NYU FRE 7773 - Fall 2023

Machine Learning and Finance

Prof. Jacopo Tagliabue and Ethan Rosenthal

Your teachers

About us

- Dr. Jacopo Tagliabue, joint Ph.D. in Logic and Cognitive Sciences (UNISR, MIT).
- Founder of Tooso (acquired by TSX:CVO), now founder of Bauplan.
- Open source contributor and ML / NLP researcher.

- Dr. Ethan Rosenthal Ph.D. in Physics from Columbia University.
- AI Lead at Square working on NLP, formerly on Risk.
- Prior to Square, data science consulting and ML at two ecommerce startups.

Your course

FRE Fall 2023: Overview

- “This course is an introduction to Machine Learning concepts and Machine Learning Operations (MLOps) best practices, with applications to the financial industry.” (from your Syllabus!)

FRE Fall 2023: Overview

- “This course is an **introduction** to **Machine Learning** concepts and **Machine Learning** Operations (MLOps) best practices, with **applications** to the financial industry.” (from your Syllabus!)
  - *Introduction*: we assume only that you are comfortable with Python and basic quantitative skills, you are passionate about the topic and you like to spend time “hacking” away on your laptop.
  - *Machine Learning* (best practices): we leverage almost 20 years of combined experience in the industry to teach you **Dos and Don'ts** when setting up an ML project.
  - *MLOps* (best practices): we teach you how your ML model fits into the broader context of how software is deployed in the real-world, providing you first-hand experience with tools and ideas that will help you stand out in the job market.
  - *Applications*: this is a *practical* course, and you will be judged mostly by the *artifacts* you produce (typically working code). We always emphasize the practical significance of what we teach and encourage you to “live and breath” the material by participating in class and using your time to iterate on your coding skills.

FRE Fall 2023: Housekeeping

- Your TA is: **Prajwal Pitlehra** (who attended FRE 2022!). Please refer to him for any questions about scheduling, homework, etc.
- While we strive to provide as much context as we can with slides, please note that the code and our live lectures / commentary will add a ton of useful information that makes your life easier. Class attendance is **mandatory for this course**: more importantly, the more you participate, the more you will get out of this class.
- An essential part of our teaching is about **tooling**, as a good setup is 50% of your productivity. In particular, your *TA will share soon a Google Sheet*: please make sure to fill it with the required information so that we can give you access to GitHub, Comet, AWS etc. (if you don't have a GitHub account, make one for free today).
- **(Basically) all materials for this class are shared through GitHub: basic Git commands are an industry standard, and you should get comfortable working with them. It starts at day 1!**

FRE Fall 2023: Evaluation

- Your grade will be based on the grades from **six** assignments throughout the course, weighted by a point system (some homework are harder, longer etc.).
- Your final is a *ML team project*, so you will have the chance of working with your teammates on an interesting challenge that could immediately become a portfolio project as well. **The more effort and passion you put into it, the more it will help you also after the course is done.**
- Generally speaking, we try to be pretty light on homework, as we want you to spend as much time as possible on your project.
- By week 7, we require everybody to have a final project confirmed with us (with final teams, 3 people max / team): at week 14 you will all “pitch” your project to the class with a final presentation and a *live demo*.
    - Read more here
    - Your TA will circulate some GDocs so you can start putting some ideas down

FRE Fall 2023: Evaluation

To give you some context, in the last few years we had projects such as:

- A recommender system for movies
- A model to predict the price of used cars
- A model trying to forecast Bitcoin, based on other crypto assets
- A sentiment analysis model on financial news

If we (we + you!) do our job properly, by the end of this class you should be able to:

- pick an appropriate ML technique for your problem;
- appreciate the subtleties involved in a ML project, and know how the pieces fit together;
- build a small, end-to-end project, and expose it to the world through the cloud.

What FRE 2023 is NOT

1. **A theory-heavy course**: we do *some* theory of course, but we typically emphasize an intuitive and pragmatic understanding of machine learning techniques.
2. **An all-encompassing *ML in Finance* course**: we discuss *some* topics in ML based on 1) our opinionated view of what is important / feasible to teach, 2) relevance to your curriculum and to what the job market demands.
    - Note: we will discuss important finance applications (fraud, text understanding, time series), but we focus on *transferable skills* that you are unlikely to pick up elsewhere in your Master!
    - Note 2: we will have several guest lecturers (from tech and finance) to give you the chance of asking focused questions on specific topics.
3. **A software engineering course**: as part of the first few weeks, we will try and teach you some basic notions of software engineering (as we are all about *ML that works*). However, we won't have time to teach *everything explicitly*; we expect you to spend time on your own to tinker with the code, explore the additional readings (like this!) and Google your way out of programming issues (like all professionals do!)

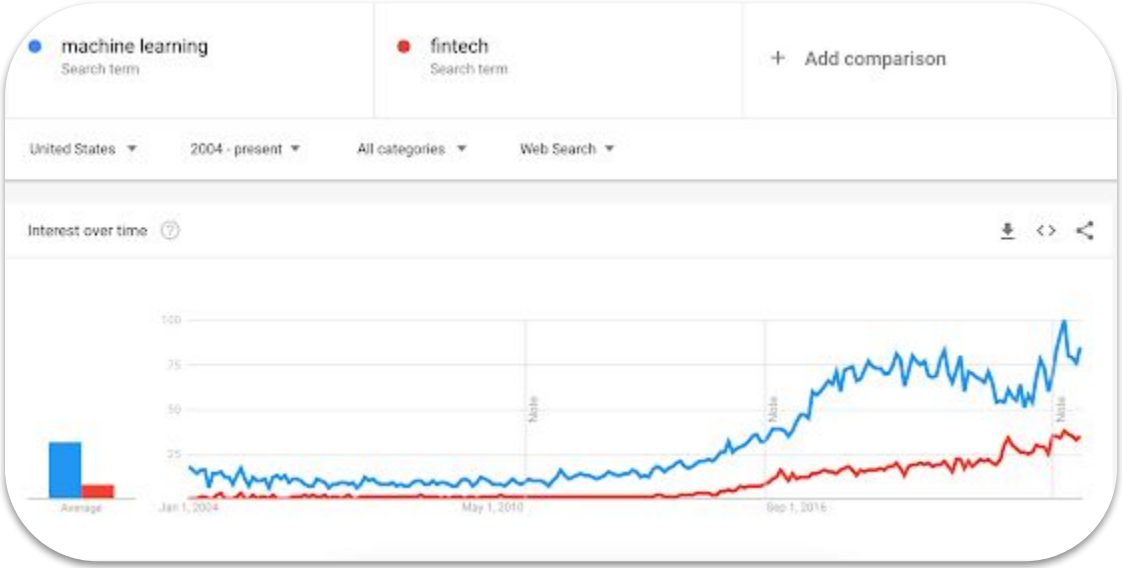
What FRE 2023 is NOT

As a general rule, there is a lot of excellent educational material on ML, so we (mostly) spend our time discussing what fewer people are teaching.

ML, Finance, MLOps

The golden era of Machine Learning

- **Fact**: this is a fantastic moment to be in Machine Learning!

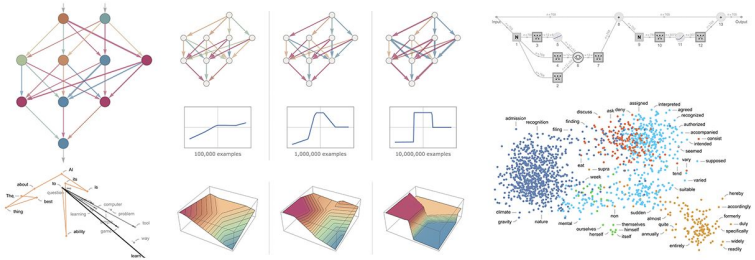


The golden era of ~~Machine Learning~~ NLP

- **Fact**: this is a fantastic moment to be in ~~Machine Learning~~ NLP

What Is ChatGPT Doing ... and Why Does It Work?

February 14, 2023



It's Just Adding One Word at a Time

That ChatGPT can automatically generate something that reads even superficially like human-written text is remarkable, and unexpected. But how does it do it? And why does it work? My purpose here is to give a rough outline of what's going on inside ChatGPT—and then to explore

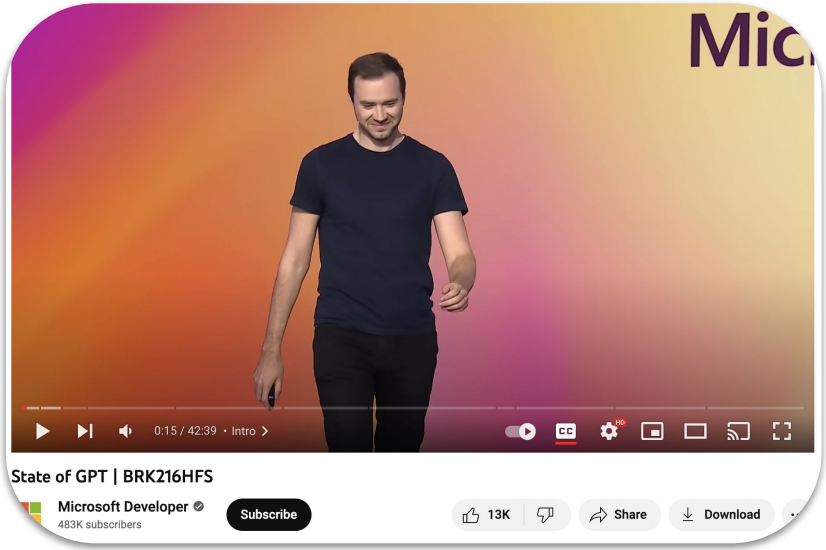
The Brilliance and Weirdness of ChatGPT

A new chatbot from OpenAI is inspiring awe, fear, stunts and attempts to circumvent its guardrails.

Share full article 422

The era of Large Language Models

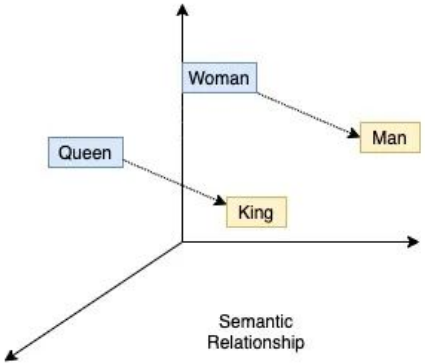
- **Large language models** may be the next big thing!



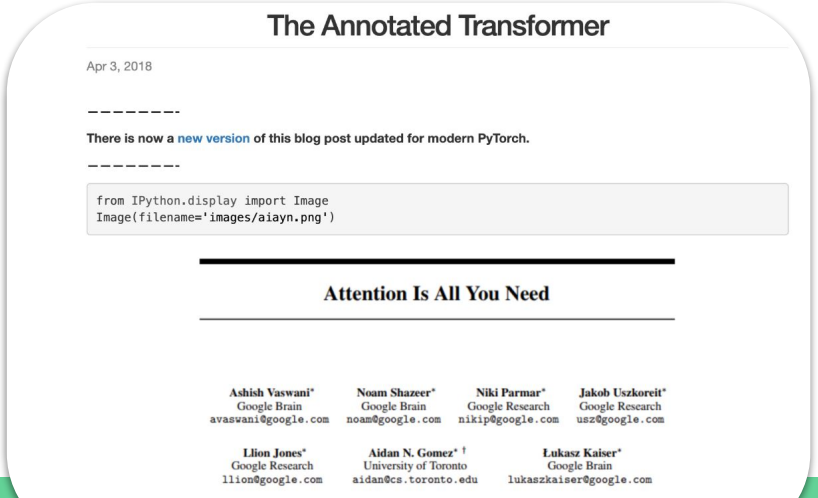
The era of Large Language Models

- **Large language models** are based (sort of) on two simple ideas:

Words are dense vectors



Neural network are good with sequences



The Annotated Transformer

Apr 3, 2018

There is now a new version of this blog post updated for modern PyTorch.

```
from IPython.display import Image
Image(filename='images/aiayn.png')
```

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

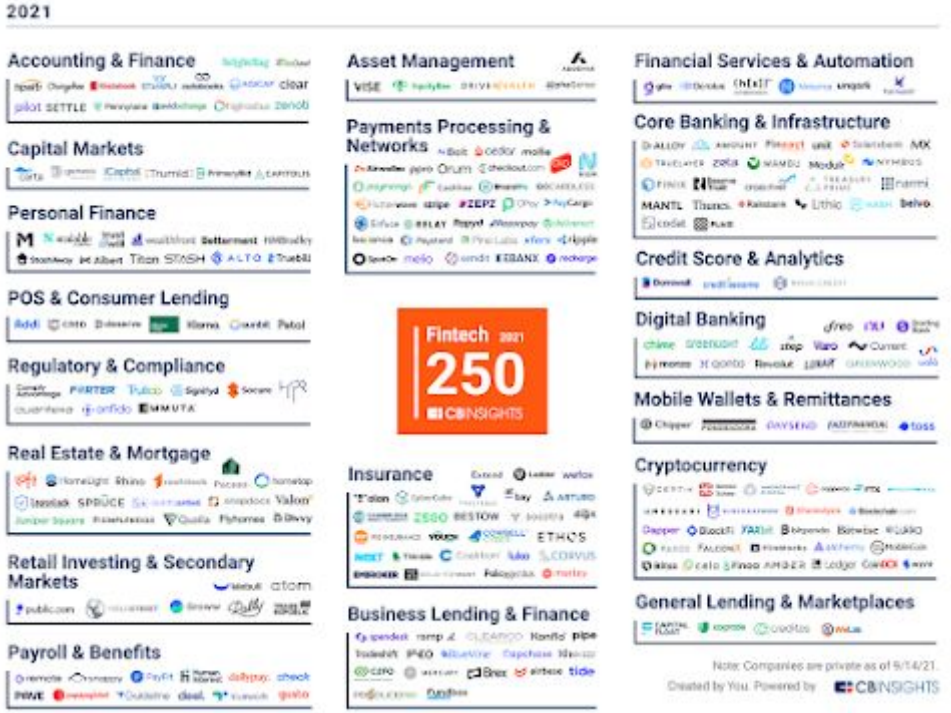
Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

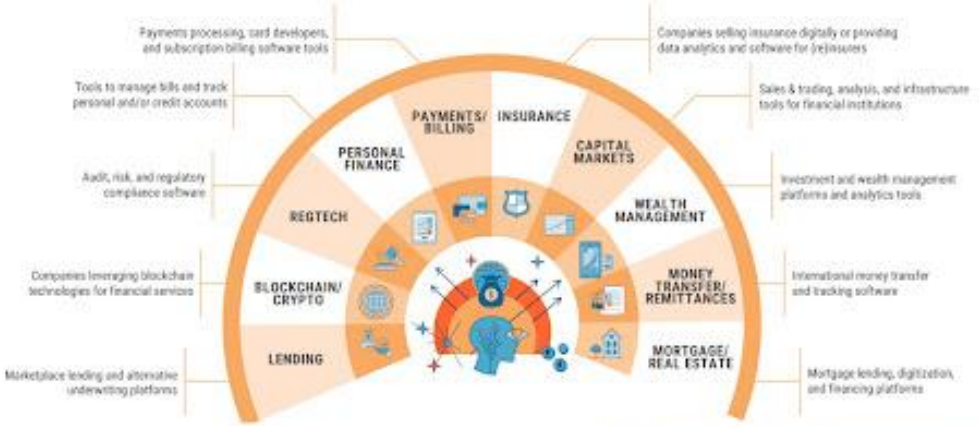
The intersection of tech and finance

- **FinTech**: “Technologically enabled financial innovation that could result in new business models, applications, processes or products with an associated material effect on financial markets and institutions and the provision of financial services”



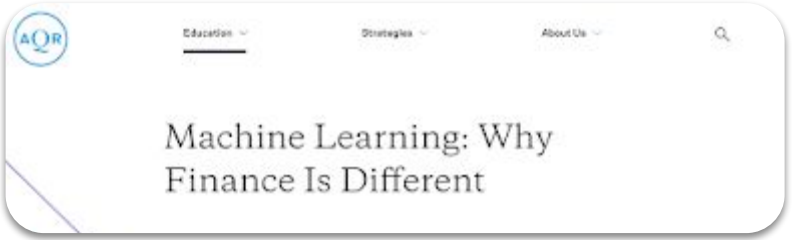
The intersection of tech and finance

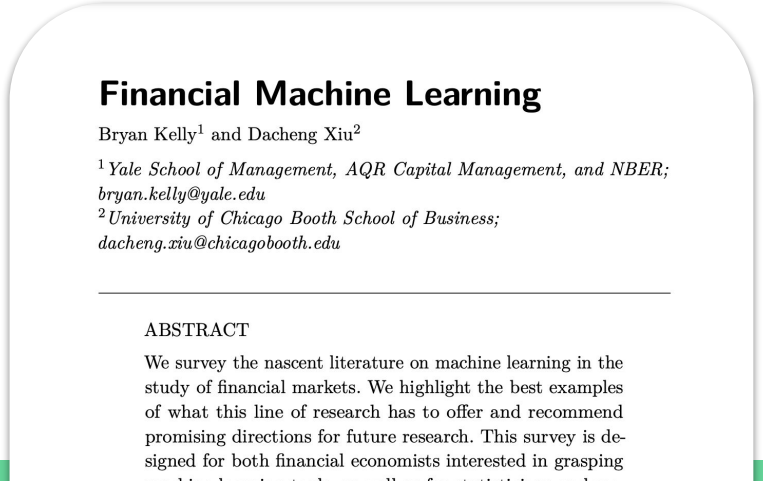
- **Who is doing it?**
  - **Everybody**, both startups and large institutions need to innovate - compared to other markets, incumbents have regulatory and financial advantages. Examples: Stripe, JPMorgan, etc.
- **Where?**
  - **Everywhere**, as more and more tech firms offer services that have usually been offered by traditional financial institutions. Examples: blockchain, stock prediction, payment systems, fraud detection etc.



