0 | 0.58 | 0.29 | 0.63 | $\rightarrow 1$ |
| 1 | 1 | 1 | 0 | 0 | 0.18 | 0.13 | 0.82 | $\rightarrow 0$ |

  ...

- Learn *model* that predicts outputs in train set from input *features*.

- Use model to make predictions on cases not used for training.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 0.20 | 0.16 | 0.68 | $\rightarrow ?$ |



How long did your last analytics project take?

Why does this happen?

Why does this happen?

Spoiler: It’s not the computer’s fault.

# Outline

Introduction

Where does the time go?

How to Increase Productivity

Closing Thoughts

# Many Steps, and All Take Time

| Stage | Median % Time |
|---|---|
| Data Access | 20% |
| Prepare Data | 30% |
| Modeling | 14% |
| Evaluate & Study Model | 20% |
| Report Results | n/a |
| Deployment | n/a |
| | 57 respondents |

*M.A. Munson. A study on the importance of and time spent on different modeling steps. SIGKDD Explorations Newsletter, 2011.*

# Running Example: Prioritized Call Lists

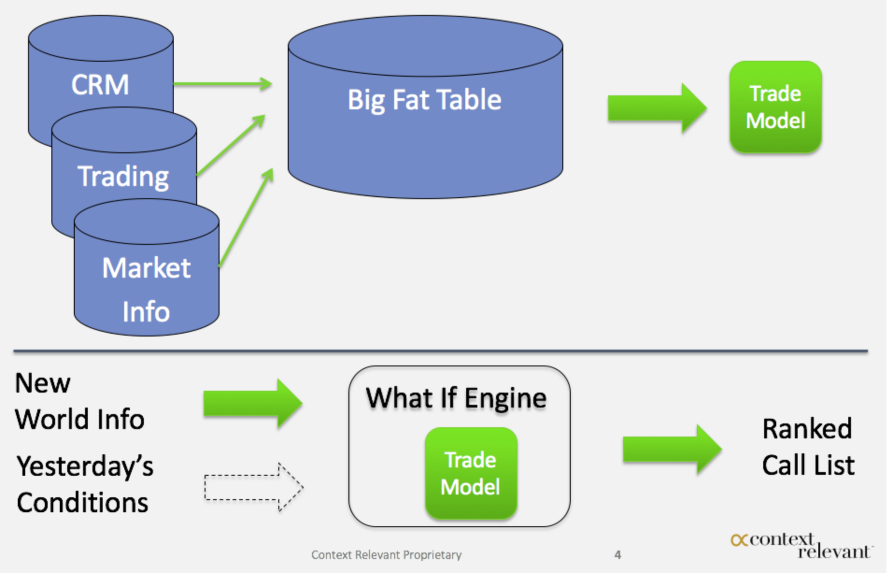
How do I optimize my sales force
to maximize profit?

# Running Example: Prioritized Call Lists

How do I optimize my sales force
to maximize profit?

Which customers to call, about which products?

# Running Example: Prioritized Call Lists (cont'd)



# Step 0: Data Access

Data access is a collaborative, iterative process.

For example …

| Month | Activity |
|---|---|
| * | Coordination meetings, data reviews. |
| Nov | Brainstorm useful data, go find owners. |
| Dec | Analysts get samples of key data, start prototyping. |
| Jan | Data update: column changes, delimiter. |
| Feb | Full data feeds available. |
| | Reorganized data: new folders, new server. |
| Mar | Data update: column changes. |
| | Added 3 new data feeds. |
| | Figure out missing join logic. |
| Apr | Data update: column changes. |
| | What does this column *really* mean? |
| | Time period mismatch! Now what? |

# Step 0: Data Access

Data access is a collaborative, iterative process.

For example ...

| MONTH | ACTIVITY |
|---|---|
| * | Coordination meetings, data reviews. |
| Nov | Brainstorm useful data, go find owners. |
| Dec | Analysts get samples of key data, start prototyping. |
| Jan | Data update: column changes, delimiter. |
| Feb | Full data feeds available. |
| | Reorganized data: new folders, new server. |
| Mar | Data update: column changes. |
| | Added 3 new data feeds. |
| | Figure out missing join logic. |
| Apr | Data update: column changes. |
| | What does this column *really* mean? |
| | Time period mismatch! Now what? |

## Step 0: Data Access

Data access is a collaborative, iterative process.

