Data Science

You Aren't Gonna Need It

Christopher Wetherill

State Auto Insurance Companies

08 November 2018

Contents

About me

Psychologist turned cancer biologist turned data scientist

Currently Data Science Technical Lead at State Auto. (Previously at SafeAuto for maximal confusion)

Programming in Python and R for 10-ish years

Say hi: chris@tbmh.org

About this talk

We will. . .

- Put some guardrails around what data science is (and isn't);
- Discuss what makes data science distinct from traditional business intelligence and predictive modeling roles;
- Review the data science project lifecycle (with examples!); and
- Talk about why data science teams fail, and strategies for building successful ones

Some ground rules

- **No buzzwords, please.** Let's forget about the Big Data and AI hype trains for a minute
- **Questions are encouraged.** If I gloss over something or don't explain it well, interrupt me. It's cool, I promise
- **Machine learning isn't magic.** If I'm describing something like it is magical, see the previous point and call me out on it.

Contents

What makes a Data Scientist?

- Machine learning?
- Software engineering?
- Database administration?
- Distributed systems knowledge?
- Data analysis?
- Data visualization?
- Report/dashboard building?

What makes a Data Scientist?

- Machine learning?
- Software engineering?
- Database administration?
- Distributed systems knowledge?
- Data analysis?
- Data visualization?
- Report/dashboard building?

In reality, it's a grab bag

What makes a Data Scientist?

- Machine learning?
- Software engineering?
- Database administration?
- Distributed systems knowledge?
- Data analysis?
- Data visualization?
- Report/dashboard building?

In reality, it's a grab bag

What makes a Data Scientist?

- Machine learning?
- Software engineering?
- Database administration?
- Distributed systems knowledge?
- Data analysis?
- Data visualization?
- Report/dashboard building?

In reality, it's a grab bag

What makes a Data Scientist?

- Machine learning?
- Software engineering?
- Database administration?
- Distributed systems knowledge?
- Data analysis?
- Data visualization?
- Report/dashboard building?

In reality, it's a grab bag

Types of Data Scientist

- Machine Learning Engineer
  - Emphasis on machine learning, algorithm development
  - Deep theoretical understanding of ML algorithms
- Data Engineer
  - Build data pipelines, integrate disparate sources of data
  - Transform raw data into something usable by ML models
- Data Analyst
  - Provide insight to the business, summarize and communicate data to drive business decisions
  - Usually, this means lots of dashboards. Everybody loves dashboards

Types of Data Scientist

- Machine Learning Engineer
  - Emphasis on machine learning, algorithm development
  - Deep theoretical understanding of ML algorithms
- Data Engineer
  - Build data pipelines, integrate disparate sources of data
  - Transform raw data into something usable by ML models
- Data Analyst
  - Provide insight to the business, summarize and communicate data to drive business decisions
  - Usually, this means lots of dashboards. Everybody loves dashboards

Types of Data Scientist

- Machine Learning Engineer
  - Emphasis on machine learning, algorithm development
  - Deep theoretical understanding of ML algorithms
- Data Engineer
  - Build data pipelines, integrate disparate sources of data
  - Transform raw data into something usable by ML models
- Data Analyst
  - Provide insight to the business, summarize and communicate data to drive business decisions
  - Usually, this means lots of dashboards. Everybody loves dashboards

Types of Data Scientist

- Machine Learning Engineer
  - Emphasis on machine learning, algorithm development
  - Deep theoretical understanding of ML algorithms
- Data Engineer
  - Build data pipelines, integrate disparate sources of data
  - Transform raw data into something usable by ML models
- Data Analyst
  - Provide insight to the business, summarize and communicate data to drive business decisions
  - Usually, this means lots of dashboards. Everybody loves dashboards

Understand your organization's needs!

Okay. . . but what's data *science*?

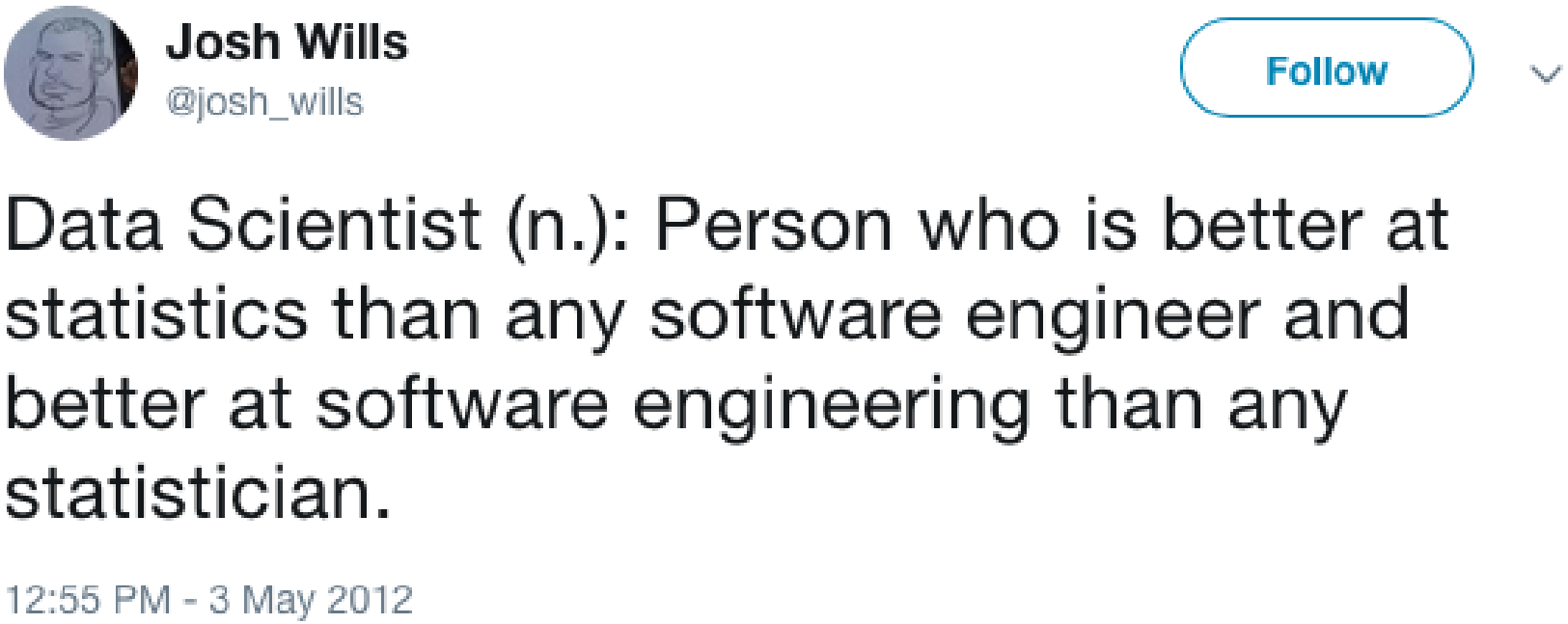


Figure 1: Twitter knows what's up. (Source: @josh_wills)

Okay. . . but what's data *science*?