The intersection of tech and finance

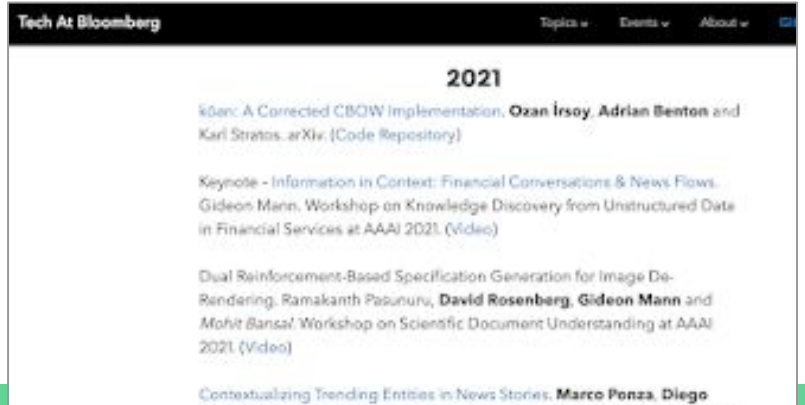
- **How?**
  - A lot of the basic ML concepts we will teach you are important across the entire spectrum of applications.
  - That said, each industry has some peculiarities, which may impact some of the ML choices you will end up making: for example, if *interpretability* is important for regulatory reasons, you need to choose an appropriate modelling technique (or be prepared to defend your choice!).

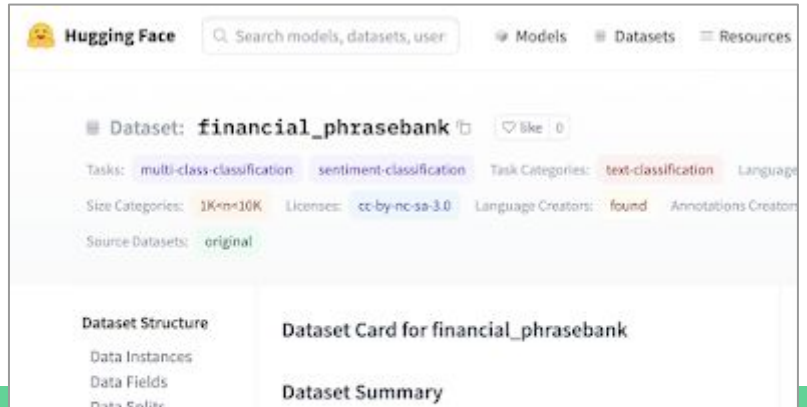




Stories from the trenches

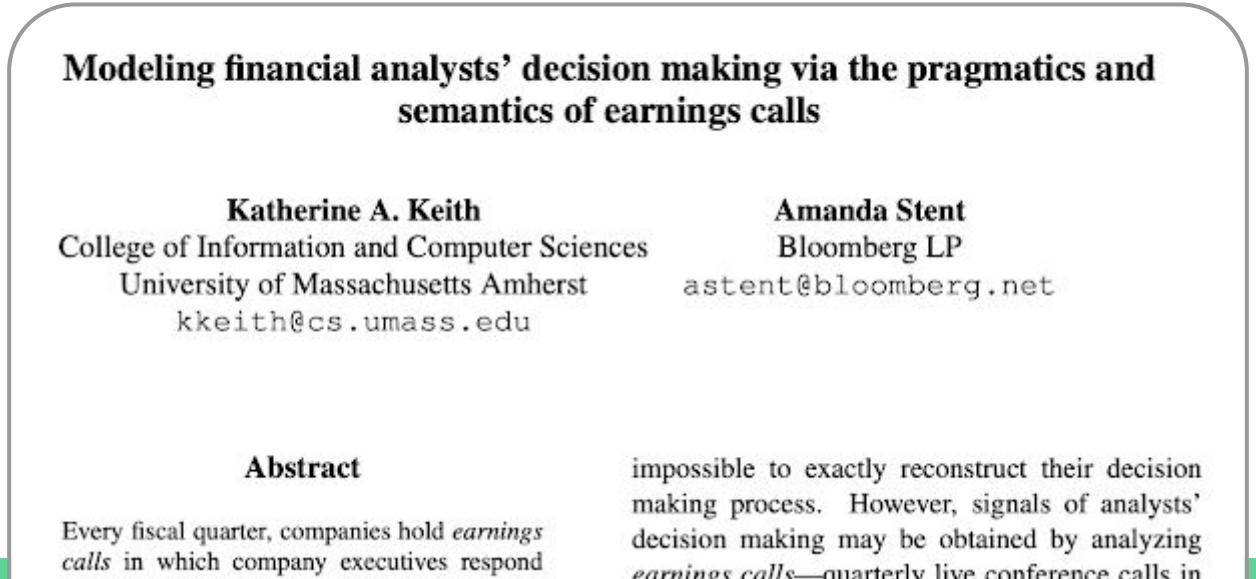
- It is easy to see the variety and the centrality of Machine Learning for a modern understanding of the financial industry:
  - sentiment analysis of finance news
  - stock market prediction
  - document classification





NLP in Finance - A Research Example

- “Modeling financial analysts’ decision making via the pragmatics and semantics of earnings calls”, from ACL 2019.



Modeling financial analysts’ decision making via the pragmatics and semantics of earnings calls

Katherine A. Keith
College of Information and Computer Sciences
University of Massachusetts Amherst
kkeith@cs.umass.edu

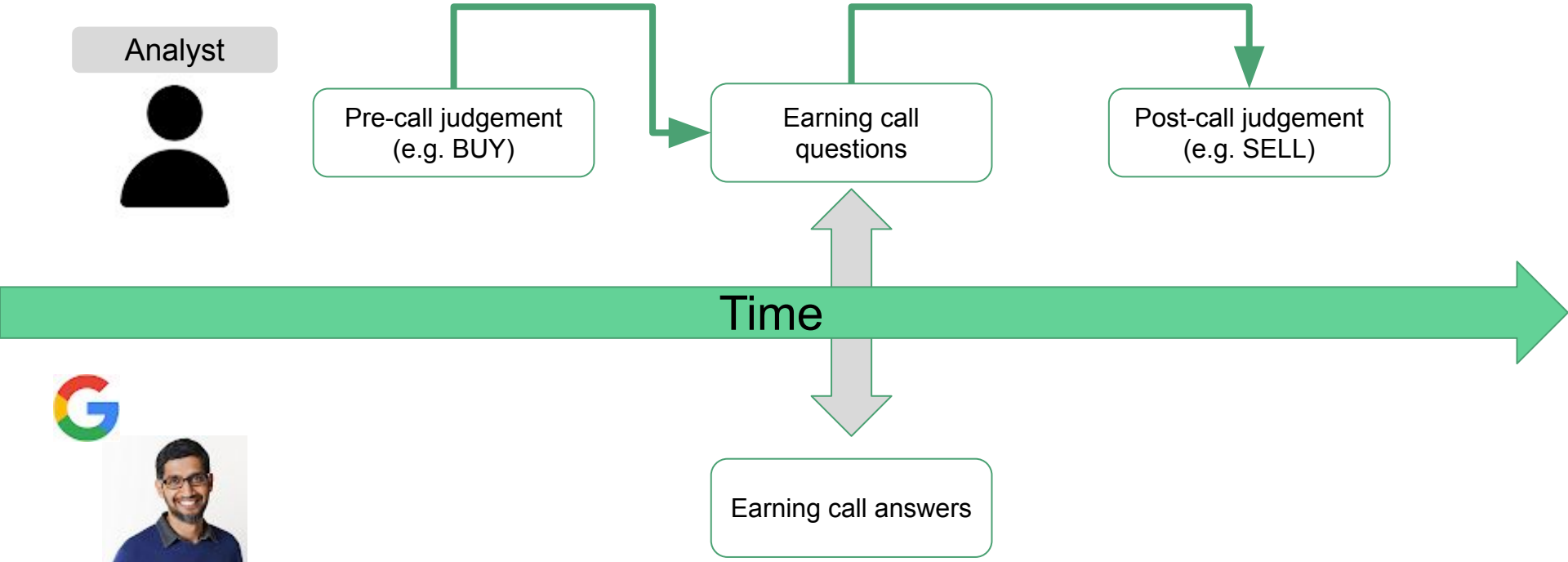
Amanda Stent
Bloomberg LP
astent@bloomberg.net

Abstract

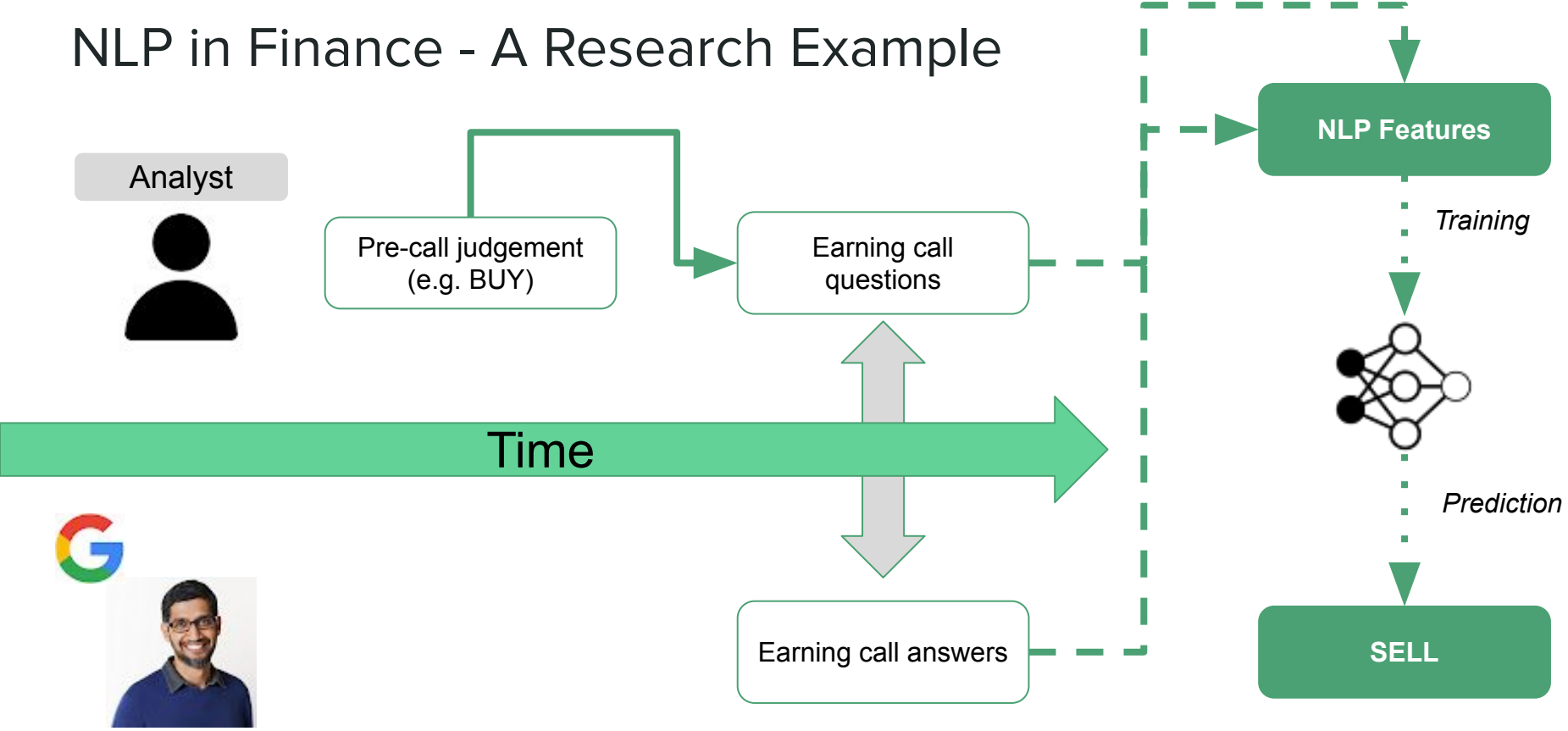
Every fiscal quarter, companies hold *earnings calls* in which company executives respond

impossible to exactly reconstruct their decision making process. However, signals of analysts’ decision making may be obtained by analyzing *earnings calls*—quarterly live conference calls in

NLP in Finance - A Research Example



NLP in Finance - A Research Example



The rise of MLOps

- “If a ~~tree falls in the forest~~ a **model** runs on your laptop with ~~ears to hear~~ **users** does it make ~~a sound~~ an **alpha**?”

The rise of MLOps

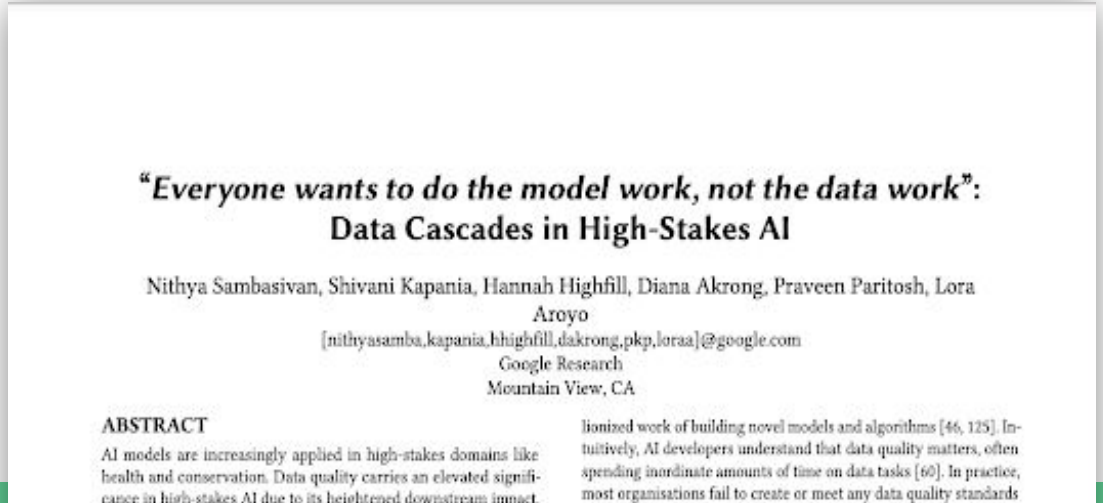
- “If a ~~tree falls in the forest~~ a **model** runs on your laptop with ~~ears to hear~~ **users** does it make ~~a sound~~ an **alpha**?”

“Through 2023, at least 50% of IT leaders will struggle to move their AI predictive projects past proof of concept to a production level of maturity.”
Gartner, 2022

“Gartner research shows only 53% of projects make it from AI prototypes to production. CIOs and IT leaders find it hard to scale AI projects because they lack the tools to create and manage a production-grade AI pipeline.”
Gartner, 2021

The rise of MLOps

- Models are a tiny part of ML platforms, and often the least problematic (with some *caveat*);
- while everybody wants to do the model work, data work is often equally (or more) important in practice.



"*Everyone wants to do the model work, not the data work*": Data Cascades in High-Stakes AI

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, Lora Aroyo

[nithyasamba,kapania,hhighfill,dakrong,pkp,loraa]@google.com

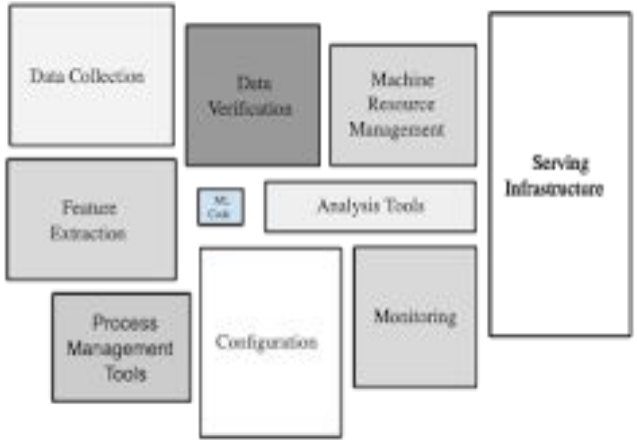
Google Research

Mountain View, CA

ABSTRACT

AI models are increasingly applied in high-stakes domains like health and conservation. Data quality carries an elevated significance in high-stakes AI due to its heightened downstream impact.

