

NYU FRE 7773 - Fall 2022

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.
- [Founder of Tooso](#) (acquired by TSX:CVO), director of AI at Coveo (2019-2022).
- 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 2022: 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 2022: 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 2022: House-keeping

- Your TA is: **Guojun Chen** (who attended FRE 2021!). 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 2022: 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*.
 - Your TA will circulate some GDocs so you can start putting some ideas down

FRE Fall 2022: Evaluation

To give you some context, last 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 2022 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 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.
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 this course is all about *machine learning that works in practice*). 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 and Google your way out of programming issues (like all professionals do!)

What FRE 2022 is NOT

Abubakar Abid @abidlabs · Aug 4

The way we teach ML doesn't consider real-world reliability. We teach:

1. pick static dataset (MNIST)
2. split into train/test
3. train until high test acc
4. move on

Instead of moving on, we should:

5. deploy the model
6. get users to break models
7. adapt dataset to fix issues

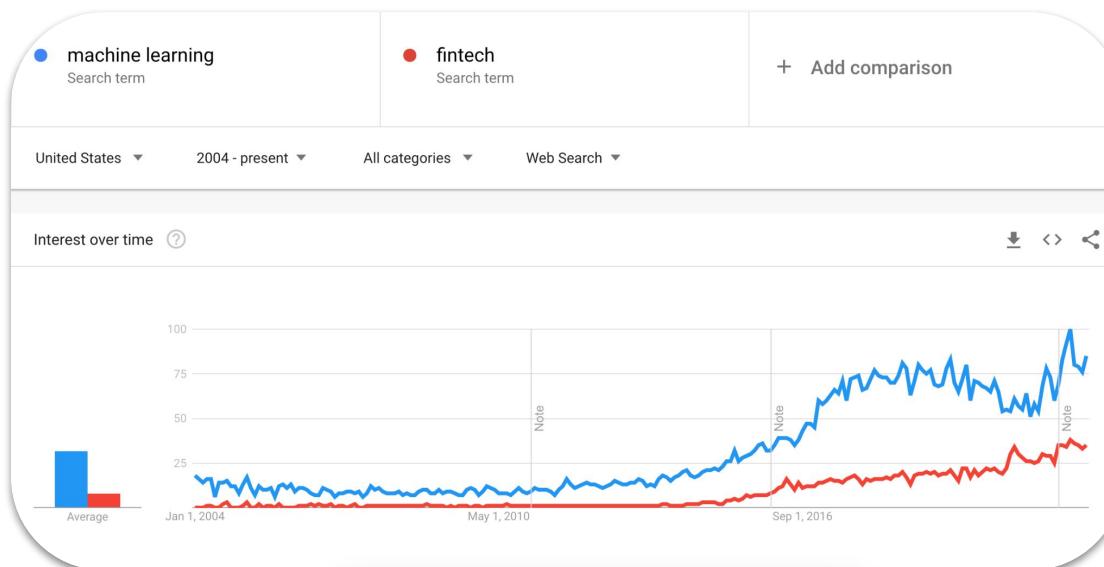
30 125 708

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

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

TEXT PROMPT an armchair in the shape of an avocado. . . .

AI-GENERATED IMAGES



Edit prompt or view more images↓

This block shows a user interface for generating images from text prompts. It includes a text input field, a button to edit the prompt, and a grid of five generated images. The images depict an armchair designed to look like a ripe avocado, with various color schemes (yellow, green, orange) and different base designs.

TECH ARTIFICIAL INTELLIGENCE

GitHub and OpenAI launch a new AI tool that generates its own code

Microsoft gets a taste of OpenAI's tech

By [Dave Gershgorin](#) | Jun 29, 2021, 1:46pm EDT

[f](#) [t](#) [e](#) SHARE

This block is a news snippet from TechCrunch. It features a header, a sub-headline, author information, a timestamp, and social sharing icons.

Meet GPT-3. It Has Learned to Code (and Blog and Argue).