Data science is about finding answers to *known unknowns*

Okay. . . but what's data *science*?

In practice, this can mean a *lot* of different things:

- Generate and test hypotheses; run experiments
- Build data pipelines; synthesize disparate data from multiple sources
- Develop, validate, and optimize algorithms
- Build statistical models to describe relationships among data
- Frame mathematical relationships in terms of business outcomes; tell stories from data

Data science is defined by its outcomes

Data Science: The practice of using "multidisciplinary methods to understand and have a positive impact on a business process or product"[1]

[1] Skipper Seabolt, *Introduction to Python for Data Science*.

So, to recap. . .

- "Data Science" is just a methodology for finding answers to hard questions
- Data scientists come in different flavors, but are fundamentally multidisciplinary
- Data science doesn't exist in a vacuum, and any good data scientist should have deep domain knowledge and work closely with business partners

Contents

The data science lifecycle

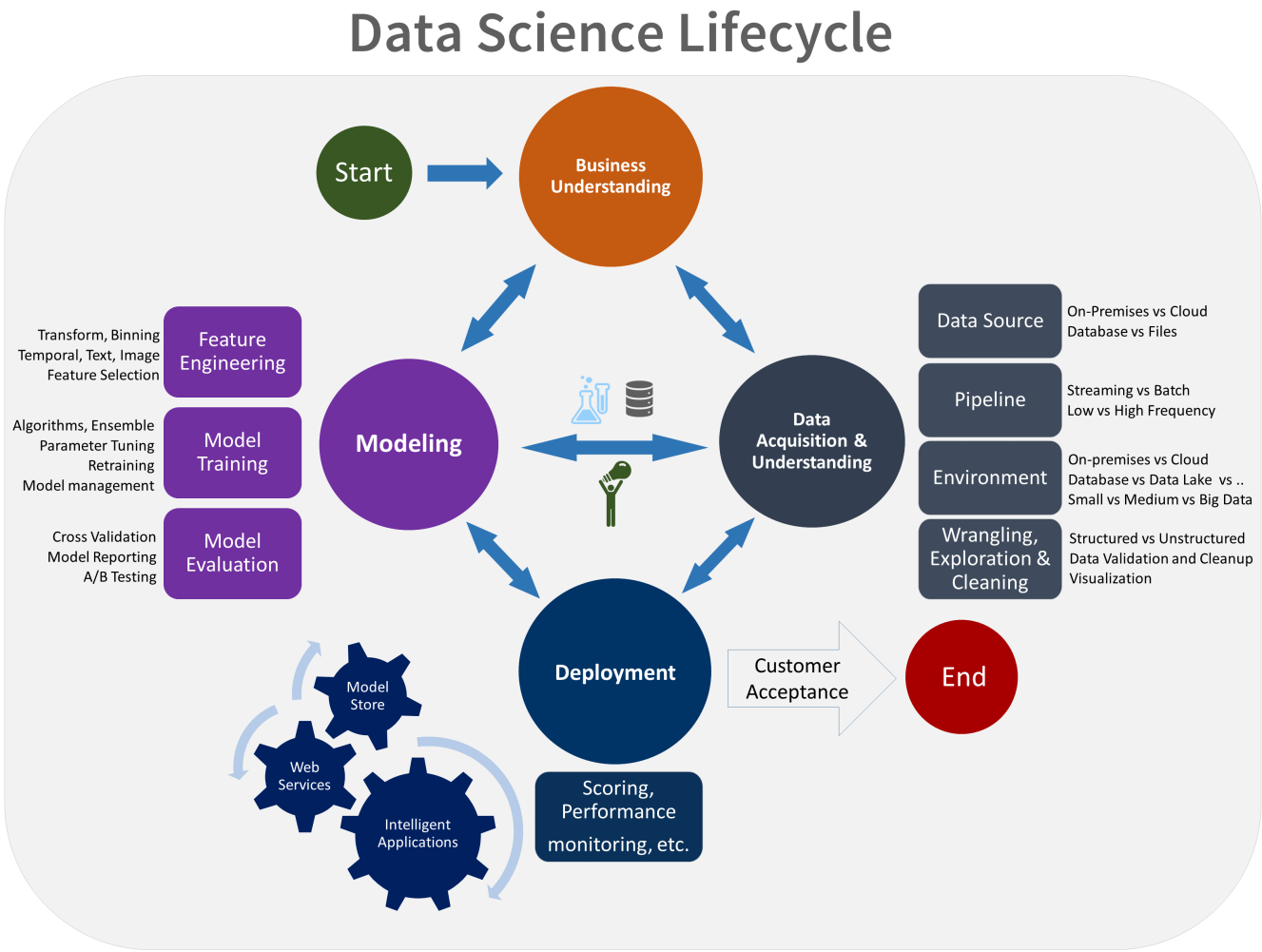


Figure 2: Someone thought this was a good idea. (Source: Microsoft)

The data science lifecycle

- Find a business problem. They're everywhere. Usually in the form, "I wish I knew why X was happening"
- Insert yourself in the conversation. Be relentlessly cheery and helpful. Maybe pull some data to help the business (and yourself) clarify their understanding of the problem
- Build a PoC. It'll be glued together, but it's a quick (within a couple weeks) indication of a project's viability
- Build an MVP. Deploy the model behind the scenes and score live data with it to observe how it performs
- Productionalize. Build out user interfaces, alter automated processes to react to the model's predictions

The data science lifecycle

- Find a business problem. They're everywhere. Usually in the form, "I wish I knew why X was happening"
- Insert yourself in the conversation. Be relentlessly cheery and helpful. Maybe pull some data to help the business (and yourself) clarify their understanding of the problem
- Build a PoC. It'll be glued together, but it's a quick (within a couple weeks) indication of a project's viability
- Build an MVP. Deploy the model behind the scenes and score live data with it to observe how it performs
- Productionalize. Build out user interfaces, alter automated processes to react to the model's predictions