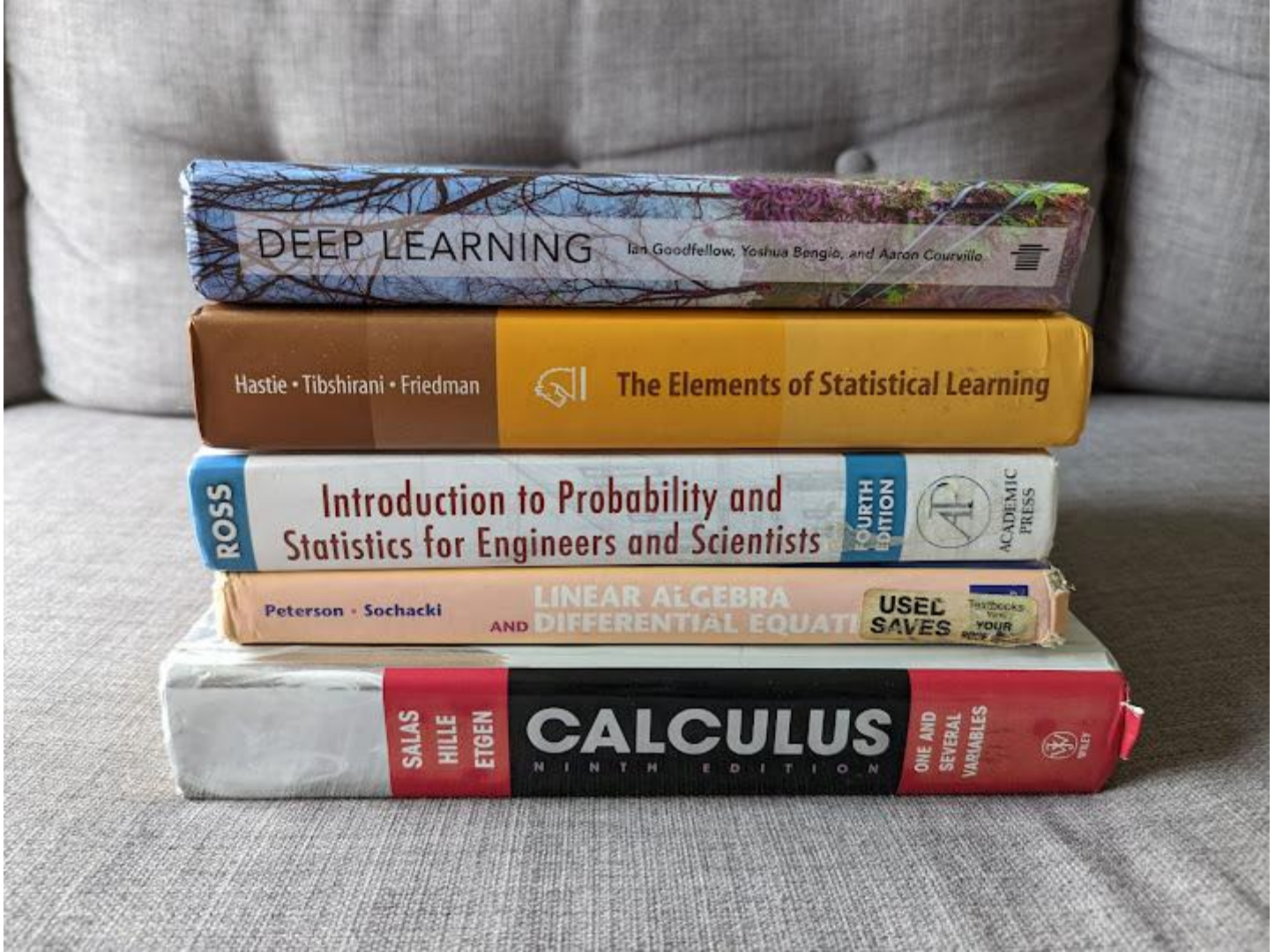
lionized work of building novel models and algorithms [46, 125]. Intuitively, AI developers understand that data quality matters, often spending inordinate amounts of time on data tasks [60]. In practice, most organisations fail to create or meet any data quality standards

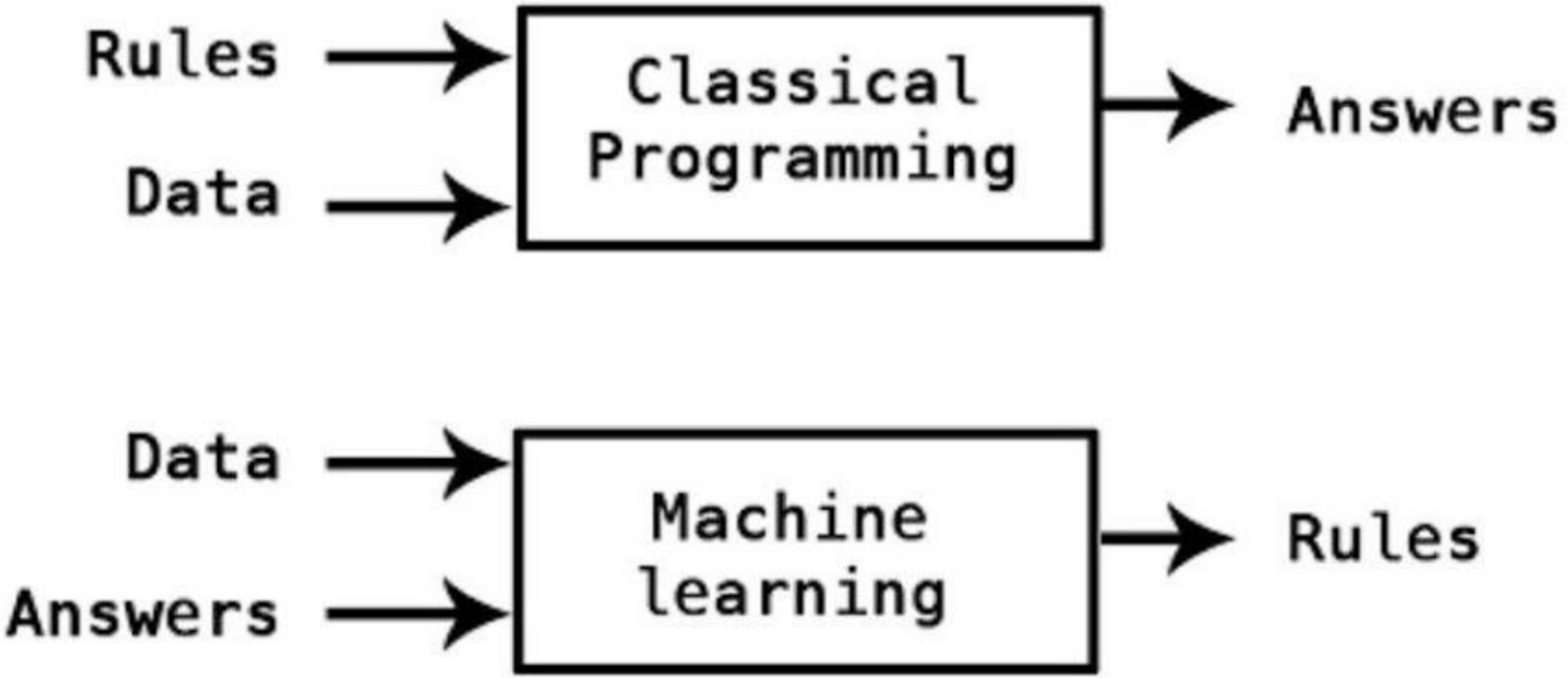


Putting it all together

- Ethan
  - Python / coding fundamentals
  - Introduction to Machine Learning
  - Use cases: fraud detection, time-series
  - Introduction to deep learning
  - **HERE**: you should be able to have a *local*, well structured, ML project that trains a model.
- Jacopo
  - Deep learning for NLP: word embeddings and language modelling
  - Monitoring and testing
  - Use cases: text analysis, classification
  - **HERE**: you should be able to instrument your *local* project to be cloud-ready, and serve predictions from an endpoint.

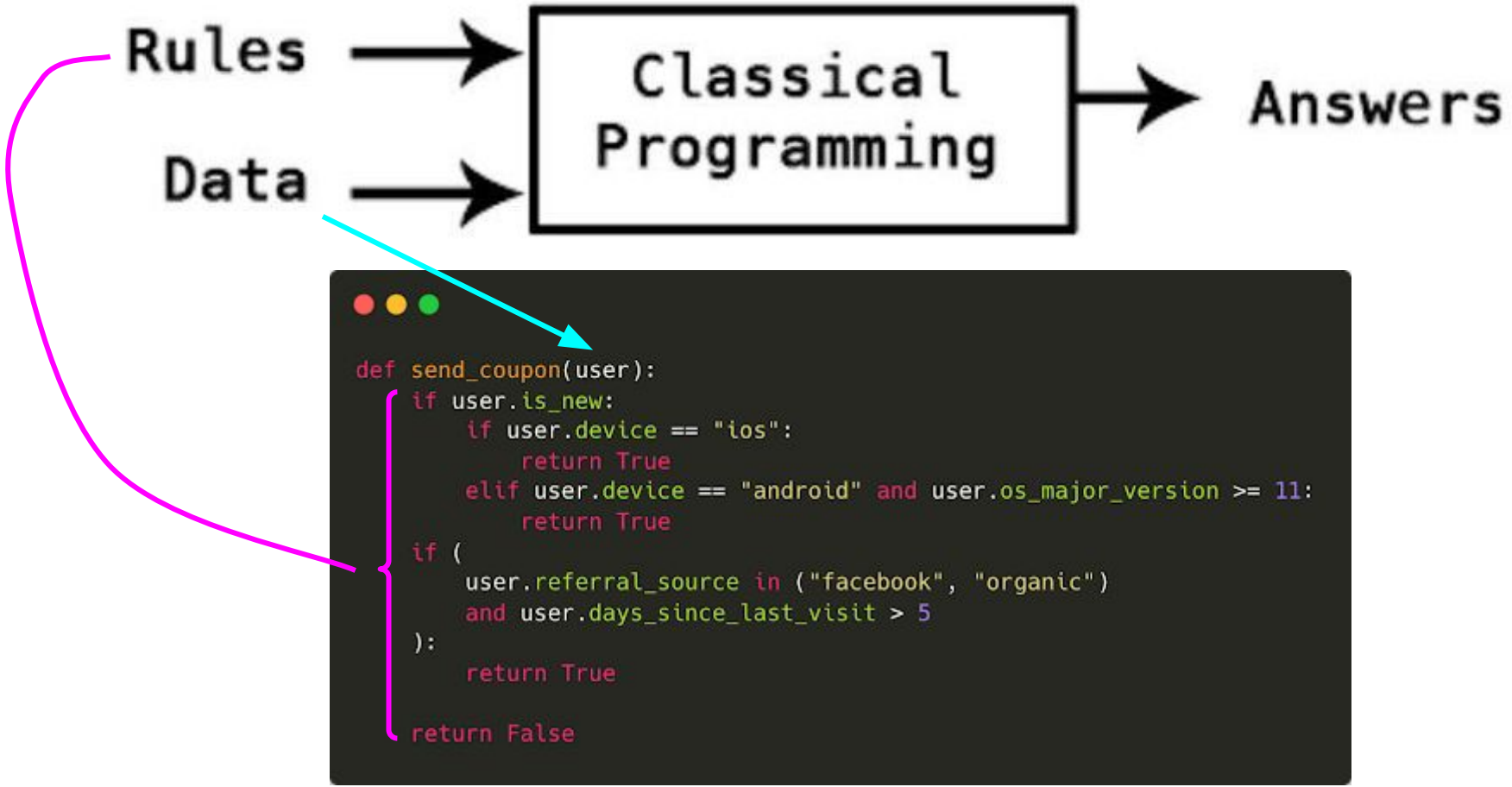
👐 Intro to Machine Learning & Deep Learning 👐

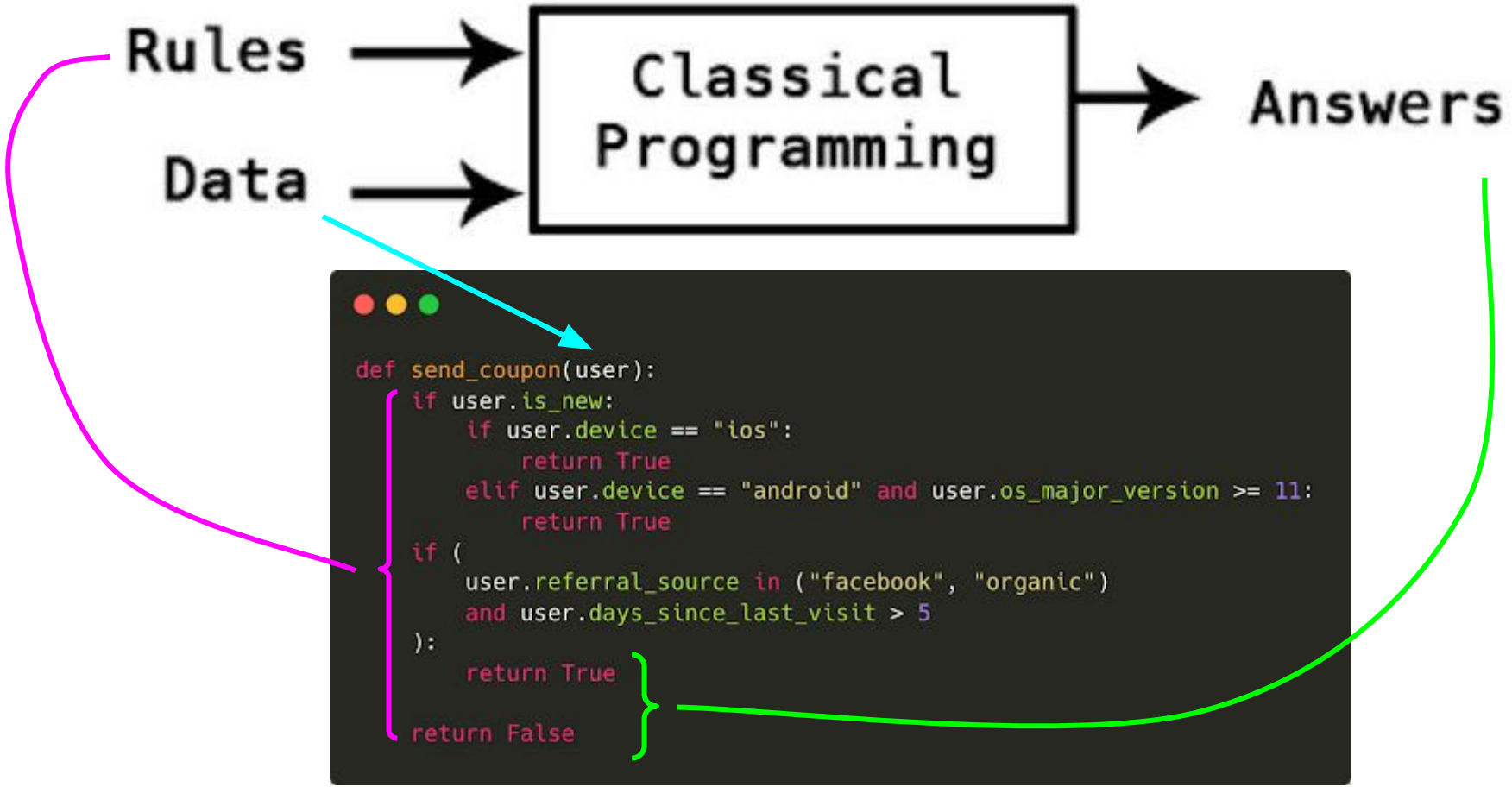

DEEP LEARNING
Ian Goodfellow, Yoshua Bengio, and Aaron Courville
Hastie • Tibshirani • Friedman
The Elements of Statistical Learning
ROSS
Introduction to Probability and Statistics for Engineers and Scientists
FOURTH EDITION
ACADEMIC PRESS
Peterson • Sochacki
AND
USED
SAVES
SALAS
HILLE
ETGEN
CALCULUS
NINTH EDITION
ONE AND SEVERAL VARIABLES
WILEY


Rules
Data
Classical Programming
Answers
Data
Answers
Machine learning
Rules


```python
def send_coupon(user):
    if user.is_new:
        if user.device == "ios":
            return True
        elif user.device == "android" and user.os_major_version >= 11:
            return True
    if (
        user.referral_source in ("facebook", "organic")
        and user.days_since_last_visit > 5
    ):
        return True

    return False
```