The latest natural-language system generates tweets, pens poetry, summarizes emails, answers trivia questions, translates languages and even writes its own computer programs.

[Let's try DALLE!](#)

The golden era of Machine Learning

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

RecSys is a 16BN dollar market

Market Overview

The Recommendation Engine market was valued at USD 2.12 billion in 2020, and it is expected to reach USD 15.13 billion by 2026, registering a CAGR of 37.46% during the period of 2021-2026. With the growing amount of information over the internet along with a significant rise in the number of users, it is becoming essential for companies to search, map, and provide them with the relevant chunk of information according to their preferences and tastes.

The golden era of Machine Learning

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

Meta mentions RecSys in earning calls!

This quarter, we saw a more than 30% increase in the time that people spent engaging with Reels across Facebook and Instagram. AI advances are driving a lot of these improvements. And one example is that after launching a new large AI model for recommendations, we saw a 15% increase in watch time in the Reels video player on Facebook alone. So I think that there are many improvements like this that we're going to be able to continue to make.

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”

2021



Asset Management

VISE EquityBee DRIVE WEALTH AlphaSense

Payments Processing & Networks

Bolt cedar mollie Airwallex pro Orum checkout.com Nium pagimongo Cashfree BharatPe GOCARDLESS Flutterwave stripe ZEPZ OPay PayCargo Enfuse RELAY Rapyd Razorpay codirect balance Peppstand Pine Labs xfers ripple SpotOn melio xendit EBANX recharge



Insurance

Extend Ladder wefox *alan CyberCube TIGER EBAY ARTURO CAMBRIDGE ZEGO BESTOW socatra digit PIE INSURANCE voice COWELL ETHOS NEXT Thimble Coalition luko CORVUS EMBROKER RIBOT GERMANY Policygenius marley

Business Lending & Finance

spendesk ramp CLEARCO Konflic pipe Tradeshift PEO Bluevine Capchase Rho C2FO MERCURY BREX airbase tide PRODUCEPAY Fundbox

Financial Services & Automation

gia Octopus (n|s) HYPERFACE Notarize unqork Fairmarket

Core Banking & Infrastructure

ALLOY AMOUNT Finxact unit Solarisbank MX TRUEAYER zeta MAMBU Modul NYMBUS FINIX Reserve cross river PRIME harmi MANTL Thunes Raisbank Lithic HASH belvo codat PLAIN

Credit Score & Analytics

Borrowell credit sesame NOVA CREDIT

Digital Banking

cfrei nu Starling Bank chime GREENLIGHT step Varo Current monzo qonto Revolut LUNAR GREENWOOD valo

Mobile Wallets & Remittances

Chipper PONDEADERA PAYSENDE FAZZFINANCIAL toss

Cryptocurrency

CERTINT Bitcoin Suisse ANCHORIAS copper FTX BYNTHETIX MESSARI BLOCKDAGOMON Chainalysis Blockchain.com Dapper BlockFi TAXbit bitpanda Bitwise Lukka PAIXOS FALCONX Fireblocks alchemy MobileCom Bitso celo Finoa AMBER Ledger CoinDCX wyrre

General Lending & Marketplaces

CAPITAL FLOAT upgrade credites WeLab

Note: Companies are private as of 9/14/21.