The data science lifecycle

- Find a business problem. They're everywhere. Usually in the form, "I wish I knew why X was happening"
- Insert yourself in the conversation. Be relentlessly cheery and helpful. Maybe pull some data to help the business (and yourself) clarify their understanding of the problem
- Build a PoC. It'll be glued together, but it's a quick (within a couple weeks) indication of a project's viability
- Build an MVP. Deploy the model behind the scenes and score live data with it to observe how it performs
- Productionalize. Build out user interfaces, alter automated processes to react to the model's predictions

The data science lifecycle

- Find a business problem. They're everywhere. Usually in the form, "I wish I knew why X was happening"
- Insert yourself in the conversation. Be relentlessly cheery and helpful. Maybe pull some data to help the business (and yourself) clarify their understanding of the problem
- Build a PoC. It'll be glued together, but it's a quick (within a couple weeks) indication of a project's viability
- Build an MVP. Deploy the model behind the scenes and score live data with it to observe how it performs
- Productionalize. Build out user interfaces, alter automated processes to react to the model's predictions

The data science lifecycle

- Find a business problem. They're everywhere. Usually in the form, "I wish I knew why X was happening"
- Insert yourself in the conversation. Be relentlessly cheery and helpful. Maybe pull some data to help the business (and yourself) clarify their understanding of the problem
- Build a PoC. It'll be glued together, but it's a quick (within a couple weeks) indication of a project's viability
- Build an MVP. Deploy the model behind the scenes and score live data with it to observe how it performs
- Productionalize. Build out user interfaces, alter automated processes to react to the model's predictions

Not pictured here...

There's a lot of boring, detailed work that has to go into each of those steps. The bulk of this is usually consumed by:

- Mapping, documenting, and querying data
- Discovering that half of the features you're looking at aren't consistently collected
- Tracking down the business process changes that impact the features you care about
- Getting consensus on which of these 5 different technical definitions we want to use for a common business metric
- Engineering, re-engineering, and re-re-engineering features

A real-world example: Claims fraud

What is it?

Auto claims fraud is any case where an insured submits a claim either that is false, or the details of which are exaggerated or substantially altered in order to receive a larger monetary payout

Why do we care?

At about $90,000$ COMP/COLL/PD claims annually with an average severity of $\$2,000$, we're looking at an annual savings of around \$4MM

A real-world example: Claims fraud

What is it?

Auto claims fraud is any case where an insured submits a claim either that is false, or the details of which are exaggerated or substantially altered in order to receive a larger monetary payout

Why do we care?

At about $90,000$ COMP/COLL/PD claims annually with an average severity of $\$2,000$, we're looking at an annual savings of around \$4MM

Define the business problem

What's the ultimate goal here?

- To catch every fraudulent claim?
- To minimize overall costs?
- To provide the best claims experience to our claimants?
- To reduce the number of valid claims we have our Special Investigations Unit look into?

Define the business problem

What's the ultimate goal here?

- To catch every fraudulent claim?
- To minimize overall costs?
- To provide the best claims experience to our claimants?
- To reduce the number of valid claims we have our Special Investigations Unit look into?

Ultimately: minimize overall costs while maximizing the proportion of truly fraudulent claims we refer to investigation

Define a starting point

Compare against industry standards

On average, about 2% of all auto claims are found to be fraudulent (though more go undetected) — this should be our target to measure successful referral rates against.

Look at current performance

We underperform relative to industry standards: we refer very few claims to SIU and overall deny less than 1% of claims each year on the basis of fraud.

Look at current process

Our claim referral process is a mix of hardcoded business logic and human intuition. We don't have any feedback mechanisms to measure which rules and heuristics are effective and which aren't.

Define a starting point

Compare against industry standards

On average, about 2% of all auto claims are found to be fraudulent (though more go undetected) — this should be our target to measure successful referral rates against.

Look at current performance

We underperform relative to industry standards: we refer very few claims to SIU and overall deny less than 1% of claims each year on the basis of fraud.

Look at current process

Our claim referral process is a mix of hardcoded business logic and human intuition. We don't have any feedback mechanisms to measure which rules and heuristics are effective and which aren't.

Define a starting point

Compare against industry standards

On average, about 2% of all auto claims are found to be fraudulent (though more go undetected) — this should be our target to measure successful referral rates against.

Look at current performance

We underperform relative to industry standards: we refer very few claims to SIU and overall deny less than 1% of claims each year on the basis of fraud.

Look at current process

Our claim referral process is a mix of hardcoded business logic and human intuition. We don't have any feedback mechanisms to measure which rules and heuristics are effective and which aren't.

Scope a PoC

Deliverable: A model that can take claim characteristics available at FNOL (when we first learn about the incident) and identify which were submitted fraudulently at a rate as good as or better than our current process

	Predicted	
Observed	Fraud	Not Fraud
Fraud	0.5%	1.5%
Not Fraud	3.0%	95.0%

Some initial considerations

- Accurate predictions rely on correctly-labeled data. We have a lot of fraud that has slipped through the cracks!
- We can't just refer a boatload more claims to SIU: that's a horrible experience for insureds who just went through a traumatic experience
- This is one case where we don't really want to just wait for more data to come in: we need to be able to model on what we already have

Some initial considerations

- Accurate predictions rely on correctly-labeled data. We have a lot of fraud that has slipped through the cracks!
- We can't just refer a boatload more claims to SIU: that's a horrible experience for insureds who just went through a traumatic experience
- This is one case where we don't really want to just wait for more data to come in: we need to be able to model on what we already have

Some initial considerations

- Accurate predictions rely on correctly-labeled data. We have a lot of fraud that has slipped through the cracks!
- We can't just refer a boatload more claims to SIU: that's a horrible experience for insureds who just went through a traumatic experience
- This is one case where we don't really want to just wait for more data to come in: we need to be able to model on what we already have

So! Let's talk data

So! Let's talk data

Um. . . what data?

Alright, Plan B: Let's make some data!

We need:

- Some labelled auto claims fraud data
- A way to efficiently label our own historical data as fraud/not fraud

Data science's secret weapons: Google and academic research papers. Get excited!