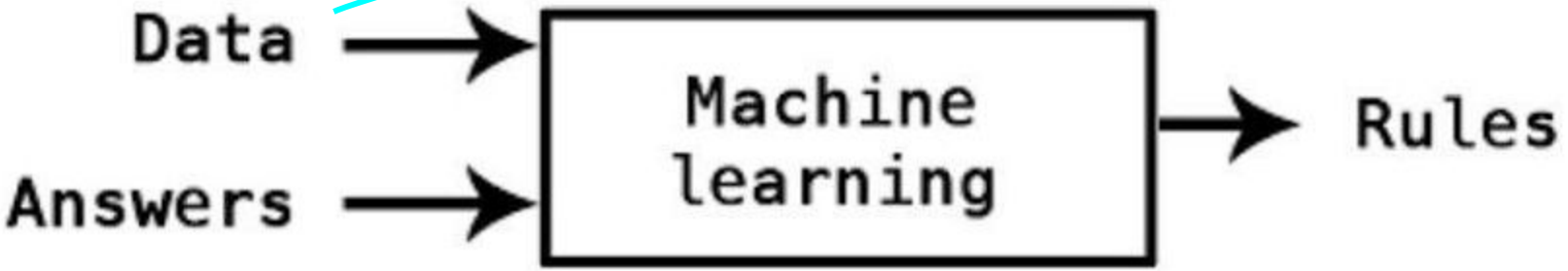

```python
def send_coupon(user):
    if user.is_new:
        if user.device == "ios":
            return True
        elif user.device == "android" and user.os_major_version >= 11:
            return True
    if (
        user.referral_source in ("facebook", "organic")
        and user.days_since_last_visit > 5
    ):
        return True

    return False
```


Rules
Data
Classical Programming
Answers
def send_coupon(user):
if user.is_new:
if user.device == "ios":
return True
elif user.device == "android" and user.os_major_version >= 11:
return True
if (
user.referral_source in ("facebook", "organic")
and user.days_since_last_visit > 5
):
return True
return False

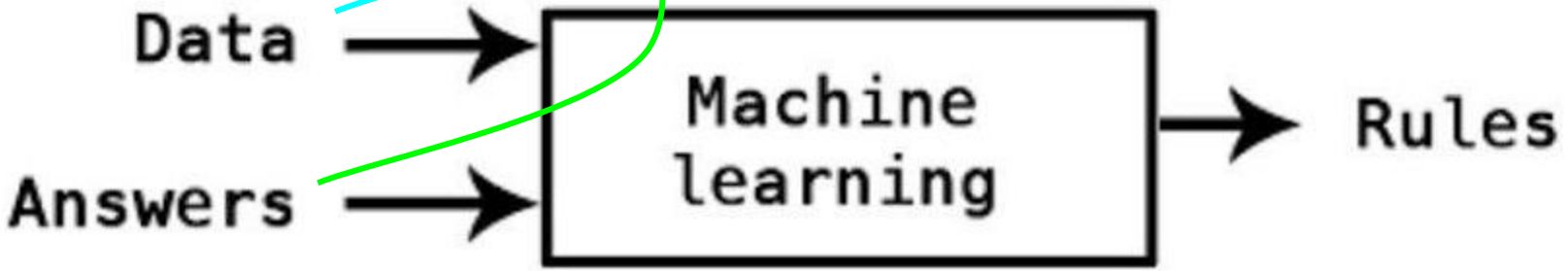

Rules
Data
Classical Programming
Answers
def send_coupon(user):
if user.is_new:
if user.device == "ios":
return True
elif user.device == "android" and user.os_major_version >= 11:
return True
if (
user.referral_source in ("facebook", "organic")
and user.days_since_last_visit > 5
):
return True
return False

```
users = [
    User(is_new=False, device="ios", os_major_version=13, referral_source="facebook", days_since_last_visit=10),
    User(is_new=True, device="android", os_major_version=11, referral_source="search", days_since_last_visit=None),
    User(is_new=False, device="android", os_major_version=10, referral_source="organic", days_since_last_visit=6),
    ...
]
conversions = [
    True,
    False,
    False,
    ...
]

model = train(users, conversions)

def send_coupon(user, model):
    return model.predict(user)
```

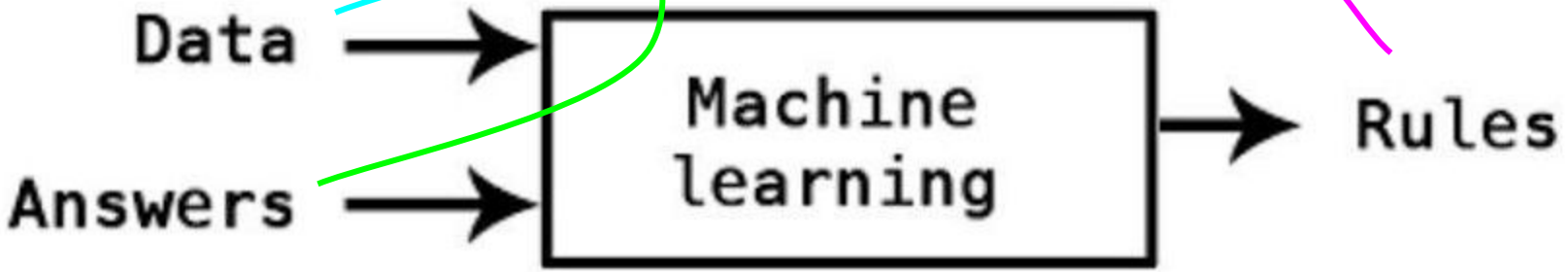
```
users = [
    User(is_new=False, device="ios", os_major_version=13, referral_source="facebook", days_since_last_visit=10),
    User(is_new=True, device="android", os_major_version=11, referral_source="search", days_since_last_visit=None),
    User(is_new=False, device="android", os_major_version=10, referral_source="organic", days_since_last_visit=6),
    ...
]
conversions = [
    True,
    False,
    False,
    ...
]

model = train(users, conversions)

def send_coupon(user, model):
    return model.predict(user)
```



```
users = [
    User(is_new=False, device="ios", os_major_version=13, referral_source="facebook", days_since_last_visit=10),
    User(is_new=True, device="android", os_major_version=11, referral_source="search", days_since_last_visit=None),
    User(is_new=False, device="android", os_major_version=10, referral_source="organic", days_since_last_visit=6),
    ...
]
conversions = [
    True,
    False,
    False,
    ...
]

model = train(users, conversions)

def send_coupon(user, model):
    return model.predict(user)
```

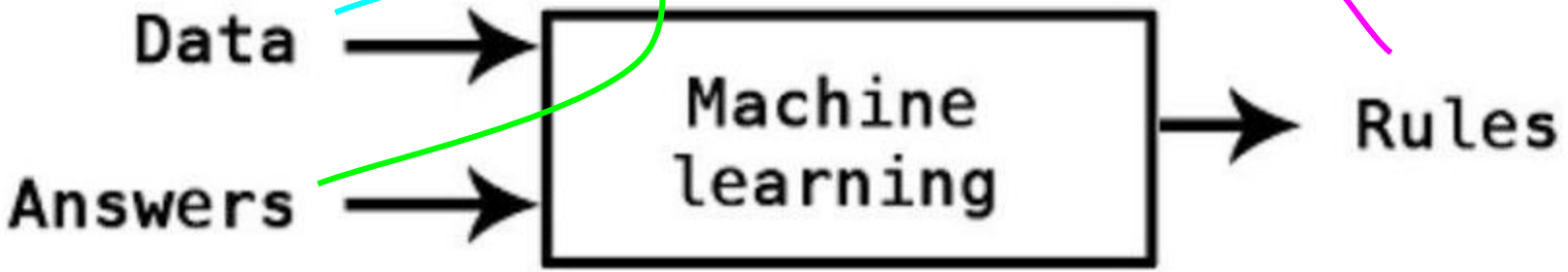


```
users = [
    User(is_new=False, device="ios", os_major_version=13, referral_source="facebook", days_since_last_visit=10),
    User(is_new=True, device="android", os_major_version=11, referral_source="search", days_since_last_visit=None),
    User(is_new=False, device="android", os_major_version=10, referral_source="organic", days_since_last_visit=6),
    ...
]
conversions = [
    True,
    False,
    False,
    ...
]

model = train(users, conversions)

def send_coupon(user, model):
    return model.predict(user)
```



```
users = [
    User(is_new=False, device="ios", os_major_version=13, referral_sour                last_visit=10),
    User(is_new=True, device="android", os_major_version=11, ref                e_last_visit=None),
    User(is_new=False, device="android", os_major_versio                ce_last_visit=6),
    ...
]
conversions = [
    True,
    False,
    False,
    ...
]

model = train(users,

def send_coupon(user,
    return model.predic
```

How do you learn the rules?

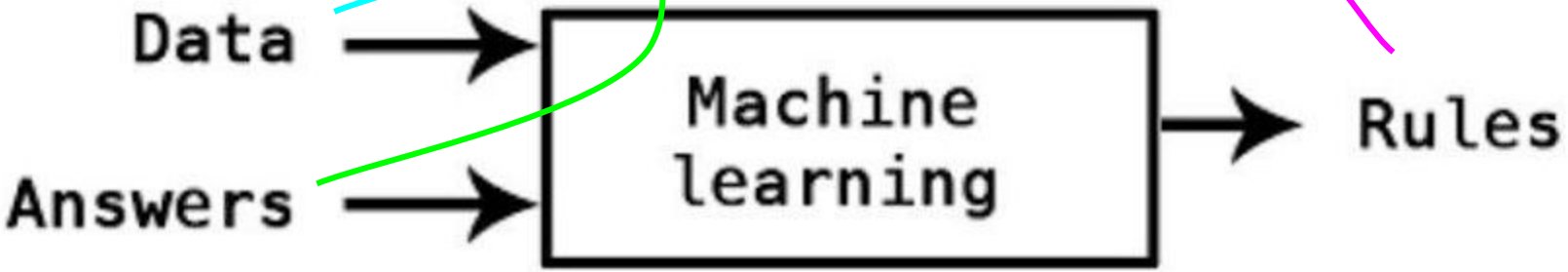


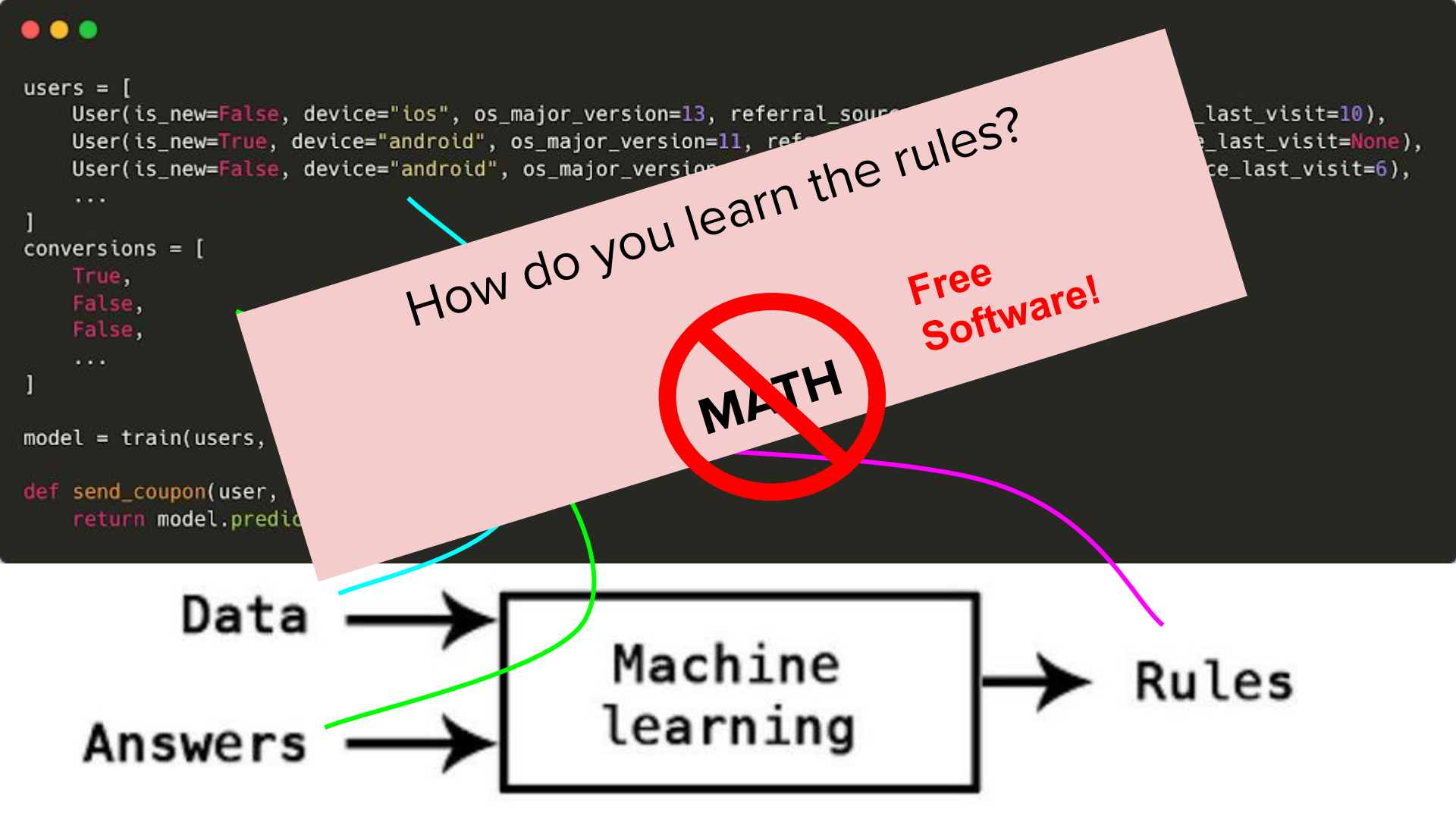
```
users = [
    User(is_new=False, device="ios", os_major_version=13, referral_sour
    User(is_new=True, device="android", os_major_version=11, ref
    User(is_new=False, device="android", os_major_versio
    ...
]
conversions = [
    True,
    False,
    False,
    ...
]

model = train(users,

def send_coupon(user,
    return model.predic
```

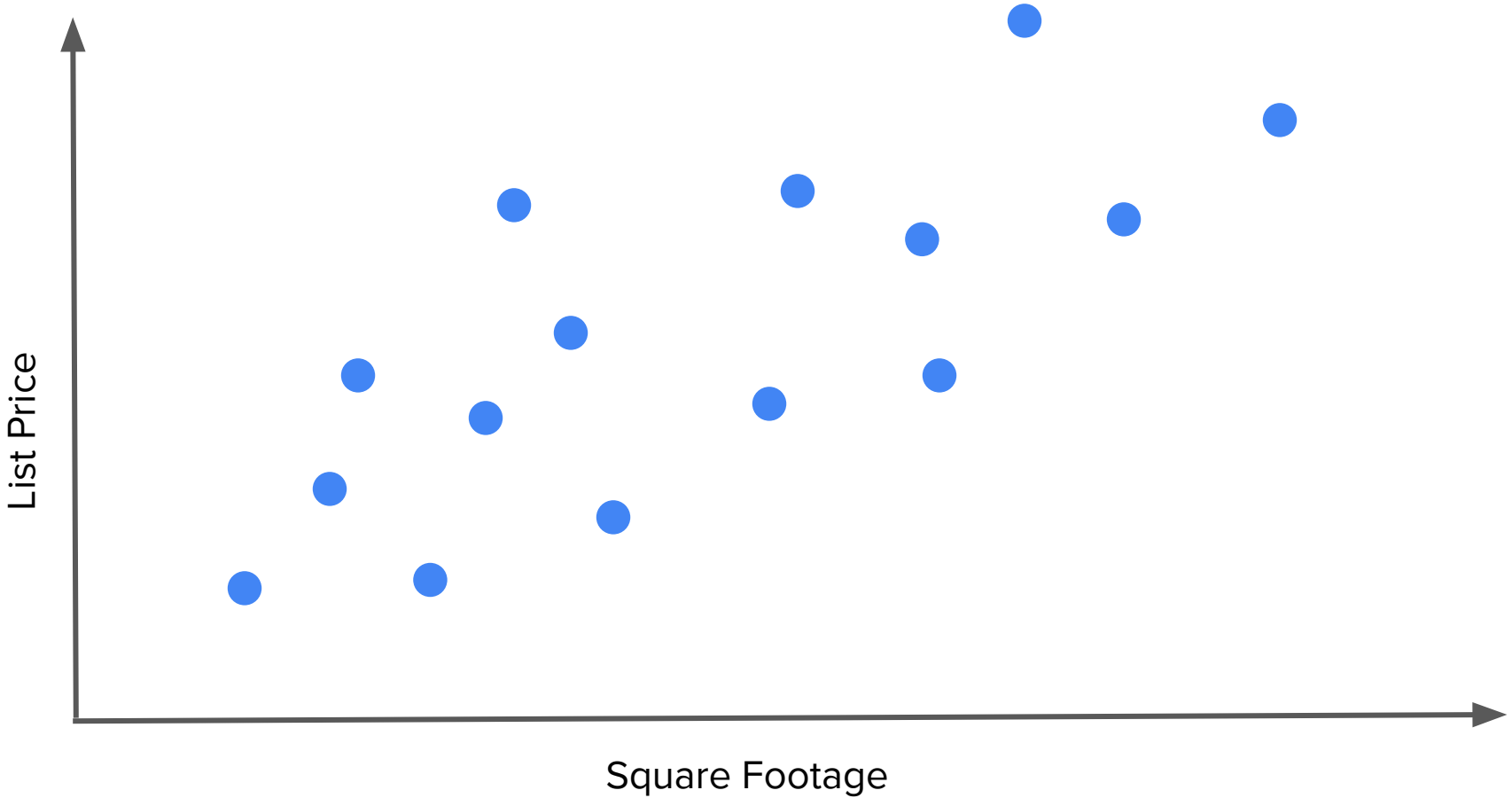
_last_visit=10),
e_last_visit=None),
ce_last_visit=6),