Created by You. Powered by

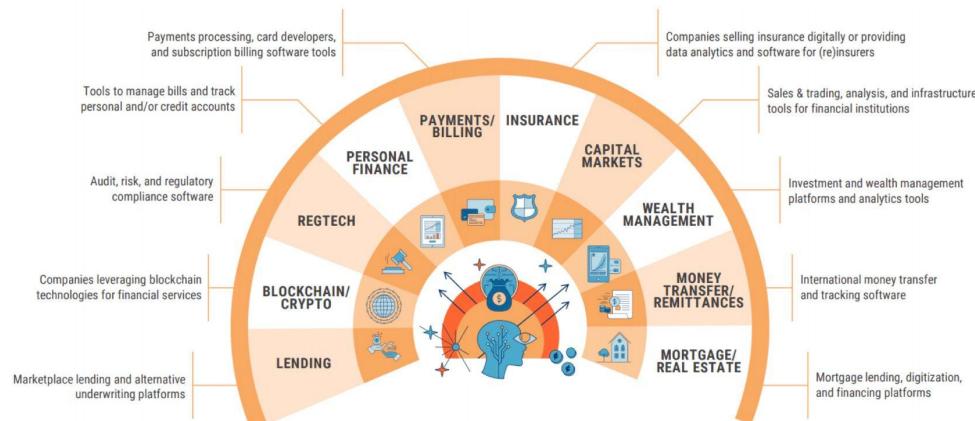
The intersection of tech and finance

- 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 when building your pipeline: 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

Tech At Bloomberg

Topics ▾ Events ▾ About ▾ Git

2021

kōan: A Corrected CBOW Implementation. **Ozan İrsøy, Adrian Benton** and Karl Stratos. arXiv. (Code Repository)

Keynote - Information in Context: Financial Conversations & News Flows. Gideon Mann. Workshop on Knowledge Discovery from Unstructured Data in Financial Services at AAAI 2021. (Video)

Dual Reinforcement-Based Specification Generation for Image De-Rendering. Ramakanth Pasunuru, **David Rosenberg, Gideon Mann** and Mohit Bansal. Workshop on Scientific Document Understanding at AAAI 2021. (Video)

Contextualizing Trending Entities in News Stories. **Marco Ponza, Diego**

Hugging Face

Search models, datasets, user:

Models Datasets Resources

Dataset: financial_phrasebank like 0

Tasks: multi-class-classification sentiment-classification Task Categories: text-classification Language

Size Categories: 1K<n<10K Licenses: cc-by-nc-sa-3.0 Language Creators: found Annotations Creators