For example …

| MONTH | ACTIVITY |
|---|---|
| * | Coordination meetings, data reviews. |
| Nov | Brainstorm useful data, go find owners. |
| Dec | Analysts get samples of key data, start prototyping. |
| Jan | Data update: column changes, delimiter. |
| Feb | Full data feeds available. |
| | Reorganized data: new folders, new server. |
| Mar | Data update: column changes. |
| | Added 3 new data feeds. |
| | Figure out missing join logic. |
| Apr | Data update: column changes. |
| | What does this column *really* mean? |
| | Time period mismatch! Now what? |

# Step 0: Data Access

Data access is a collaborative, iterative process.

For example …

| MONTH | ACTIVITY |
|---|---|
| * | Coordination meetings, data reviews. |
| Nov | Brainstorm useful data, go find owners. |
| Dec | Analysts get samples of key data, start prototyping. |
| Jan | Data update: column changes, delimiter. |
| Feb | Full data feeds available. |
| | Reorganized data: new folders, new server. |
| Mar | Data update: column changes. |
| | Added 3 new data feeds. |
| | Figure out missing join logic. |
| Apr | Data update: column changes. |
| | What does this column *really* mean? |
| | Time period mismatch! Now what? |

# Step 0: Data Access

Data access is a collaborative, iterative process.

For example ...

| MONTH | ACTIVITY |
|---|---|
| * | Coordination meetings, data reviews. |
| Nov | Brainstorm useful data, go find owners. |
| Dec | Analysts get samples of key data, start prototyping. |
| Jan | Data update: column changes, delimiter. |
| Feb | Full data feeds available. |
| | Reorganized data: new folders, new server. |
| Mar | Data update: column changes. |
| | Added 3 new data feeds. |
| | Figure out missing join logic. |
| Apr | Data update: column changes. |
| | What does this column *really* mean? |
| | Time period mismatch! Now what? |

# Step 1: Prepare Data

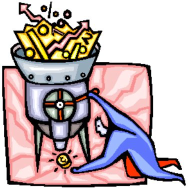
You can spend your whole life preparing data.

- Data Integration
  - Canonicalize join columns.
  - How to link data feeds missing common join key?
- Data Cleaning
  - Aggregate to daily activity.
  - Create negative examples.
- Handle Missing Values
  - Create `IsMissing` features. (auto)

# Step 1: Prepare Data — But Wait, There's More!

You can spend your whole life preparing data.

- Shape Features
  - Bin numeric features. (auto)
  - Convert strings to indicator features. (auto)
  - Encode strings as numbers (counting trick). (auto)
  - Rolling window statistics.
    *What much did Bob buy/sell last 2 weeks?*
- Transform Response Variable
  - *Is Bob likely to make a high value trade next week?*
- Feature Selection (skipped)
- Dimensionality Reduction (skipped)



# Step 2: Modeling

Lots of trial & error to get best results.

- Map business problem to ML problem.
  Pr(trade | features) vs.
  *Who should I call & why?*
- Define success metric.
  - Tried: RMSE, ROC Area, Recall@K
  - Winner: *average daily hit rate*
- Try a bunch of ML algorithms. (skipped)
- Tune hyper-parameters.
  - When to stop gradient descent? (auto)
  - Grid search for good regularization. (auto)

# Step 2: Modeling

Lots of trial & error to get best results.

- Map business problem to ML problem.
  Pr(trade | features) vs.
  *Who should I call & why?*
- Define success metric.
  - Tried: RMSE, ROC Area, Recall@K
  - Winner: *average daily hit rate*
- Try a bunch of ML algorithms. (skipped)
- Tune hyper-parameters.
  - When to stop gradient descent? (auto)
  - Grid search for good regularization. (auto)

# Step 2: Modeling

Lots of trial & error to get best results.

- Map business problem to ML problem.
  Pr(trade | features) vs.
  *Who should I call & why?*
- Define success metric.
  - Tried: RMSE, ROC Area, Recall@K
  - Winner: *average daily hit rate*
- Try a bunch of ML algorithms. (skipped)
- Tune hyper-parameters.
  - When to stop gradient descent? (auto)
  - Grid search for good regularization. (auto)

# Step 3: Evaluate & Study Model

Approach depends on goal and ML algorithm.