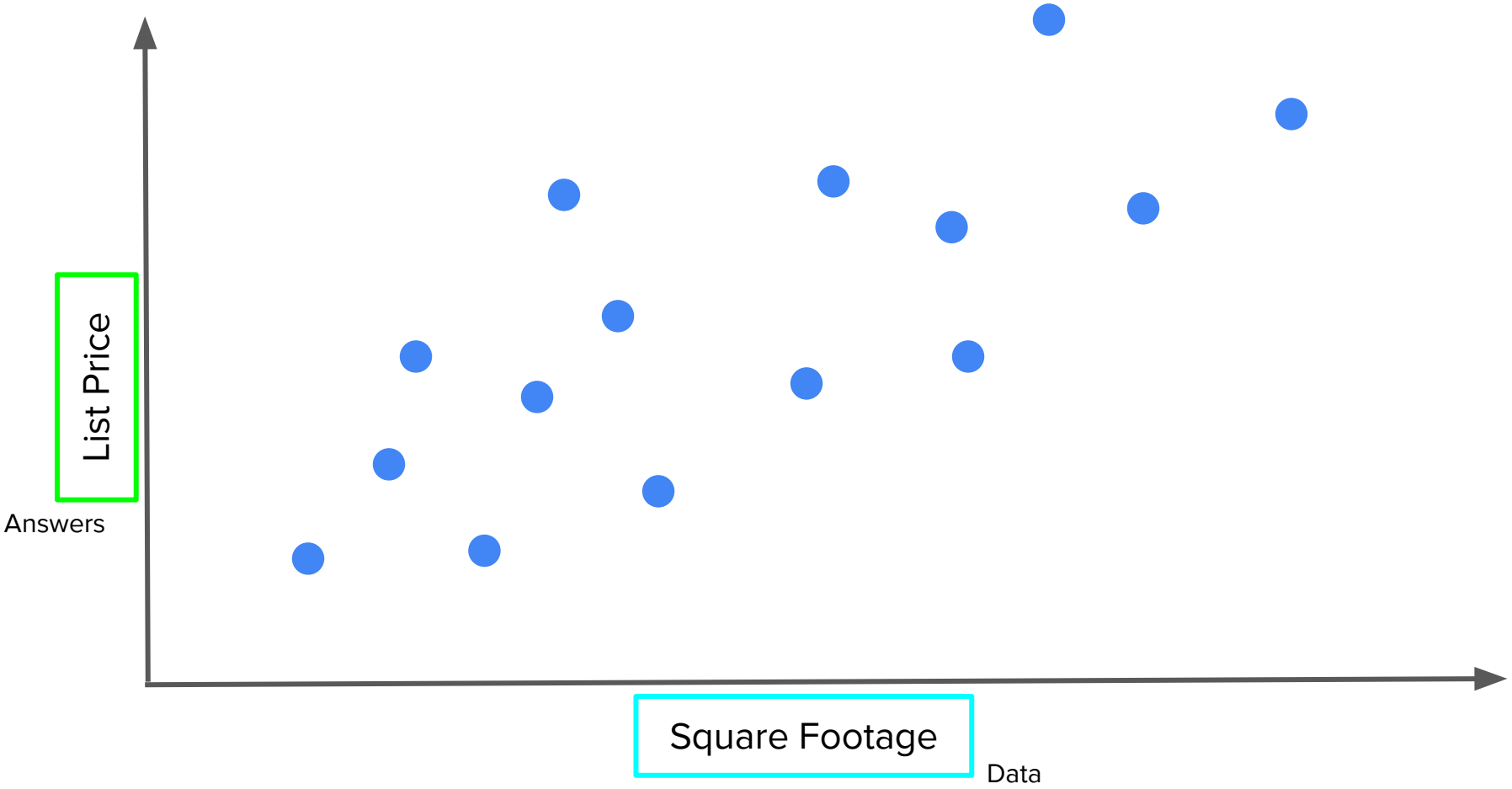
How do you learn the rules?

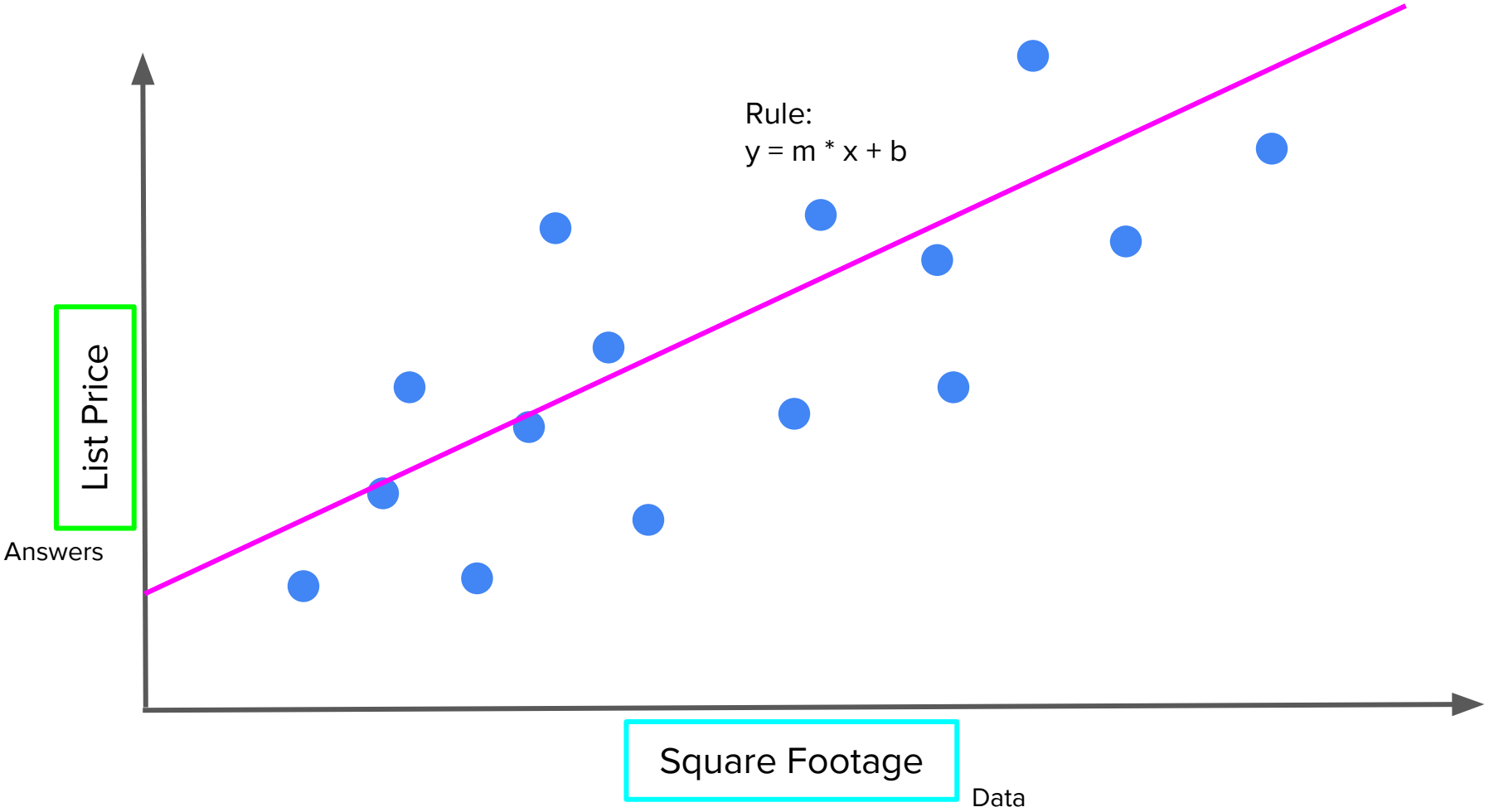
MATH

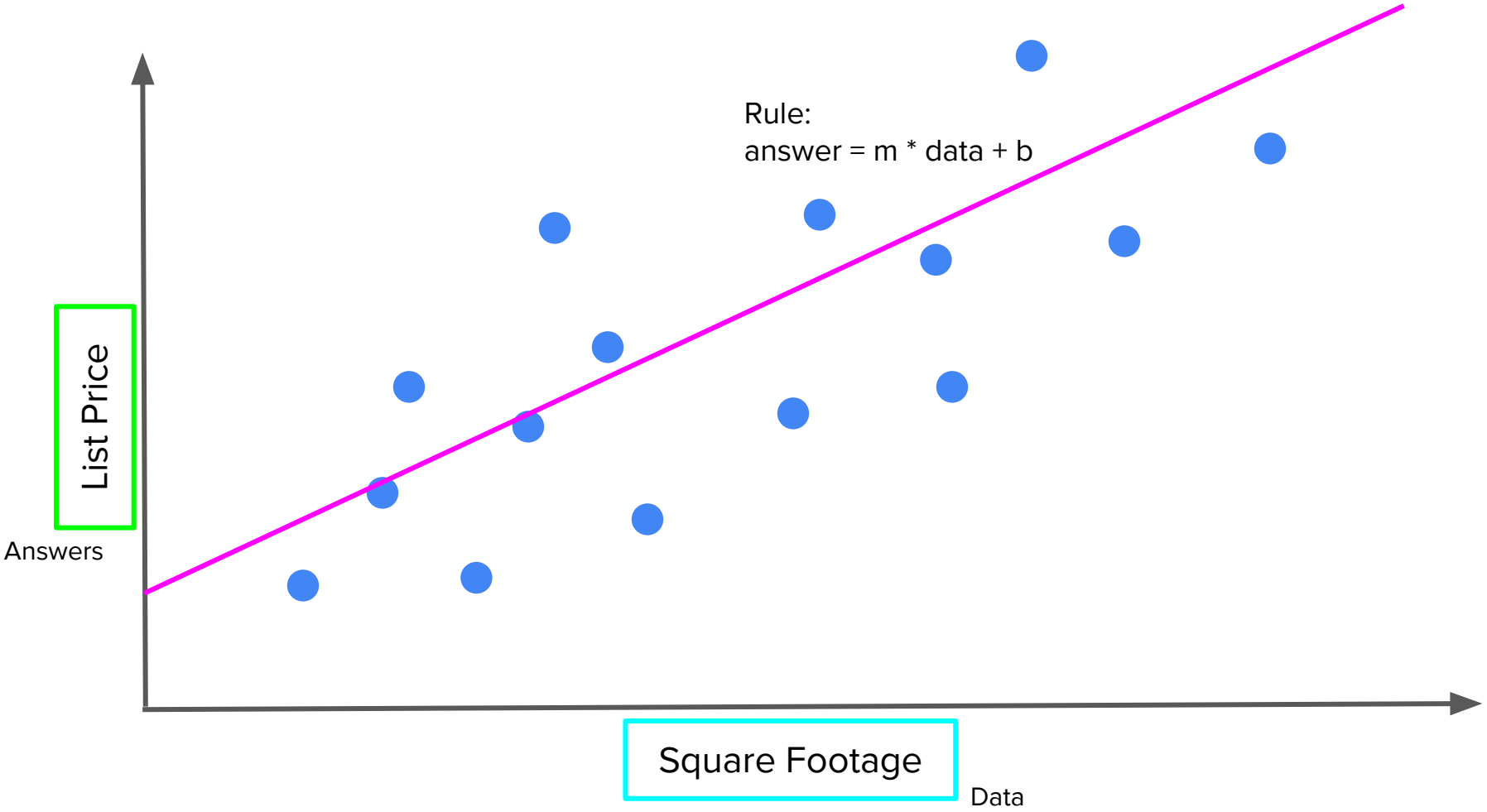



users = [
User(is_new=False, device="ios", os_major_version=13, referral_sour
_last_visit=10),
User(is_new=True, device="android", os_major_version=11, ref
e_last_visit=None),
User(is_new=False, device="android", os_major_versio
ce_last_visit=6),
...
]
conversions = [
True,
False,
False,
...
]
model = train(users,
def send_coupon(user,
return model.predic
How do you learn the rules?
MATH
Free Software!
Data
Answers
Machine learning
Rules