Source Datasets: original

Dataset Structure Dataset Card for financial_phrasebank

Data Instances Data Fields Data Splits

Dataset Summary

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

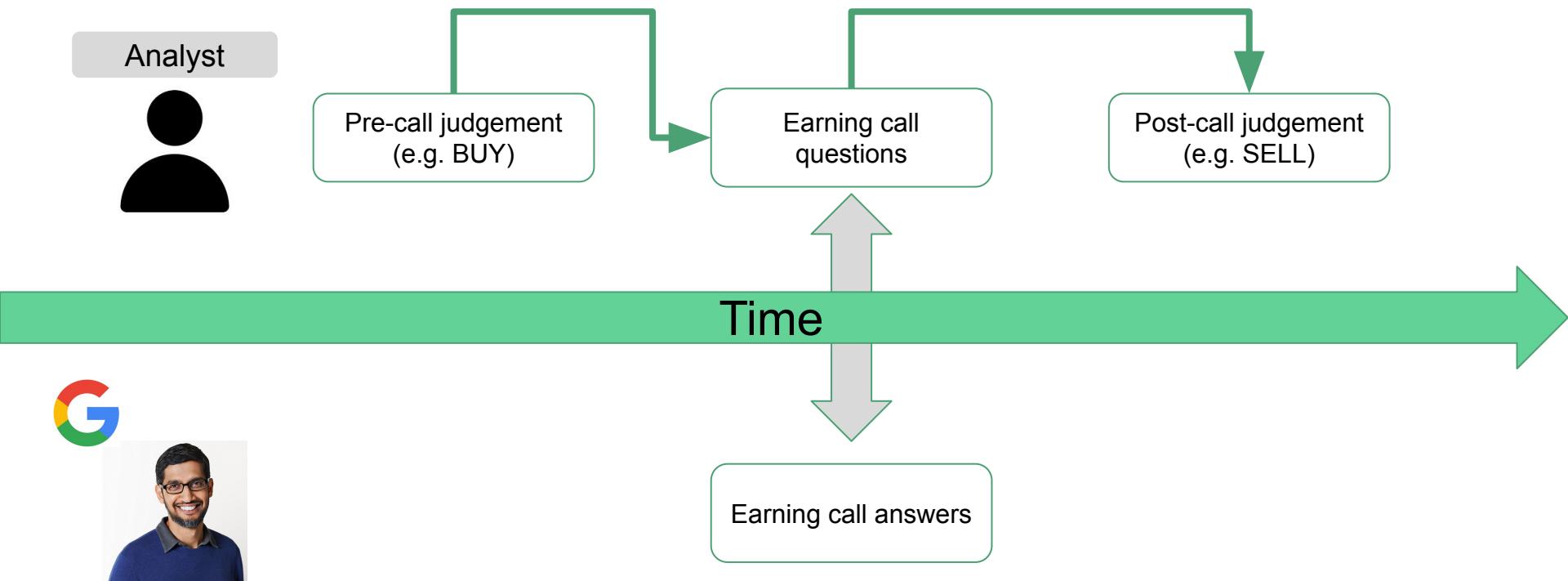
Bloomberg LP
astent@bloomberg.net

Abstract

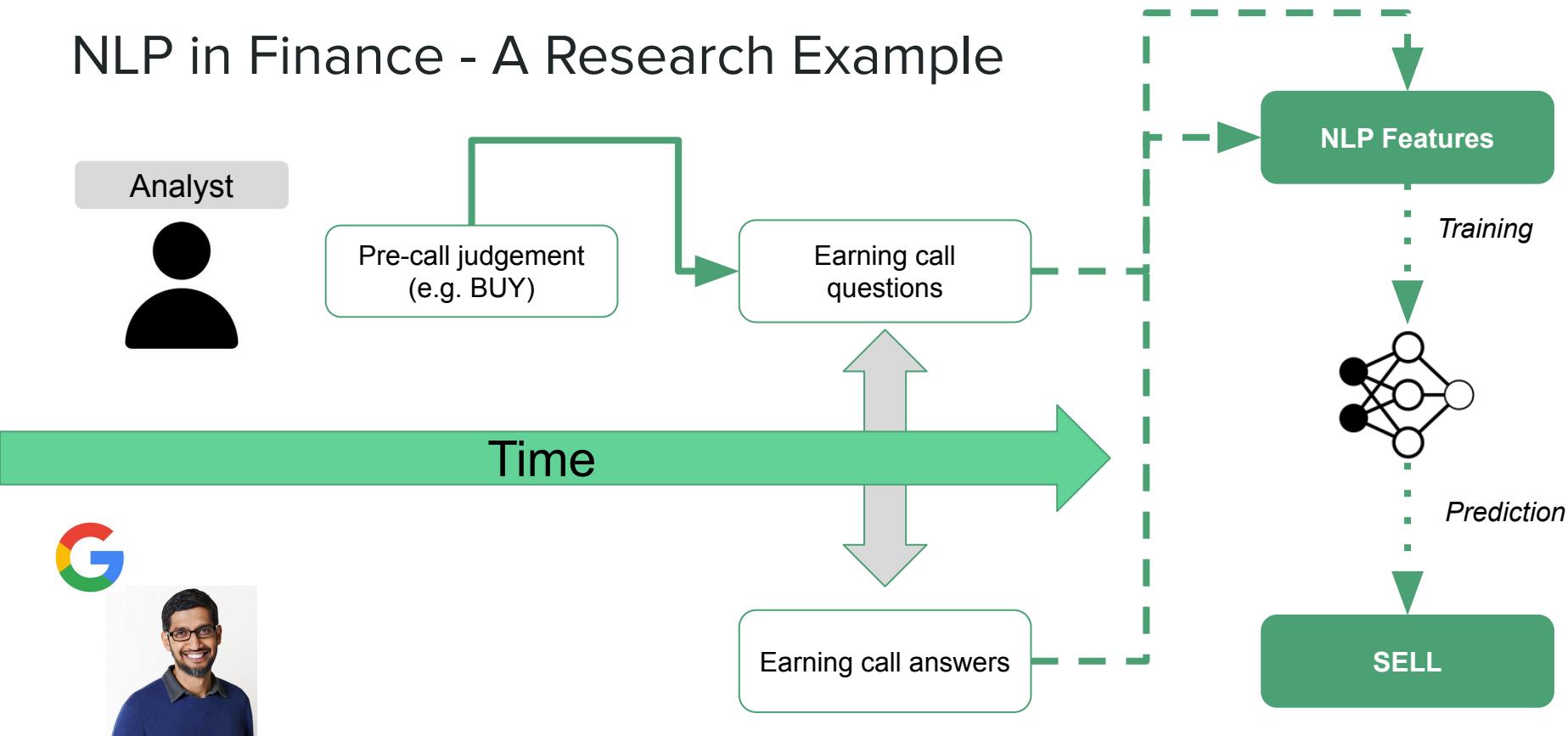
Every fiscal quarter, companies hold *earnings calls* in which company executives respond