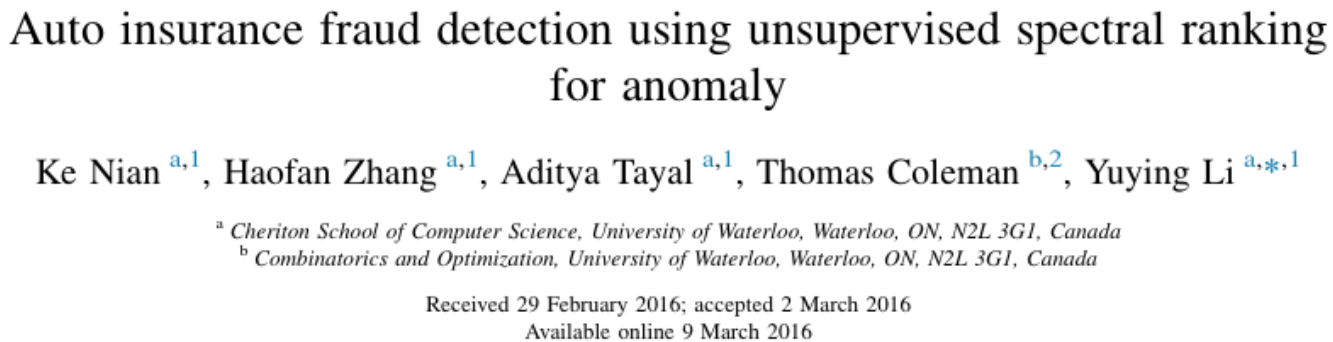
Alright, Plan B: Let's make some data!

We need:

- Some labelled auto claims fraud data
- A way to efficiently label our own historical data as fraud/not fraud

Data science's secret weapons: Google and academic research papers. Get excited!

Alright, Plan B: Let's make some data!



Auto insurance fraud detection using unsupervised spectral ranking for anomaly

Ke Nian [a,1], Haofan Zhang [a,1], Aditya Tayal [a,1], Thomas Coleman [b,2], Yuying Li [a,*,1]

[a] Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada
[b] Combinatorics and Optimization, University of Waterloo, Waterloo, ON, N2L 3G1, Canada

Received 29 February 2016; accepted 2 March 2016
Available online 9 March 2016

Figure 3: Wait. . . could it be this easy?. (Source: doi: 10.1016/j.jfds.2016.03.001)

Spectral Ranking for Anomaly Detection

The basic gist: Look at how dissimilar each observation in a dataset is from every other observation to come up with an anomaly score for each record. Larger scores probably mean something weird's going on.

The implication: We can automatically generate relatively accurate labels for our fraudulent data!

Spectral Ranking for Anomaly Detection

Great! We might have a path forward. We just need to:

- Get the original claims data referenced in the paper
- Grok the math
- Turn the math into a working Python implementation
- Replicate the authors' results
- Extend to our data and validate against the claims we know were actually fraudulent

Spectral Ranking for Anomaly Detection

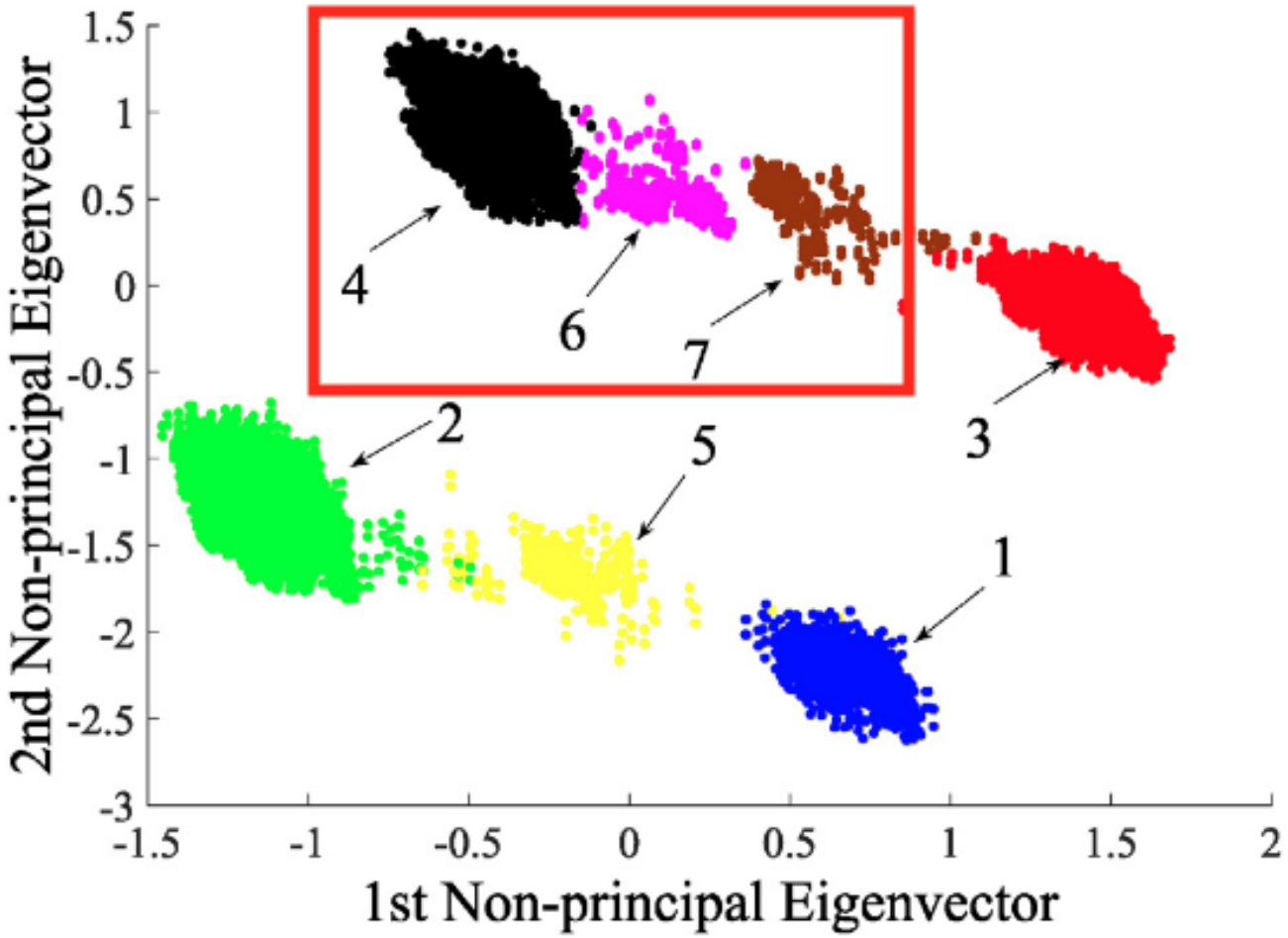


Figure 4: Distinct clusters identified by SRA. Highlighted clusters (4, 6, 7) are regions of high fraud.