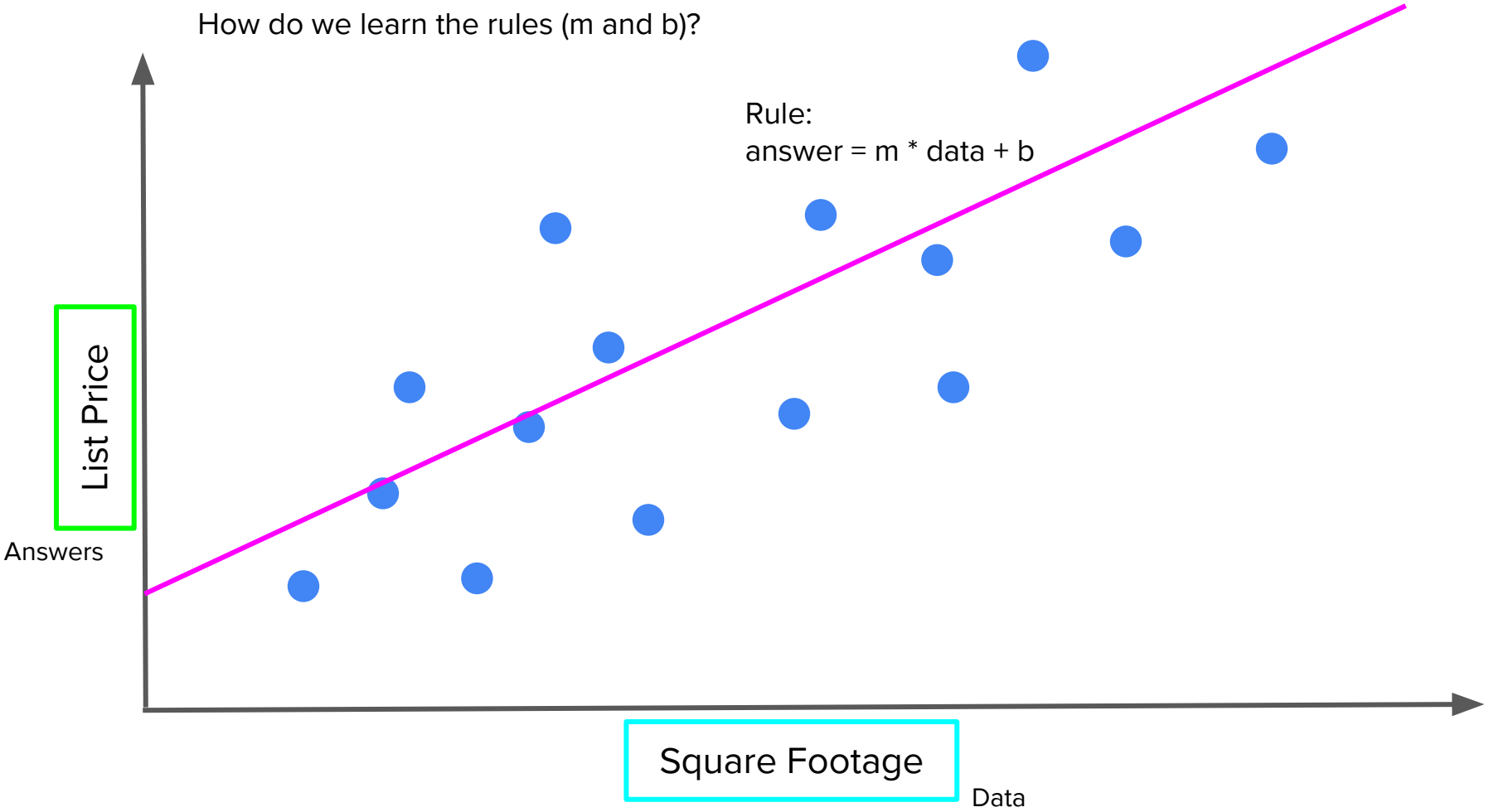

List Price
Square Footage


List Price
Answers
Square Footage
Data


Rule:
y = m * x + b
List Price
Answers
Square Footage
Data


Rule:
answer = m * data + b
List Price
Answers
Square Footage
Data

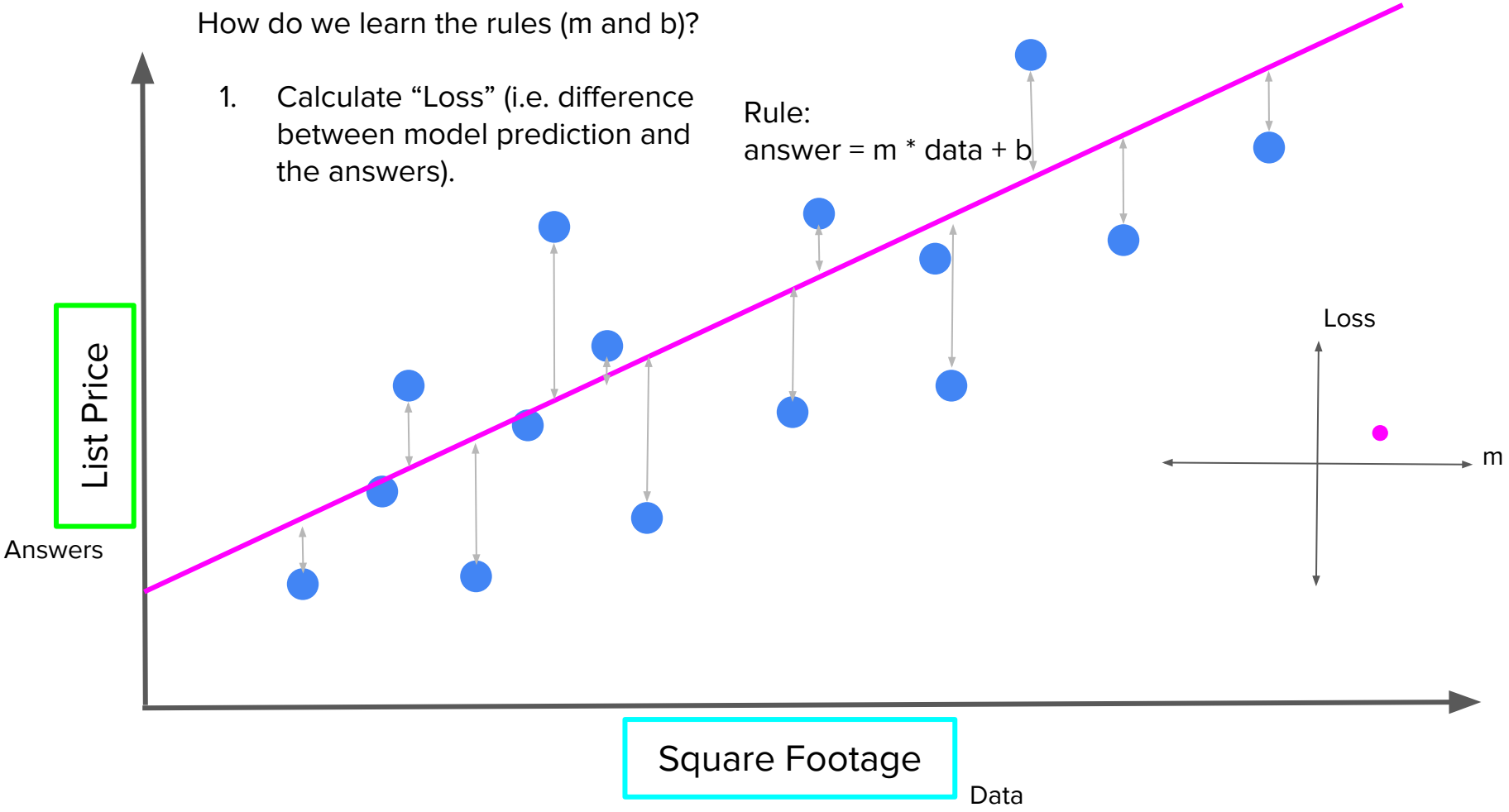
How do we learn the rules (m and b)?



How do we learn the rules (m and b)?

1. Calculate “Loss” (i.e. difference between model prediction and the answers).

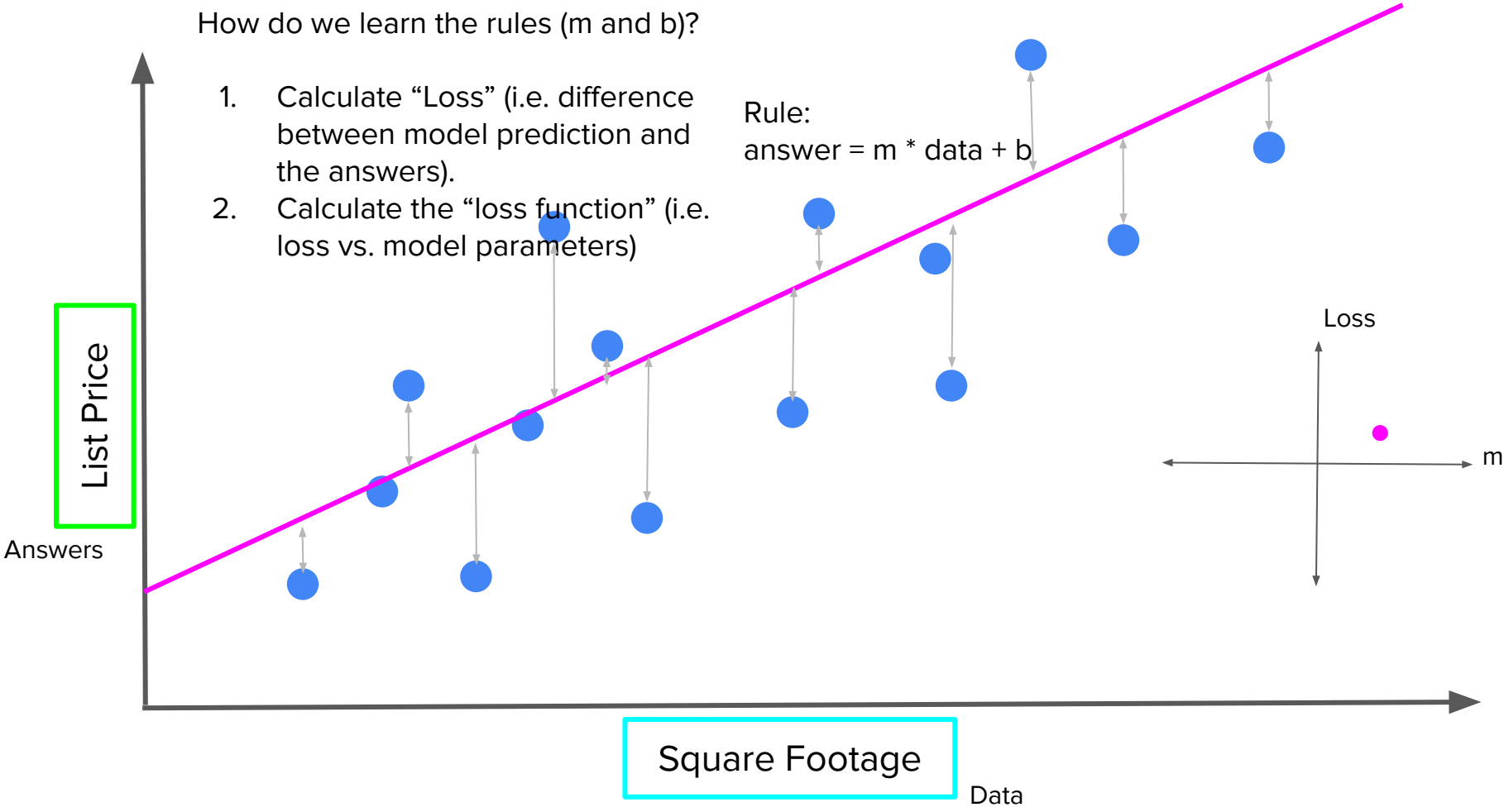
Rule:
answer = m * data + b



How do we learn the rules (m and b)?

1. Calculate “Loss” (i.e. difference between model prediction and the answers).
2. Calculate the “loss function” (i.e. loss vs. model parameters)

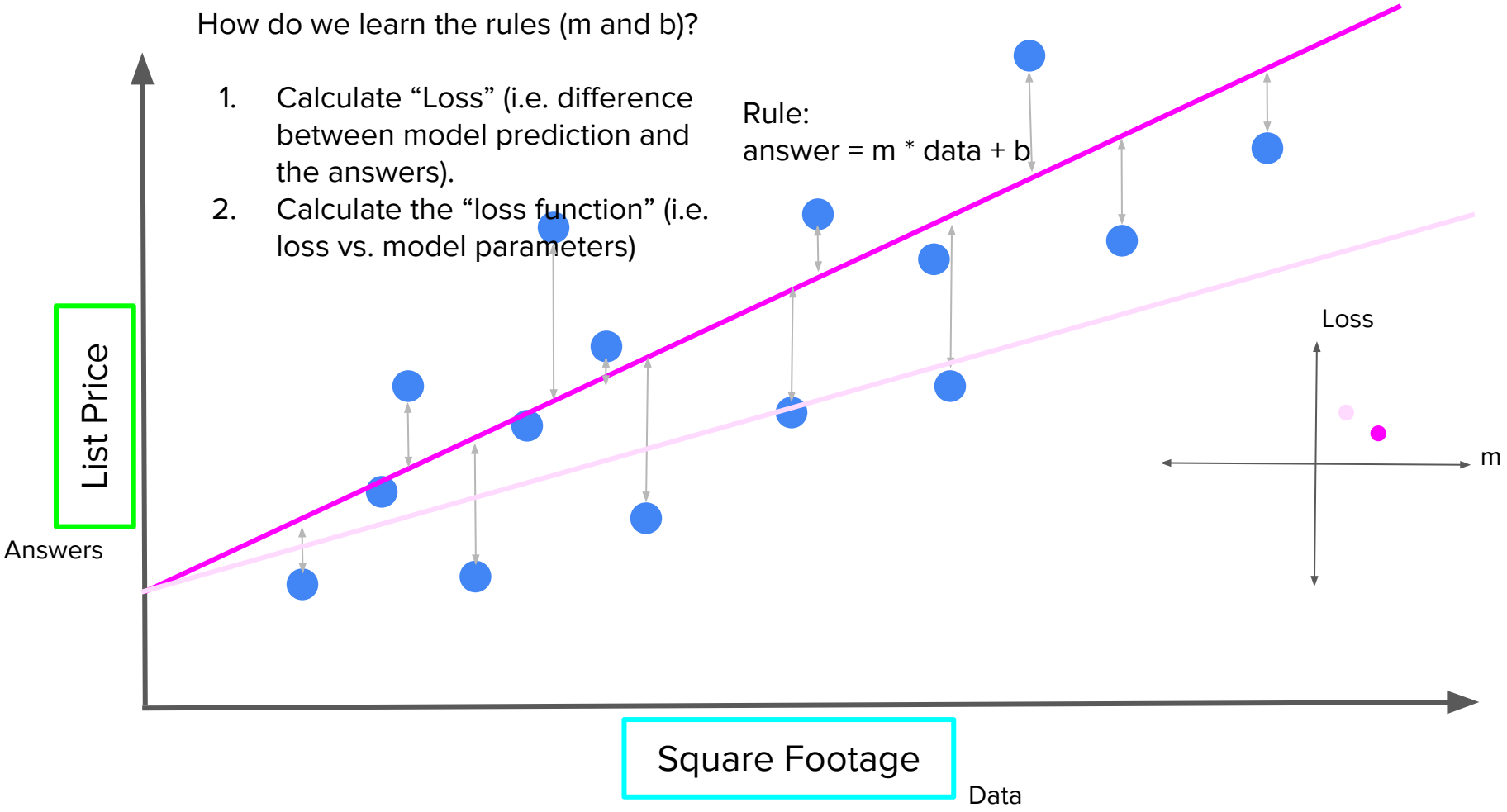
Rule:
answer = m * data + b



How do we learn the rules (m and b)?

1. Calculate “Loss” (i.e. difference between model prediction and the answers).
2. Calculate the “loss function” (i.e. loss vs. model parameters)

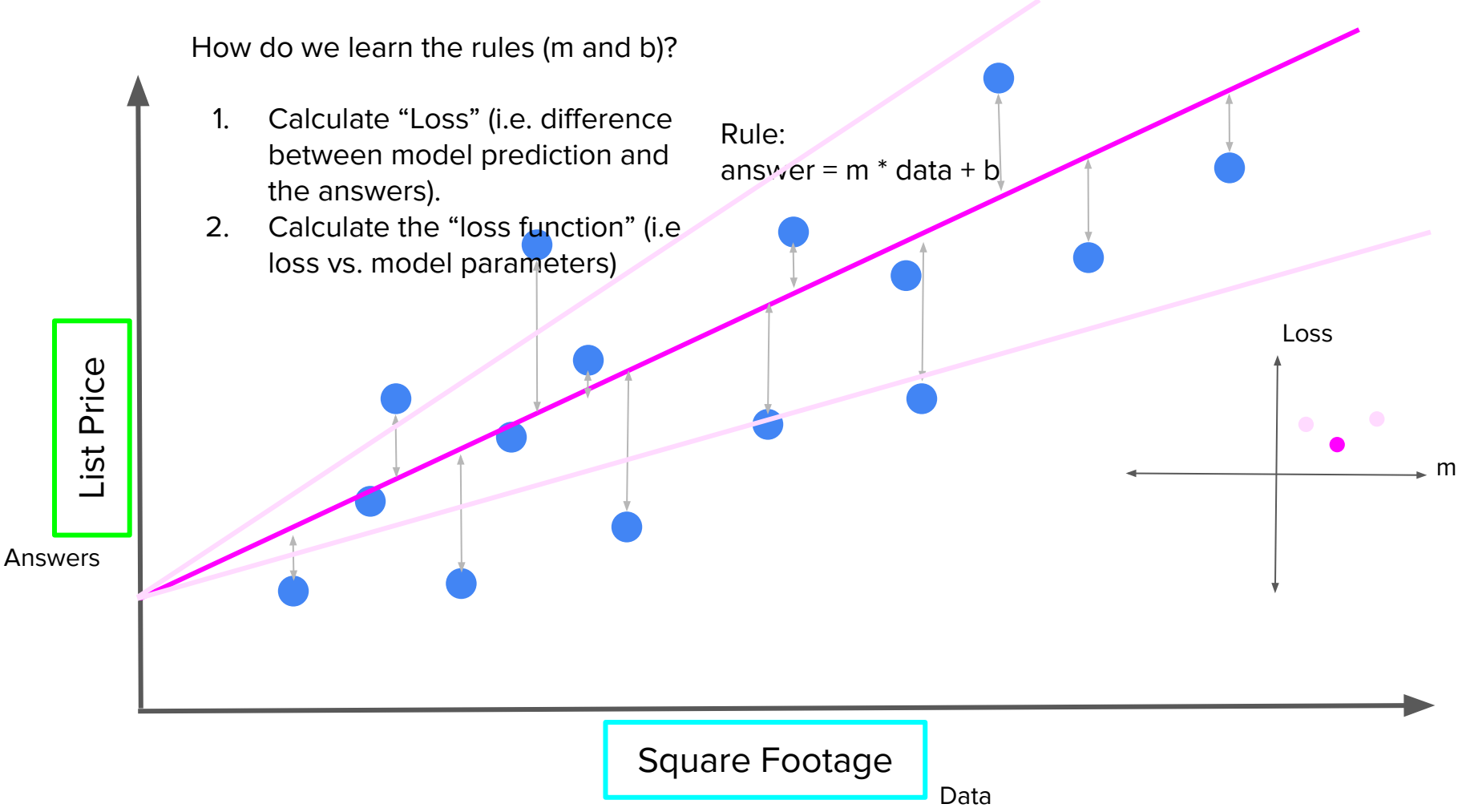
Rule:
answer = m * data + b



How do we learn the rules (m and b)?