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**? ”

Why do 87% of data science projects never make it into production?

VB Staff

July 19, 2019 4:10 AM

f t in

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,lora]@google.com

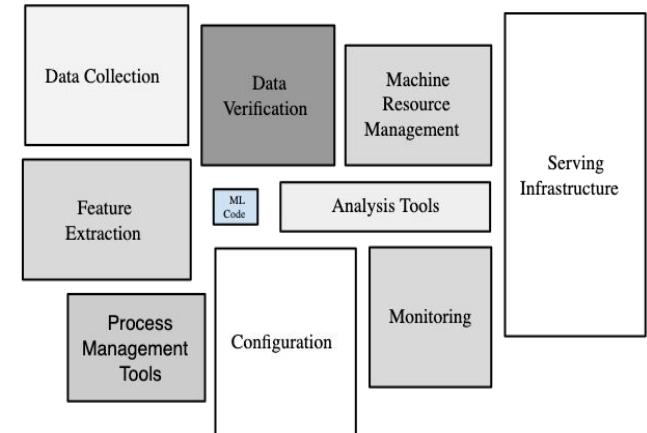
Google Research

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.

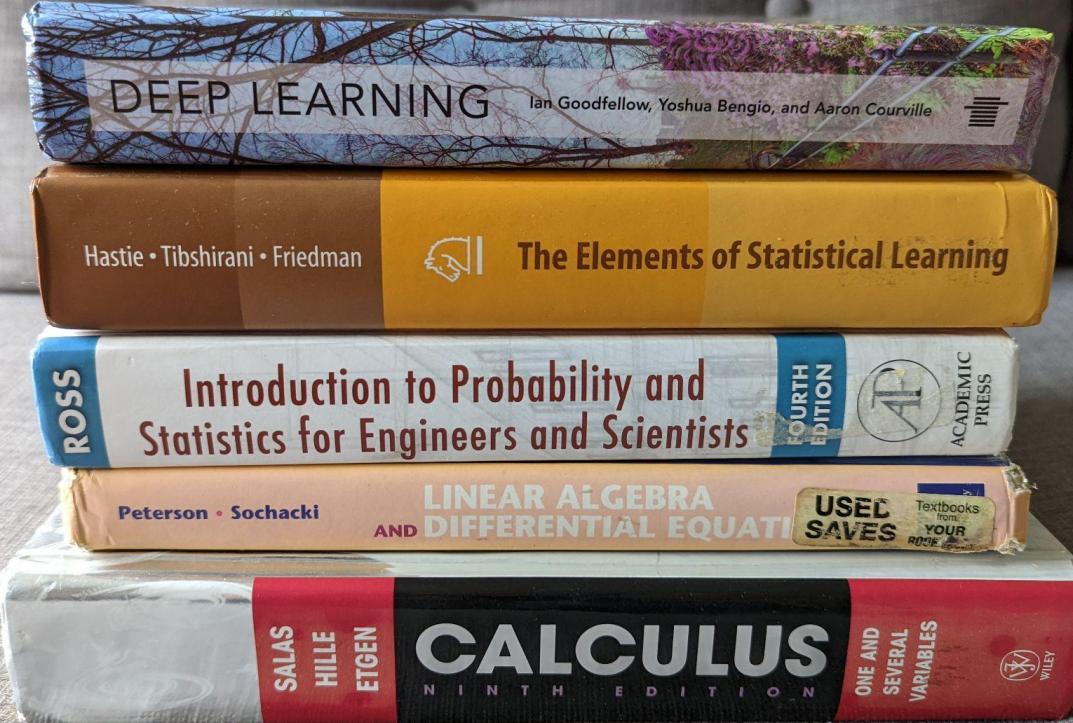
lionized work of building novel models and algorithms [46, 125]. Intuitively, AI developers understand that data quality matters, often spending inordinate amounts of time on data tasks [60]. In practice, most organisations fail to create or meet any data quality standards