Feature engineering

The application of domain knowledge to raw data

Feature engineering in context

- How long was the delay between the accident and the time it was reported to us?
- Was theft or a fire involved? Were there third party witnesses?
- Did this occur just after the insured's policy began? Just before it ended?
- Did this occur in New York or New Jersey? (No, seriously)

Neural networks as logistic regressions

Neural networks as logistic regressions

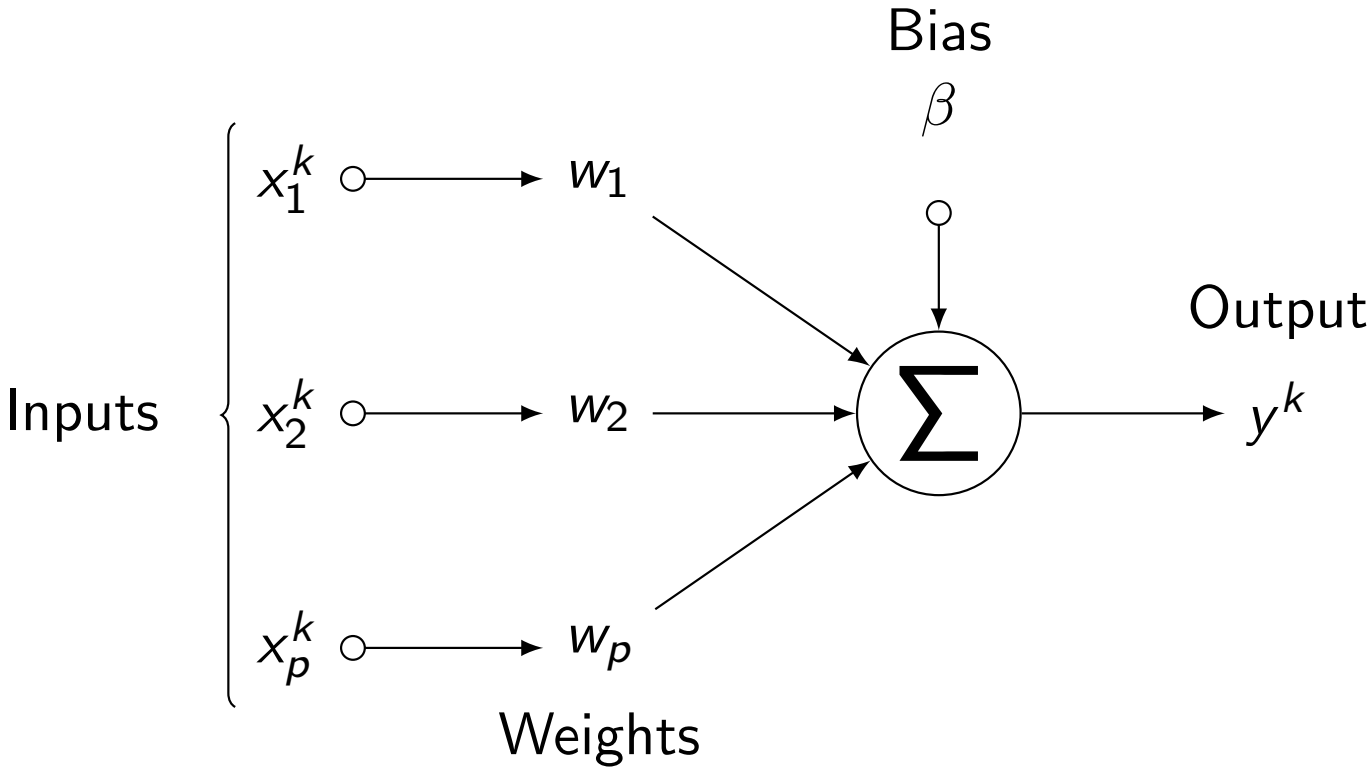
A linear model assumes a linear relationship between its dependent variable, y, and its p-vector of regressors, x. The relationship between the regressor and the dependent variable is denoted by

$$
\begin{aligned}
y^k &= w_1 \cdot x_1^k + \ldots + w_p \cdot x_p^k + \beta \\
y^k &= \sum_{i=1}^{p} (w_i \cdot x_i^k) + \beta \\
y^k &= \vec{w} \cdot \vec{x}^k + \beta
\end{aligned} \tag{1}
$$

where here w_p denotes a real-valued coefficient (a weight) and β denotes the error term, or bias. For future notation, we will consider