1. Calculate “Loss” (i.e. difference between model prediction and the answers).
2. Calculate the “loss function” (i.e loss vs. model parameters)

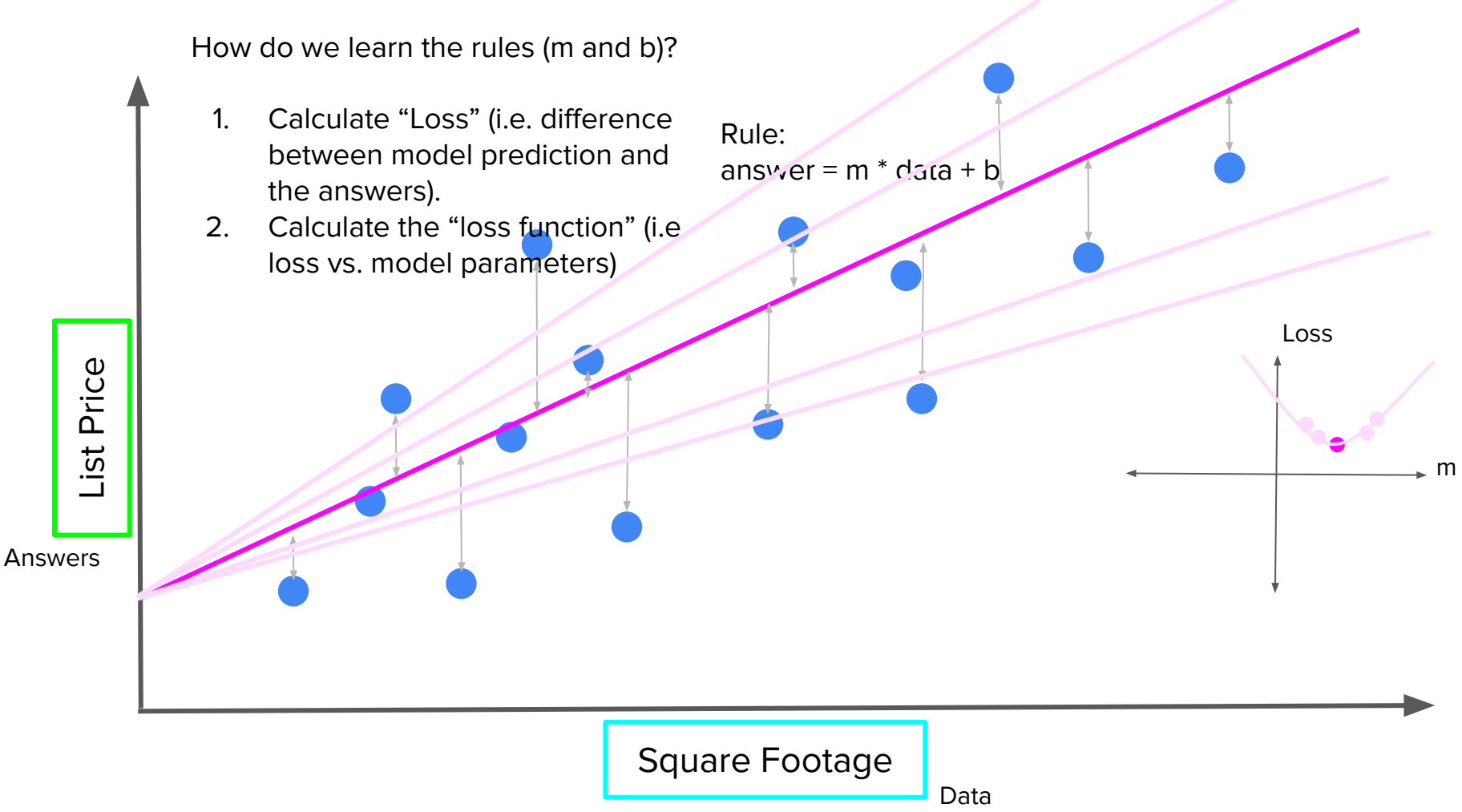
Rule:
answer = m * data + b



How do we learn the rules (m and b)?

1. Calculate “Loss” (i.e. difference between model prediction and the answers).
2. Calculate the “loss function” (i.e loss vs. model parameters)

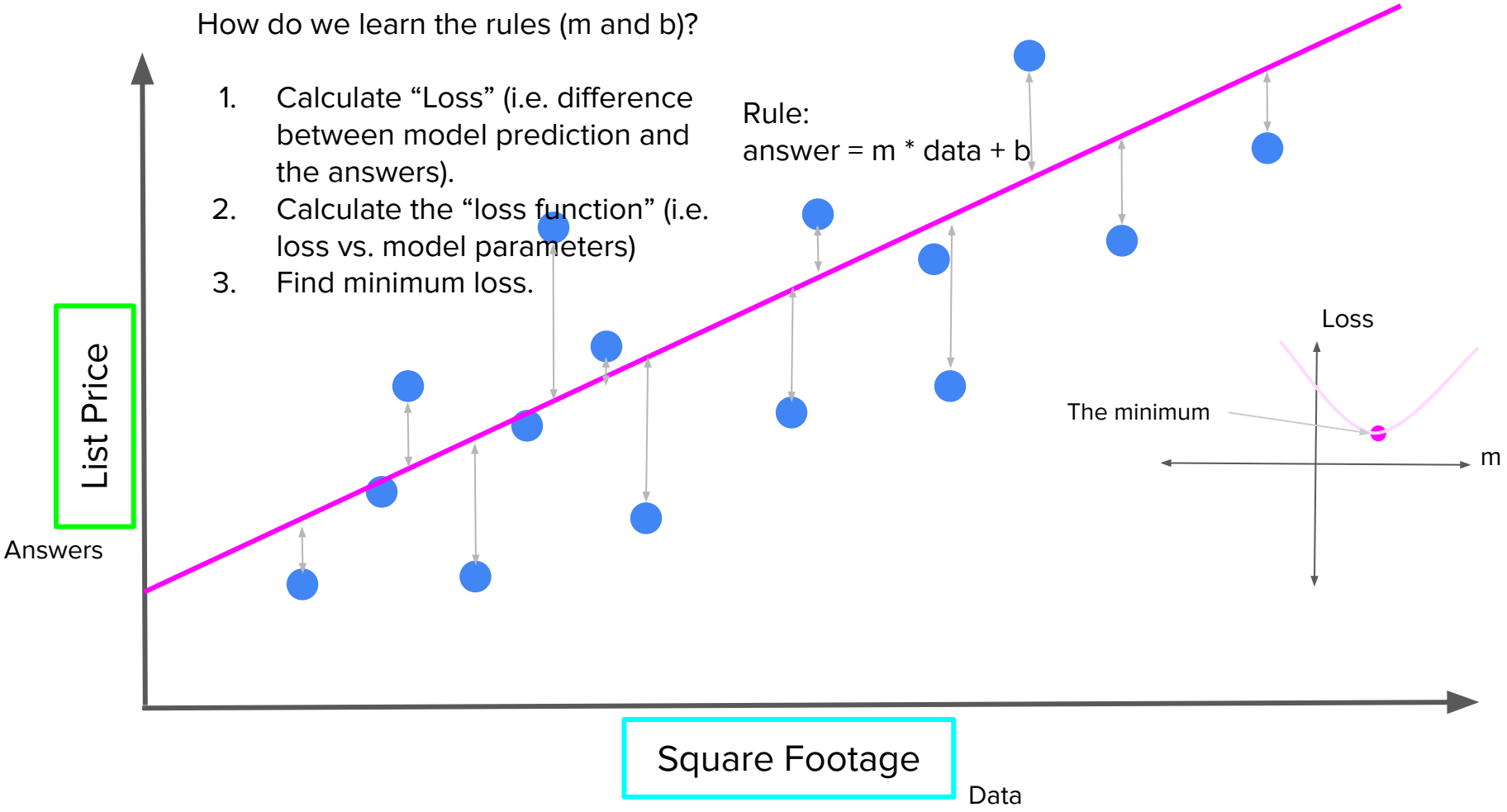
Rule:
answer = m * data + b



How do we learn the rules (m and b)?

1. Calculate “Loss” (i.e. difference between model prediction and the answers).
2. Calculate the “loss function” (i.e. loss vs. model parameters)
3. Find minimum loss.

Rule:
answer = m * data + b



The ML Recipe

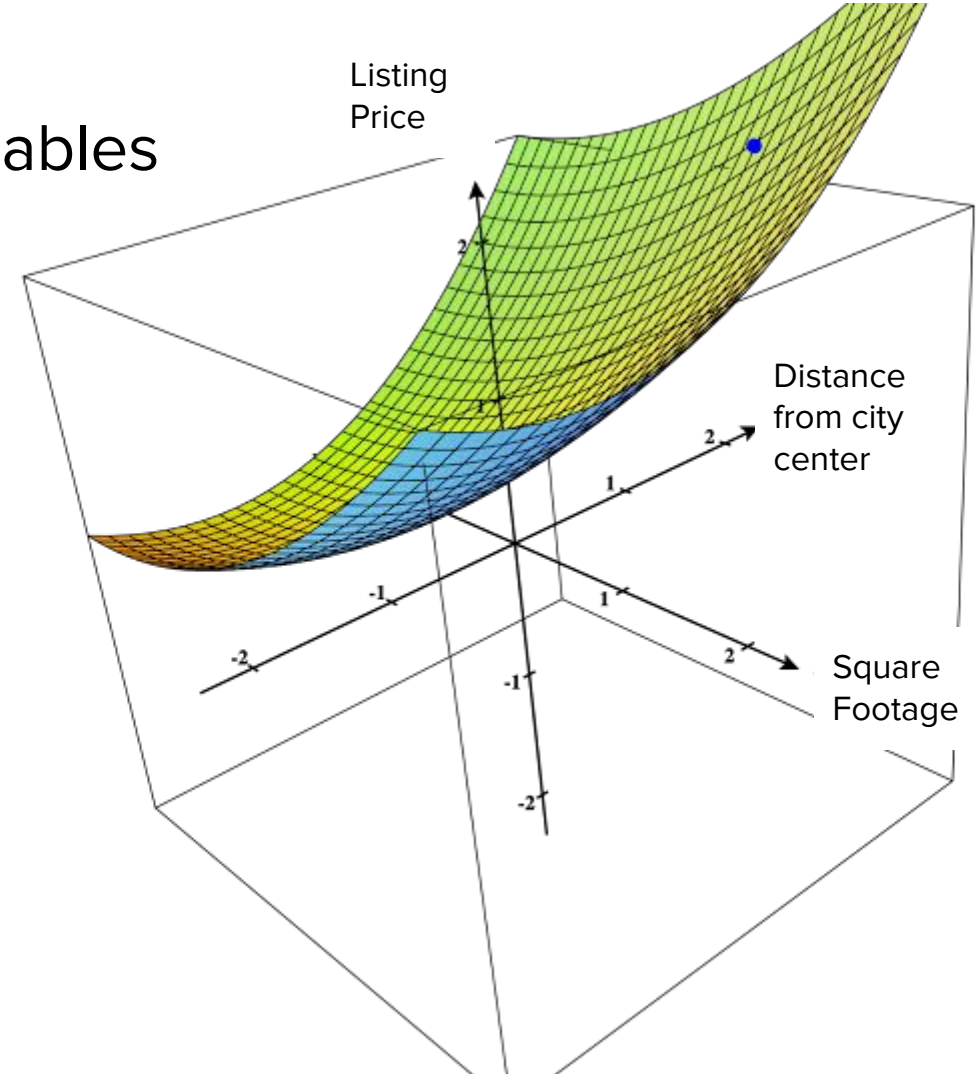
1. Think up some model
2. Feed **data** into the model and make predictions.
3. Calculate the loss between predictions and true values.
4. Determine the model parameters that produce the minimum loss.

The ML Recipe

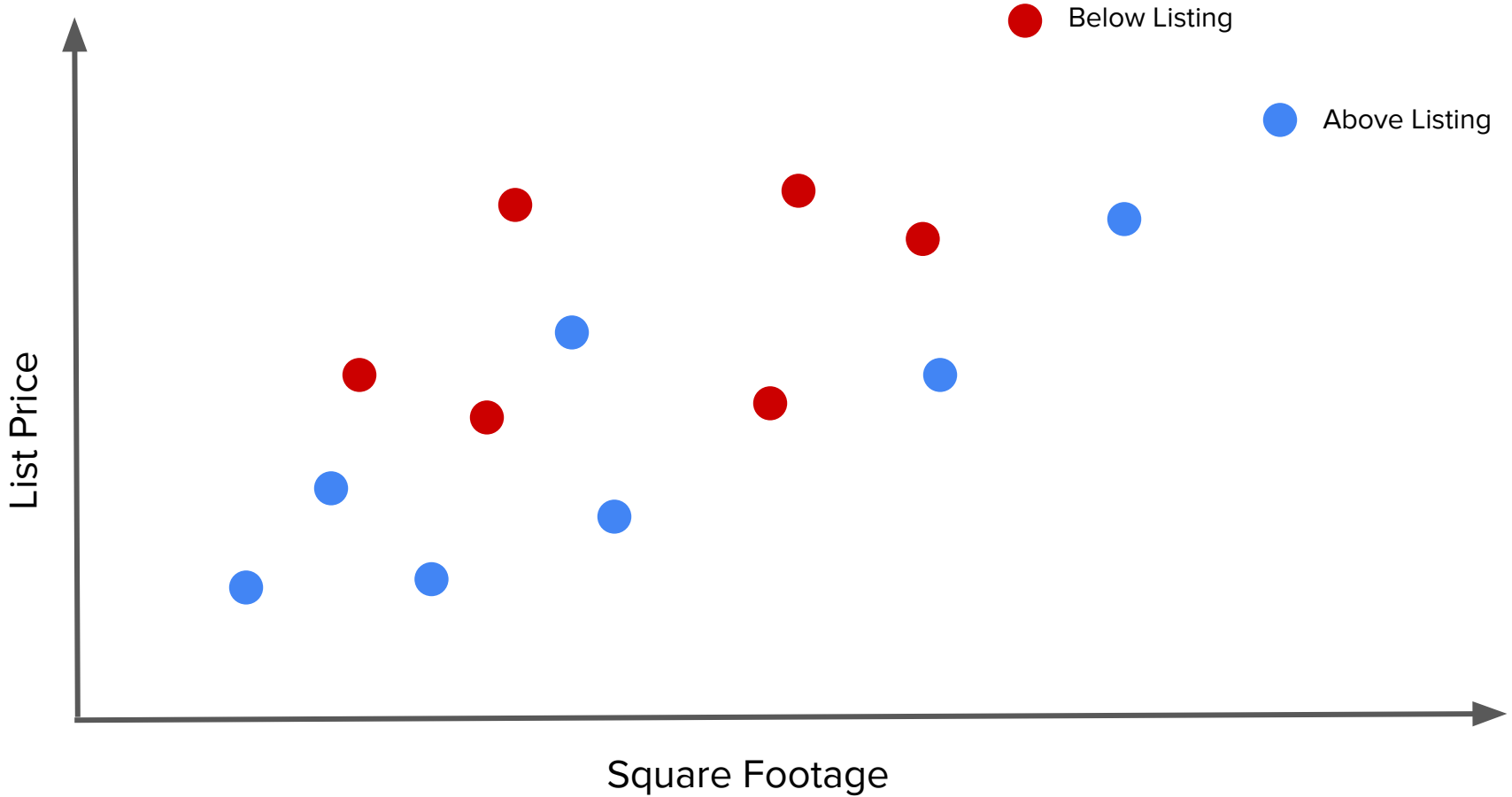
1. Think up some model
2. Feed **data** into the model and make predictions.
3. Calculate the loss between predictions and true values.
4. Determine the model parameters that produce the minimum loss.
    a. Take the derivative of your loss function with respect to the model parameters.
    b. Set it equal to zero.
    c. Solve for the model parameters.

Multiple Variables

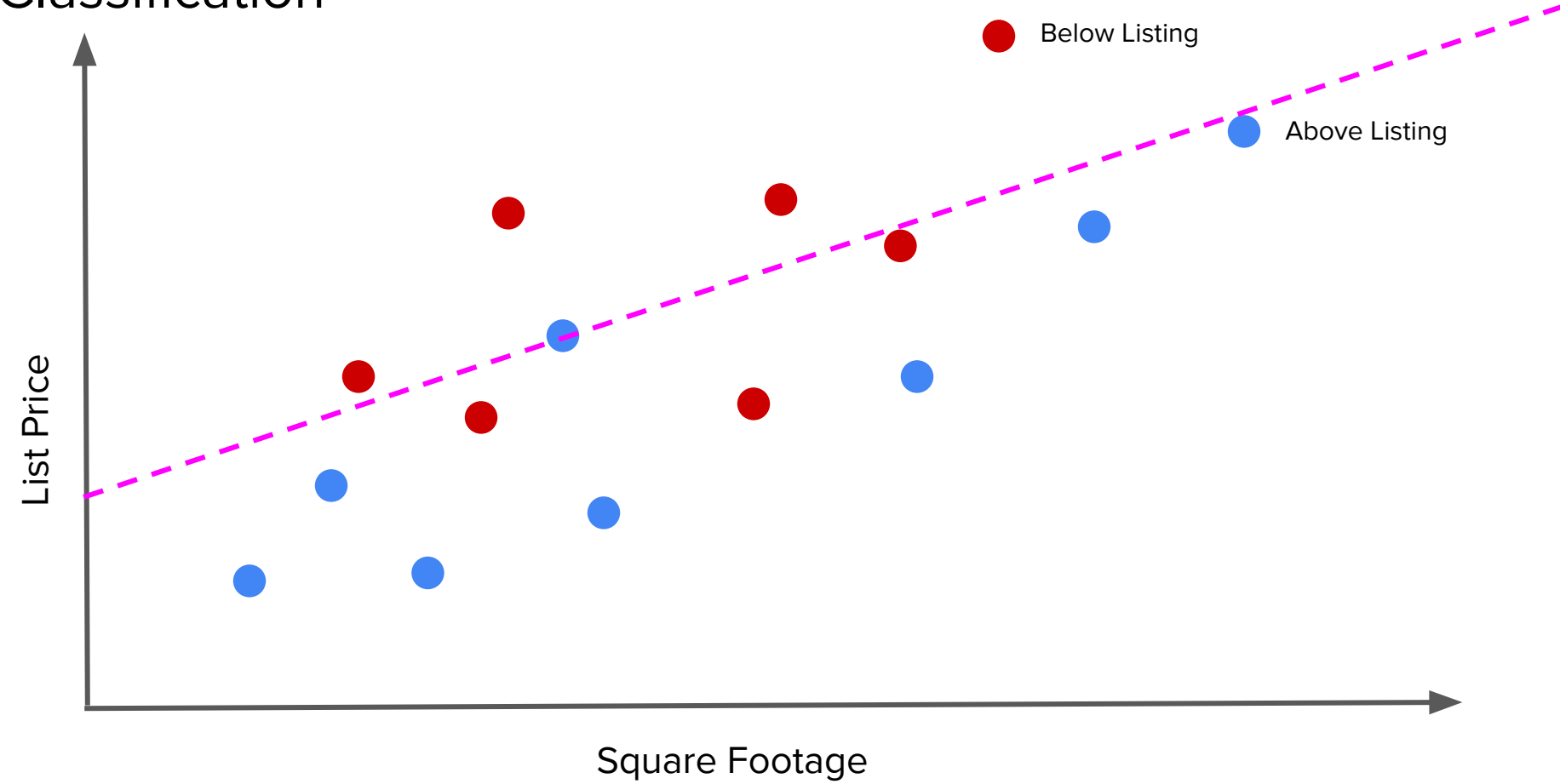
$y = m_1 * x_1 + m_2 * x_2 + b$



Classification



Classification



The Classification Recipe

1. Think up some model
2. Feed **data** into the model and make predictions.

3. Calculate the loss between predictions and true values.

4. Determine the model parameters that produce the minimum loss.
    a. Take the derivative of your loss function with respect to the model parameters.
    b. Set it equal to zero.
    c. Solve for the model parameters.