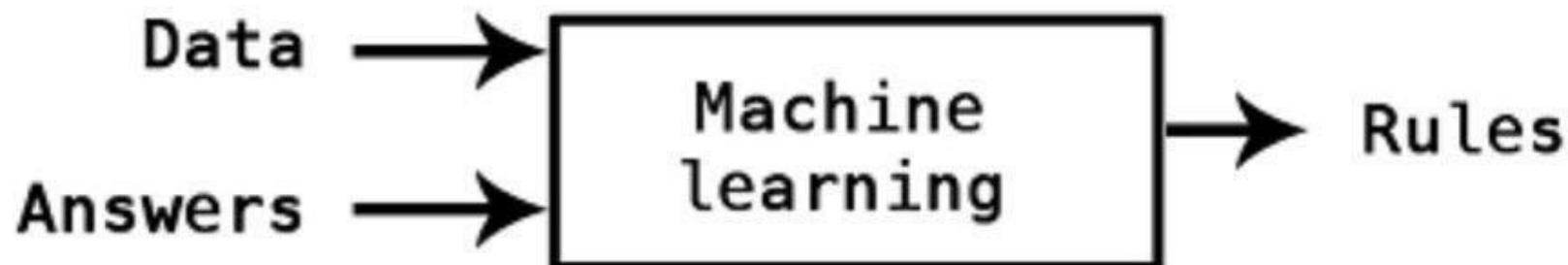
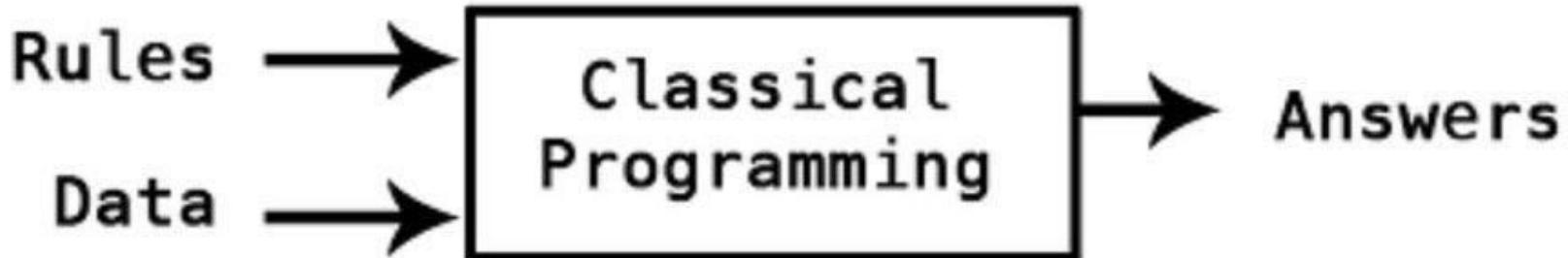


Putting it all together

- Ethan
 - Python / coding fundamentals
 - Introduction to Machine Learning
 - Fraud detection
 - **HERE:** you should be able to have a *local*, well structured, ML project that trains a model.
- Jacopo
 - Moving from your laptop to the cloud
 - RecSys
 - Monitoring and testing
 - **HERE:** you should be able to move instrument your *local* project to be cloud-ready, and serve predictions from an endpoint.