$$\beta = w_0 \cdot 1$$

Neural networks as logistic regressions



Wide networks

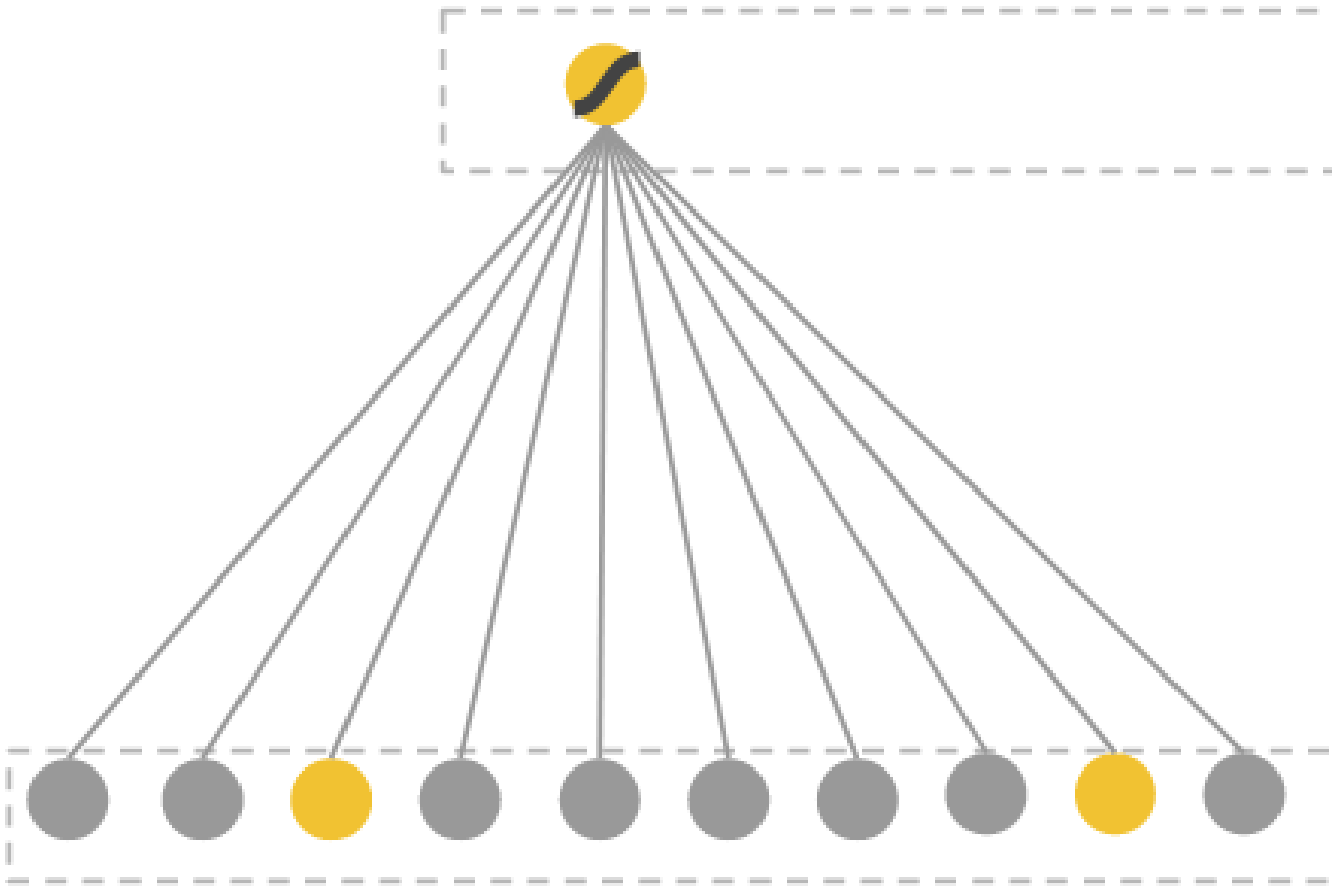


Figure 5: Wide neural network. (Source: arXiv:1606.07792)

Building a wide network

```
$ python -m trainer \
  --train-file ./data/train.csv \
  --eval-file ./data/test.csv \
  --num-epochs 75 \
  --model-type wide
```

(This abstracts a couple thousand lines of code. If you want more detail, let's chat after)

Deep networks

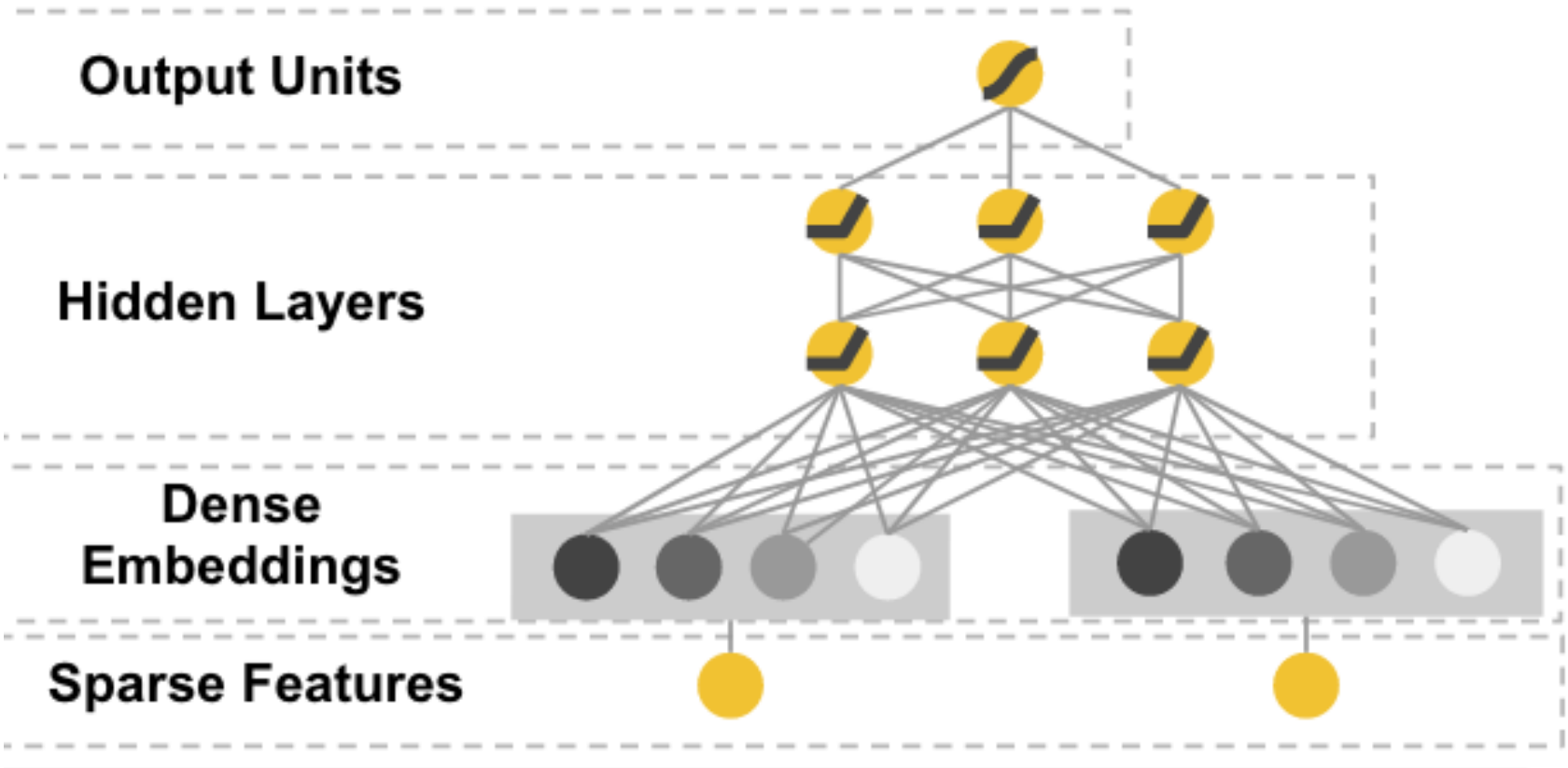


Figure 6: Deep neural network. (Source: arXiv:1606.07792)