# Step 3: Evaluate & Study Model

Approach depends on goal and ML algorithm.

Prediction Accuracy? Measure on holdout data, ask experts.
Be careful with time series data!

# Step 3: Evaluate & Study Model

Approach depends on goal and ML algorithm.

Prediction Accuracy? Measure on holdout data, ask experts.
Be careful with time series data!

Target leakage? Look for super, too-good-to-be-true features.

# Step 3: Evaluate & Study Model

Approach depends on goal and ML algorithm.

Prediction Accuracy? Measure on holdout data, ask experts.
Be careful with time series data!

Target leakage? Look for super, too-good-to-be-true features.

Justification? Annotate predictions with reason codes.

# Step 3: Evaluate & Study Model

Approach depends on goal and ML algorithm.

Prediction Accuracy? Measure on holdout data, ask experts.
Be careful with time series data!

Target leakage? Look for super, too-good-to-be-true features.

Justification? Annotate predictions with reason codes.

Plausible domain theory? (skipped)

# Step 3: Evaluate & Study Model

Approach depends on goal and ML algorithm.

Prediction Accuracy? Measure on holdout data, ask experts.
Be careful with time series data!

Target leakage? Look for super, too-good-to-be-true features.

Justification? Annotate predictions with reason codes.

Plausible domain theory? (skipped)

Extrapolation risk? (skipped)

# Step 4: Report to Stakeholders


Reproduced with permission from Glasbergen Cartoon Service.

# Step 5: Deployment

Deploying predictive analytics is a ton of work.

Used batch execution for prioritized call list deployment:

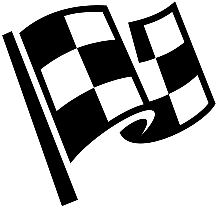
- Rebuild model daily.
- Generate updated call list hourly.
- Jobs triggered by cron-like system.
- Plumb predictions and reasons and metadata to a UI.
- Heavy customization of reason codes.
- Run book: how to install, dependency on data feeds, where are results written, how to handle errors, …

# Step 5: Deployment — Streaming Style

Deploying predictive analytics is a ton of work.

Example 2: used streaming execution for credit card fraud app:

- REST end point to get predictions.
- Latency $<$ 30ms for 99.999% of transactions.
- 99.99% uptime per data center.
- Live model updates and safety guardrails.



# Things that (Seem to) Help

# Get All Data in One Place

Everything Else:
Get Better Tooling

# Build Libraries for Common Operations

Build what you need once — not for every project.

Low hanging fruit:

- common domain transforms
- model insight tools
- quick & dirty visualization

# Build Libraries for Common Operations

Build what you need once — not for every project.

Low hanging fruit:

- common domain transforms
- model insight tools
- quick & dirty visualization

Implementation quality matters:

- 5x faster model building
  (rewrote transforms)
- 2x faster leakage diagnostic
  (caching intermediate reprs.)

# Commit to One Machine Learning Algorithm

*Algorithms sell publications.*
*Features win competitions.*

# Reduce Time to First Model

How:

- quick & dirty sub-sample
- minimize data prep, especially on features

Why:

- Many problems become obvious once you have a model.
- Many feature problems have negligible impact.

# Enable Rapid Iteration

Interactive tools reduce context switches.

Compute implications:

- scalability $\implies$ multi-core hardware
- *must* keep data in memory
- parallel or incremental algorithms

# Enable Rapid Iteration

Interactive tools reduce context switches.

Compute implications:

- scalability $\implies$ multi-core hardware
- *must* keep data in memory
- parallel or incremental algorithms

Update or rebuild?

- When you add add rows?
- When you add features?
- When you remove features?

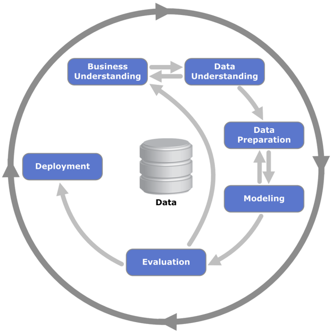



Image source: `http://bit.ly/1Nqm4yw`

# Replace Human Search with Computer Search

Computers are better than humans at search & optimization:

| OLD WAY | NEW WAY |
|---|---|
| manually set parameters | least squares regression (1821), computer solvers (1970's) |
| experts write rules | learn rules from data (1980's) |
| set hyper-parameters by intuition | grid search (1960's), stochastic optimization (2000's) |
| experts pick features | hill-climbing (1970's), LASSO (1996), AIC (2002) |
| experts transform data | advanced learning systems (now) |
| experts specify join plan | automated search (next 5 years?) |

Big wins if you embrace empiricism.