The Classification Recipe

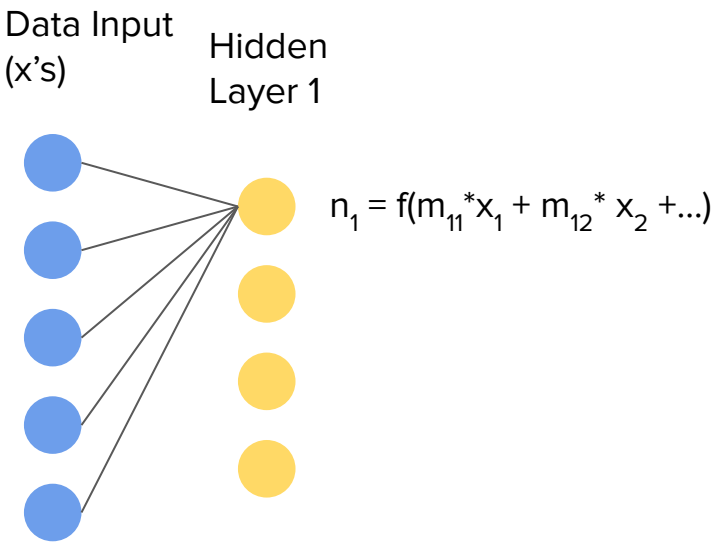
1. Think up some model
2. Feed **data** into the model and make predictions.
    a. The model should squash the predictions to lie between 0 and 1.
3. Calculate the loss between predictions and true values.
4. Determine the model parameters that produce the minimum loss.
    a. Take the derivative of your loss function with respect to the model parameters.
    b. Set it equal to zero.
    c. Solve for the model parameters.

The Classification Recipe

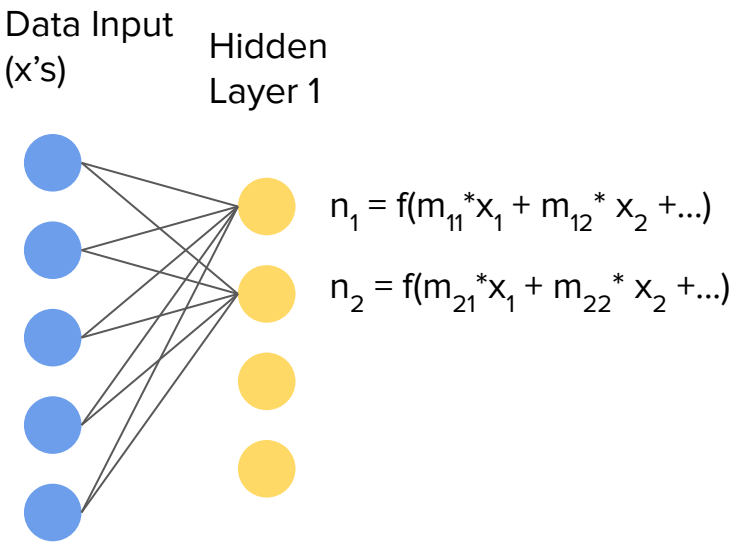
1. Think up some model
2. Feed **data** into the model and make predictions.
    a. The model should squash the predictions to lie between 0 and 1.
3. Calculate the loss between predictions and true values.
    a. The true values will be 0 or 1. The predictions will be between 0 and 1.
4. Determine the model parameters that produce the minimum loss.
    a. Take the derivative of your loss function with respect to the model parameters.
    b. Set it equal to zero.
    c. Solve for the model parameters.

Deep Learning is Just a Fancy Model

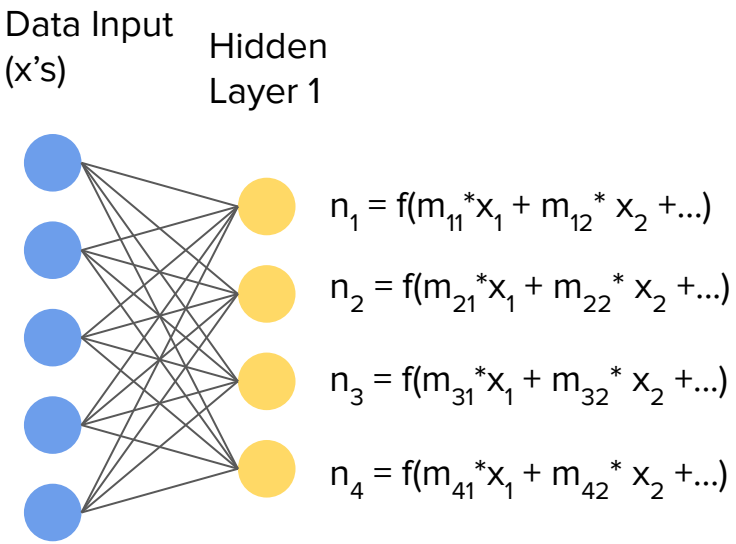
Deep Learning is Just a Fancy Model



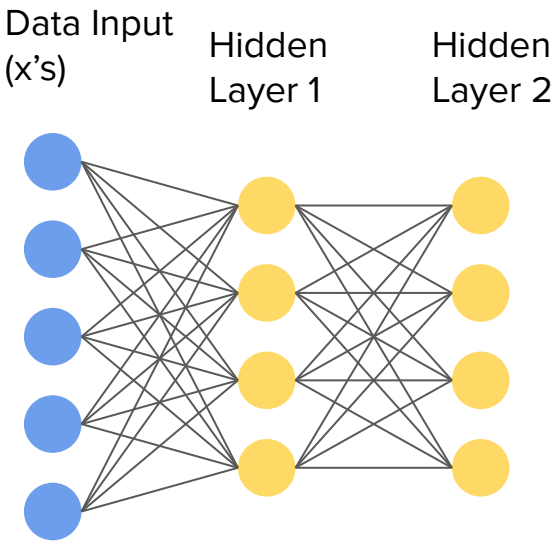
Deep Learning is Just a Fancy Model



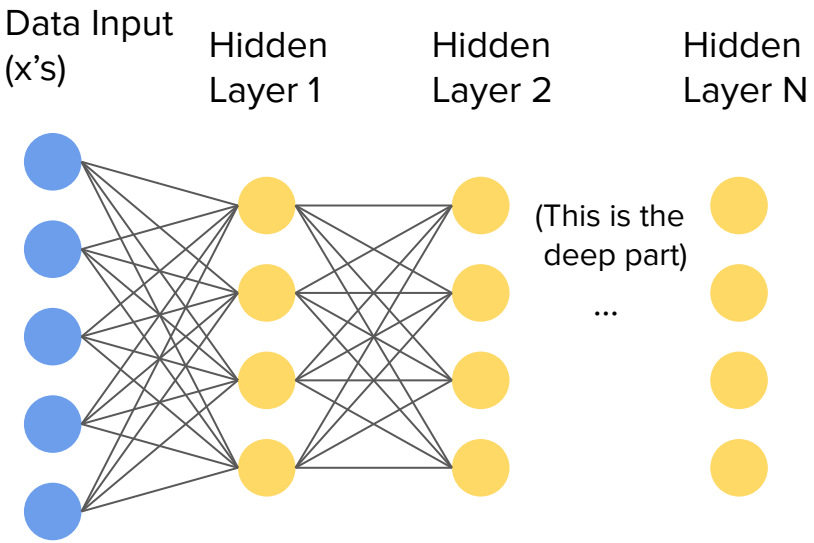
Deep Learning is Just a Fancy Model



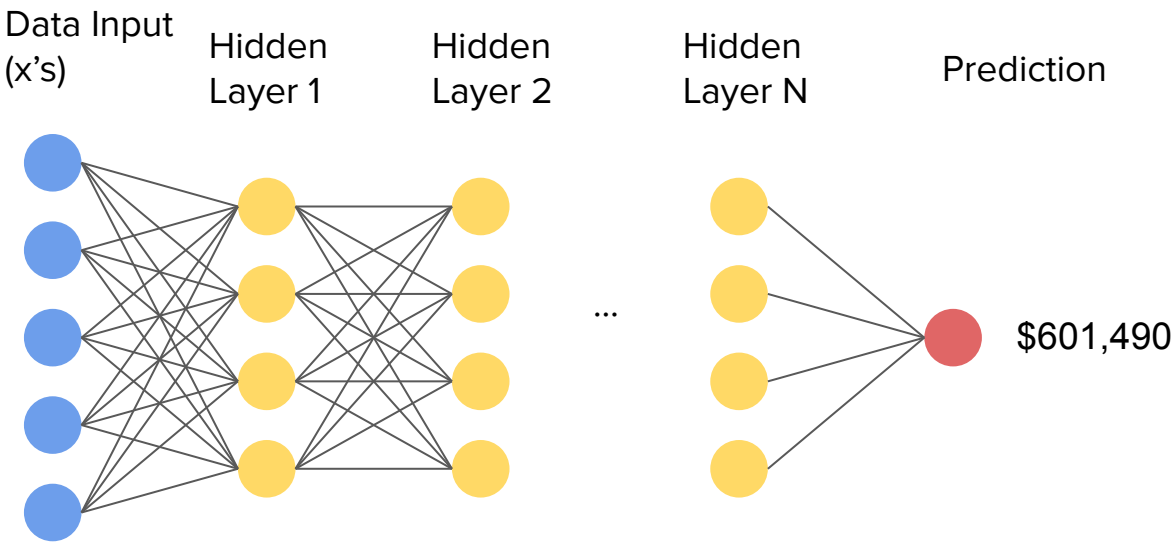
Deep Learning is Just a Fancy Model



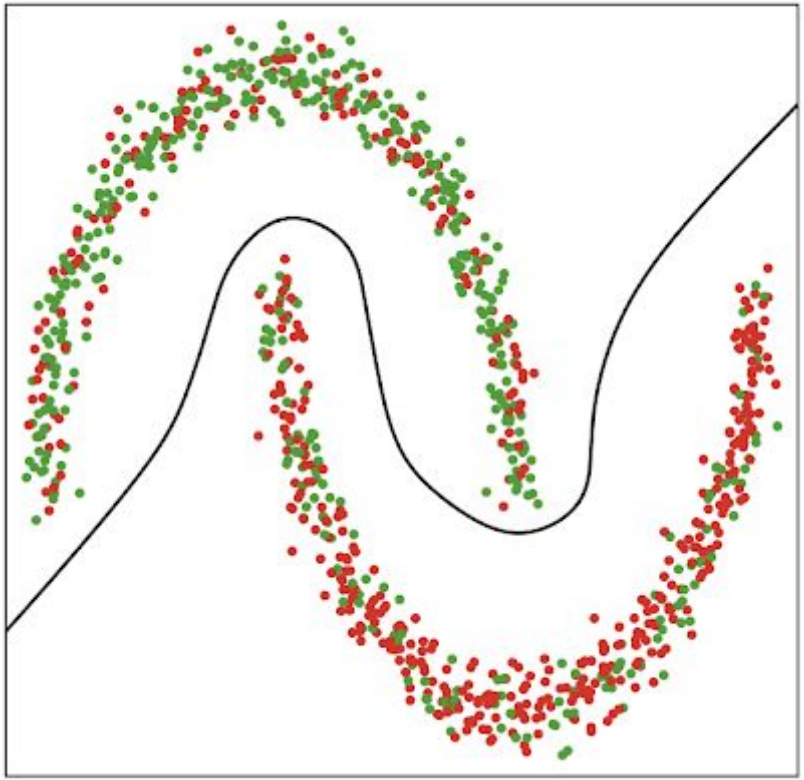
Deep Learning is Just a Fancy Model



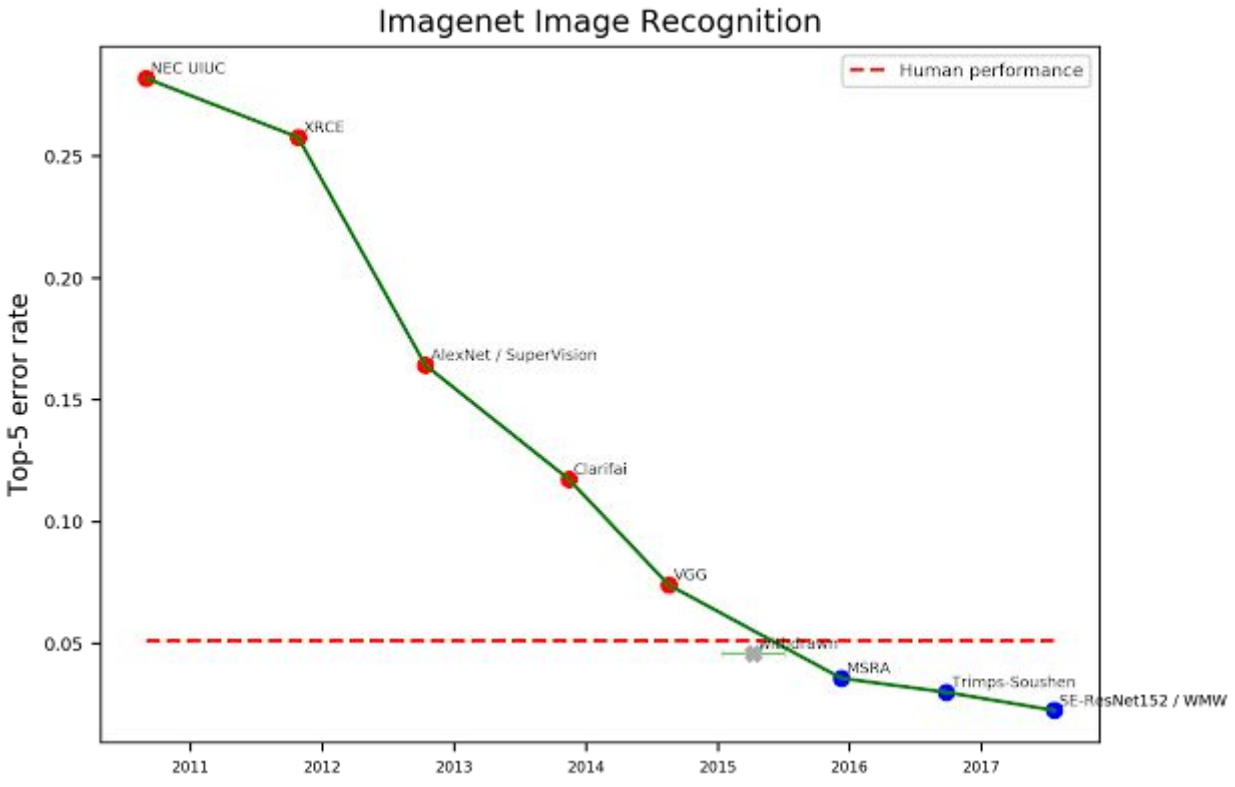
Deep Learning is Just a Fancy Model



Why Do Deep Learning? Handle Nonlinearities



Why Do Deep Learning? It Works.



Computer Walkthrough

Python is the
second-best language
for everything

The worst part about Python is installing it and its libraries

In class

1. Go to the Week 1 folder of the class repo https://github.com/jacopotagliabue/MLSys-NYU-2023
2. Walk through the fundamentals of the command line, Python, and Rye.

Your Homework

1. Go to the Week 1 folder of the class repo https://github.com/jacopotagliabue/MLSys-NYU-2023
2. Follow the Computer Setup for
    a. Code Editor
    b. Installing Git
    c. Using GitHub
    d. Installing Python