Building a deep network

```
$ python -m trainer \
  --train-file ./data/train.csv \
  --eval-file ./data/test.csv \
  --num-epochs 75 \
  --num-layers 3 \
  --hidden-units 1024,512,256 \
  --layer-sizes-scale-factor 0 \
  --model-type deep
```

Wide-and-deep networks

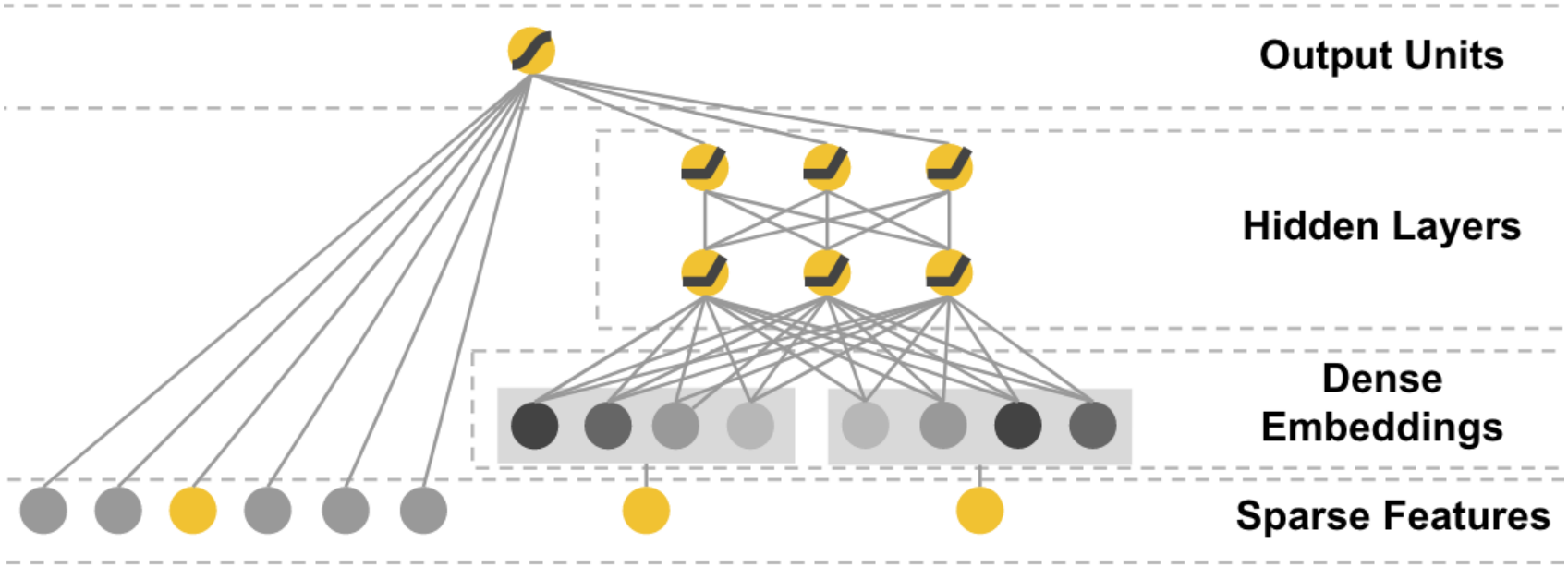


Figure 7: Wide-and-deep recommender system. (Source: arXiv:1606.07792)

Building a wide-and-deep network

```
$ python -m trainer \
  --train-file ./data/train.csv \
  --eval-file ./data/test.csv \
  --num-epochs 75 \
  --num-layers 3 \
  --hidden-units 1024,512,256 \
  --layer-sizes-scale-factor 0 \
  --model-type deepwide
```