# Replace Human Search with Computer Search

Computers are better than humans at search & optimization:

| OLD WAY | NEW WAY |
|---|---|
| manually set parameters | least squares regression (1821), computer solvers (1970's) |
| experts write rules | learn rules from data (1980's) |
| set hyper-parameters by intuition | grid search (1960's), stochastic optimization (2000's) |
| experts pick features | hill-climbing (1970's), LASSO (1996), AIC (2002) |
| experts transform data | advanced learning systems (now) |
| experts specify join plan | automated search (next 5 years?) |

Big wins if you embrace empiricism.

# Replace Human Search with Computer Search

Computers are better than humans at search & optimization:

| OLD WAY | NEW WAY |
|---|---|
| manually set parameters | least squares regression (1821), computer solvers (1970's) |
| experts write rules | learn rules from data (1980's) |
| set hyper-parameters by intuition | grid search (1960's), stochastic optimization (2000's) |
| experts pick features | hill-climbing (1970's), LASSO (1996), AIC (2002) |
| experts transform data | advanced learning systems (now) |
| experts specify join plan | automated search (next 5 years?) |

Big wins if you embrace empiricism.

# Replace Human Search with Computer Search

Computers are better than humans at search & optimization:

| Old Way | New Way |
|---|---|
| manually set parameters | least squares regression (1821), computer solvers (1970's) |
| experts write rules | learn rules from data (1980's) |
| set hyper-parameters by intuition | grid search (1960's), stochastic optimization (2000's) |
| experts pick features | hill-climbing (1970's), LASSO (1996), AIC (2002) |
| experts transform data | advanced learning systems (now) |
| experts specify join plan | automated search (next 5 years?) |

Big wins if you embrace empiricism.

# Replace Human Search with Computer Search

Computers are better than humans at search & optimization:

| OLD WAY | NEW WAY |
|---|---|
| manually set parameters | least squares regression (1821), computer solvers (1970's) |
| experts write rules | learn rules from data (1980's) |
| set hyper-parameters by intuition | grid search (1960's), stochastic optimization (2000's) |
| experts pick features | hill-climbing (1970's), LASSO (1996), AIC (2002) |
| experts transform data | advanced learning systems (now) |
| experts specify join plan | automated search (next 5 years?) |

Big wins if you embrace empiricism.

# Replace Human Search with Computer Search

Computers are better than humans at search & optimization:

| OLD WAY | NEW WAY |
|---|---|
| manually set parameters | least squares regression (1821), computer solvers (1970's) |
| experts write rules | learn rules from data (1980's) |
| set hyper-parameters by intuition | grid search (1960's), stochastic optimization (2000's) |
| experts pick features | hill-climbing (1970's), LASSO (1996), AIC (2002) |
| experts transform data | advanced learning systems (now) |
| experts specify join plan | automated search (next 5 years?) |

Big wins if you embrace empiricism.

# Closing Thoughts

# A True Story

| TIME | EMAIL |
|---|---|
| 10:30a | (CEO) I'm sitting next to CTO of ⟨customer⟩ on the flight to SF. He would like to see how accurate the estimates are for predicting total spend per cost center for each customer. Can you do a quick estimate before we land at noon? ;) |

# A True Story

| TIME | EMAIL |
|---|---|
| 10:30a | (CEO) I'm sitting next to CTO of ⟨customer⟩ on the flight to SF. He would like to see how accurate the estimates are for predicting total spend per cost center for each customer. Can you do a quick estimate before we land at noon? ;) |
| 11:01a | (CEO) No pressure. But … if you do this, he will consider it amazing. |

# A True Story

| TIME | EMAIL |
|---|---|
| 10:30a | (CEO) I'm sitting next to CTO of ⟨customer⟩ on the flight to SF. He would like to see how accurate the estimates are for predicting total spend per cost center for each customer. Can you do a quick estimate before we land at noon? ;) |
| 11:01a | (CEO) No pressure. But …if you do this, he will consider it amazing. |
| 11:08a | (Scott) On it. Pulling data from Hadoop. |

# A True Story

| TIME | EMAIL |
| --- | --- |
| 10:30a | (CEO) I'm sitting next to CTO of ⟨customer⟩ on the flight to SF. He would like to see how accurate the estimates are for predicting total spend per cost center for each customer. Can you do a quick estimate before we land at noon? ;) |
| 11:01a | (CEO) No pressure. But … if you do this, he will consider it amazing. |
| 11:08a | (Scott) On it. Pulling data from Hadoop. |
| 11:20a | (Scott) Model is now training… |

# A True Story

| Time | Email |
| --- | --- |
| 10:30a | (CEO) I'm sitting next to CTO of ⟨customer⟩ on the flight to SF. He would like to see how accurate the estimates are for predicting total spend per cost center for each customer. Can you do a quick estimate before we land at noon? ;) |
| 11:01a | (CEO) No pressure. But …if you do this, he will consider it amazing. |
| 11:08a | (Scott) On it. Pulling data from Hadoop. |
| 11:20a | (Scott) Model is now training… |
| 11:21a | (CEO) How are you backtesting? |

# A True Story

| Time | Email |
|---|---|
| 10:30a | (CEO) I'm sitting next to CTO of ⟨customer⟩ on the flight to SF. He would like to see how accurate the estimates are for predicting total spend per cost center for each customer. Can you do a quick estimate before we land at noon? ;) |
| 11:01a | (CEO) No pressure. But ... if you do this, he will consider it amazing. |
| 11:08a | (Scott) On it. Pulling data from Hadoop. |
| 11:20a | (Scott) Model is now training... |
| 11:21a | (CEO) How are you backtesting? |
| 11:30a | (Scott) The model predicts each customer's daily spend by spend category. |

# A True Story

| TIME | EMAIL |
| --- | --- |
| 10:30a | (CEO) I'm sitting next to CTO of ⟨customer⟩ on the flight to SF. He would like to see how accurate the estimates are for predicting total spend per cost center for each customer. Can you do a quick estimate before we land at noon? ;) |
| 11:01a | (CEO) No pressure. But ... if you do this, he will consider it amazing. |
| 11:08a | (Scott) On it. Pulling data from Hadoop. |
| 11:20a | (Scott) Model is now training... |
| 11:21a | (CEO) How are you backtesting? |
| 11:30a | (Scott) The model predicts each customer's daily spend by spend category. |
| 11:34a | (CEO) Roll-up over the last quarter, please. |

# A True Story

| TIME | EMAIL |
|---|---|
| 10:30a | (CEO) I'm sitting next to CTO of ⟨customer⟩ on the flight to SF. He would like to see how accurate the estimates are for predicting total spend per cost center for each customer. Can you do a quick estimate before we land at noon? ;) |
| 11:01a | (CEO) No pressure. But . . . if you do this, he will consider it amazing. |
| 11:08a | (Scott) On it. Pulling data from Hadoop. |
| 11:20a | (Scott) Model is now training. . . |
| 11:21a | (CEO) How are you backtesting? |
| 11:30a | (Scott) The model predicts each customer's daily spend by spend category. |
| 11:34a | (CEO) Roll-up over the last quarter, please. |
| 11:39a | (Scott) Can it be monthly? The data sample is 6 months. |

# A True Story

| TIME | EMAIL |
|---|---|
| 10:30a | (CEO) I'm sitting next to CTO of ⟨customer⟩ on the flight to SF. He would like to see how accurate the estimates are for predicting total spend per cost center for each customer. Can you do a quick estimate before we land at noon? ;) |
| 11:01a | (CEO) No pressure. But ... if you do this, he will consider it amazing. |
| 11:08a | (Scott) On it. Pulling data from Hadoop. |
| 11:20a | (Scott) Model is now training... |
| 11:21a | (CEO) How are you backtesting? |
| 11:30a | (Scott) The model predicts each customer's daily spend by spend category. |
| 11:34a | (CEO) Roll-up over the last quarter, please. |
| 11:39a | (Scott) Can it be monthly? The data sample is 6 months. |
| 12:11p | (Scott) This is actually rolled-up overall, but here are the results in Excel. |