Evaluating model performance

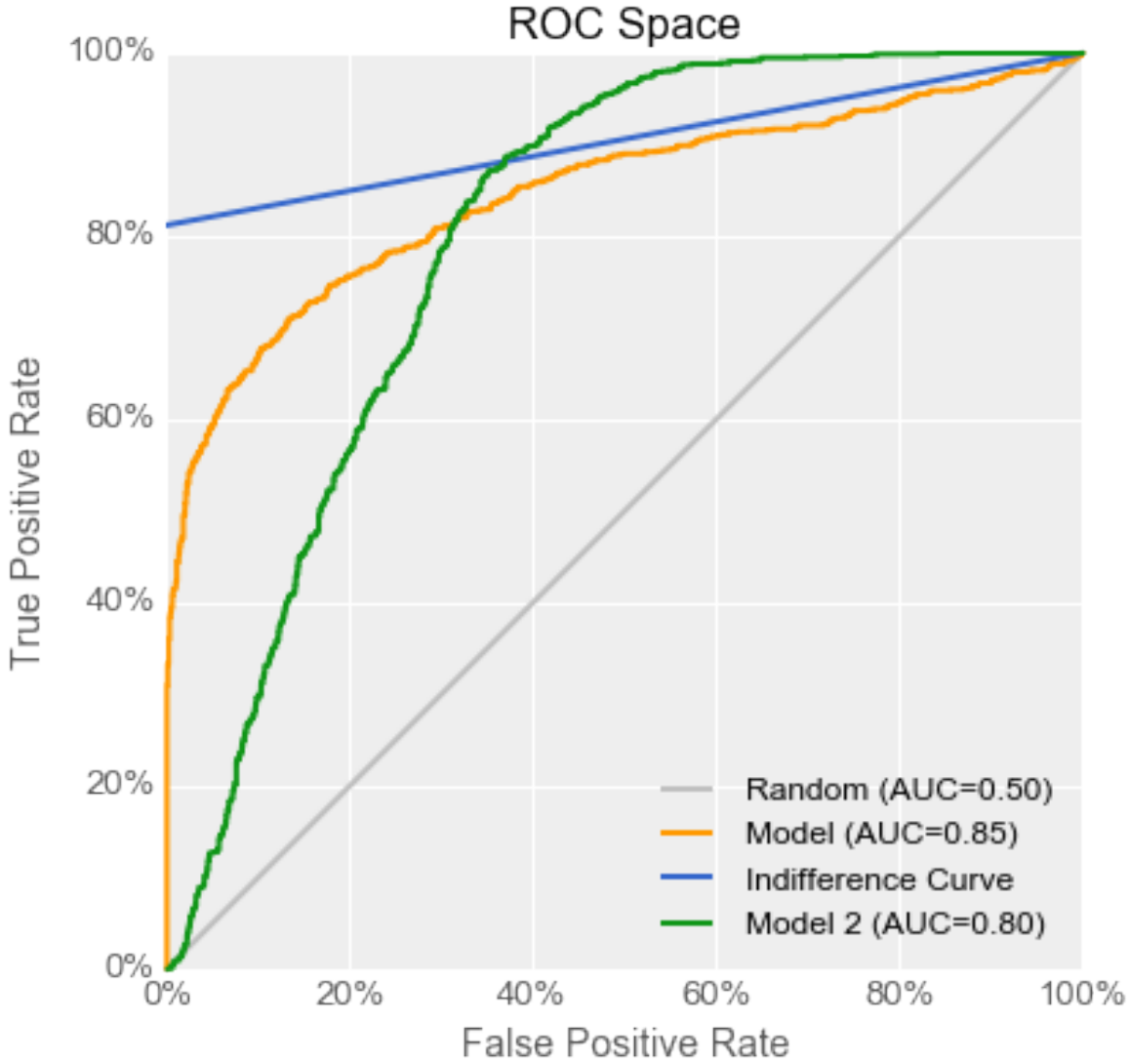


Figure 8: Indifference curve in ROC space. (Source: Nicolas Kruchten)

Deploying a model

We'll need to...

- Export the model weights;
- Deploy a webservice for online predictions;
- Put a RESTful API around the webservice;
- Allow for asset versioning, A/B testing, model performance monitoring, canary service monitoring, incremental rollouts; and
- Coordinate with other teams to build any additional application integrations

It's just that easy!

Contents

So what's the catch?

So what's the catch?

Most data science teams fail.

Wait, hold up. . . What?

Nick Heudecker
@nheudecker

Follow

We were too conservative. The failure rate is closer to 85%. And the problem isn't technology.

> **Ronda Swaney** @RondaSwaney
> Gartner says in 2017, 60% of #bigdata projects will fail to move past preliminary stages. Does that mean big data has been a failure? No. But it does mean we have to separate hype from reality. Read my latest written for @Hortonworks. hortonworks.com/article/what-h… via @Hortonworks

3:25 PM - 9 Nov 2017

Figure 9: Technology can't prevent failures in strategy. (Source: @nheudecker)

We can't blame bad data for everything

Usually there are a few scenarios:

- 'Data Science' becomes code for 'business intelligence & reporting'
- The data science team takes on unrealistic projects with outlandish timelines
- Organizational resistance to change kills models before they ever make it to production
- Lack of consistent buy-in across the business
- DS team lacks skills for end-to-end ownership of projects; leads to 'not my job' syndrome

We can't blame bad data for everything

At the end of the day, data science is just really freaking hard

Data Science: You aren't gonna need it

Is the moral of this story that data science is overblown? That you're better off without it?

Data Science: You aren't gonna need it

Is the moral of this story that data science is overblown? That you're better off without it?

No — but your organization needs to have a clear understanding of what data science is and is not

When is data science appropriate?

Do you already have basic reporting on foundational business metrics?

Does the business agree that automated, data-driven decisioning is important?

Are you willing to give a team the autonomy to investigate hard questions that they might not be able to answer?

Build a well-rounded team

The people you hire are critical. Full stop

Skills matter, but so does team dynamic

Build relationships across the organization

The best data scientists live and experience every single other part of the organization. To be effective, you need to understand how data are collected and used, and you need a deep understanding of the business context around the work that you're doing

The more the business is involved, understood, and treated as an equal collaborator, the more effective a data science team can be

Serve the right customers

Business users are your collaborators, but almost never your customers.

If your team treats others humans as the consumers of the models you build, the outcome of any project will be a god-awful PowerPoint. If you treat applications as the consumer, the end product will be an algorithm or API integrated into the engineering stack. This is how you fundamentally change the operation of the business

Serve the right customers

Business users are your collaborators, but almost never your customers.

If your team treats others humans as the consumers of the models you build, the outcome of any project will be a god-awful PowerPoint. If you treat applications as the consumer, the end product will be an algorithm or API integrated into the engineering stack. This is how you fundamentally change the operation of the business

Assume full ownership of project delivery

- ETL
- Analysis
- Code
- Deployment
- SLAs & monitoring
- Auditing

Accurately estimate timelines and risks

Model-building is nebulous. It's hard to put detailed timelines around it. That isn't an excuse for no timelines

The converse is also true: if managers/senior leaders have unrealistic expectations, address these

Accurately estimate timelines and risks

Model-building is nebulous. It's hard to put detailed timelines around it. That isn't an excuse for no timelines

The converse is also true: if managers/senior leaders have unrealistic expectations, address these

The tools don't matter until they do

You're not Google. Or Facebook. Or LinkedIn. Or Twitter. You don't have petabytes of data. Off-the-shelf tools will probably work just fine

But a commitment to open source can still be helpful.

So what's next?

Data science is complicated, it's expensive, it has a high failure rate, and I don't have any easy answers

If you're considering starting a data science team, do your due diligence first. Understand what these teams do and how they fit into your organization

Thank you!

Get in touch:
chris@tbmh.org

Download the slides:
https://go.c18l.org/u/dama

Questions?

Neural networks as logistic regressions

We can then extend the linear model form to the case of a logistic regression, or, by altering the link function and assumed probability distribution, to any in the family of generalized linear models. In the case of a logistic model, it is represented by

$$
\begin{aligned}
y^k &= ln\left(\frac{\hat{p}^k}{1-\hat{p}^k}\right) \\
\hat{p}^k &= \frac{e^{\vec{w}\cdot\vec{x}^k+\beta}}{1+e^{\vec{w}\cdot\vec{x}^k+\beta}} \\
&= \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}^k+\beta)}}
\end{aligned}
\tag{2}
$$

Neural networks as logistic regressions

This concept extends to a neural network with respect to a neuron's activation function. In the simple case of a sigmoid neuron, the activation function is equivalent to the log link function above and is represented as

$$\sigma(t) = \frac{1}{1 + e^{(-t)}} \tag{3}$$

This non-linear transformation is applied after the summation of the Hadamard Product of all inputs and weights.

Neural networks as logistic regressions