🙌 Intro to Machine Learning & Deep Learning 🙌







```
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
```



```
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



Classical Programming

Data



Answers

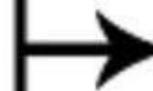
```
def send_coupon(user):
    if user.is_new:
        if user.device == "ios":
            return True
        elif user.device == "android" and user.os_major_version >= 11:
            return True
    if (
        user.referral_source in ("facebook", "organic")
        and user.days_since_last_visit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

Rules



Classical Programming

Data

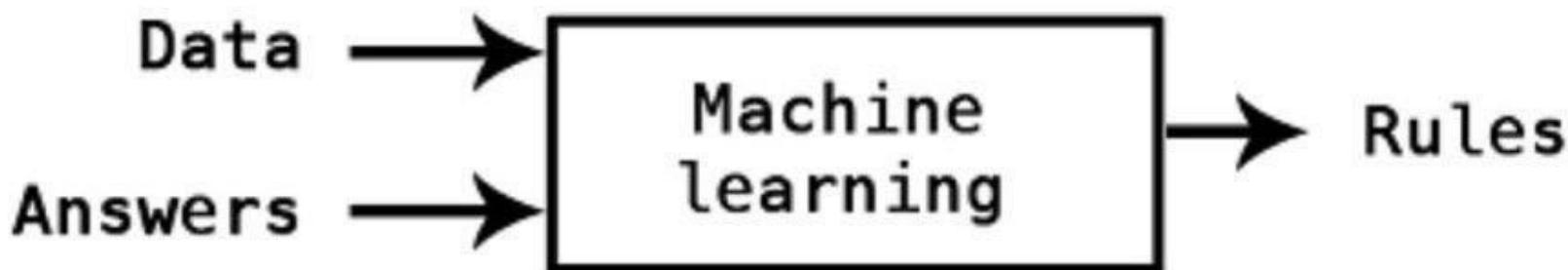


Answers

```
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)
```



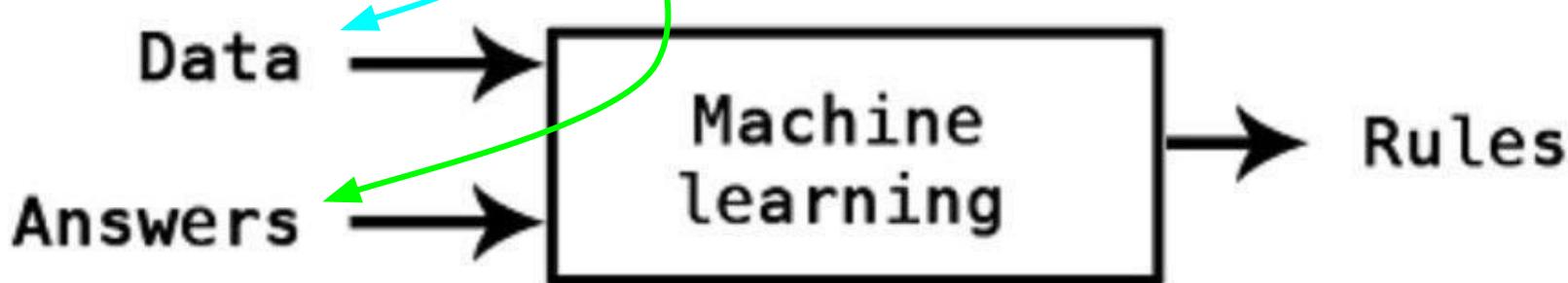


```
users = [  
    User(is_new=False, device="ios", os_major_version=13, referral_source="facebook", days_since_last_visit=10),  
    User(is_new=True, device="android", os_major_version=11, referral_source="search", days_since_last_visit=None),  
    User(is_new=False, device="android", os_major_version=10,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



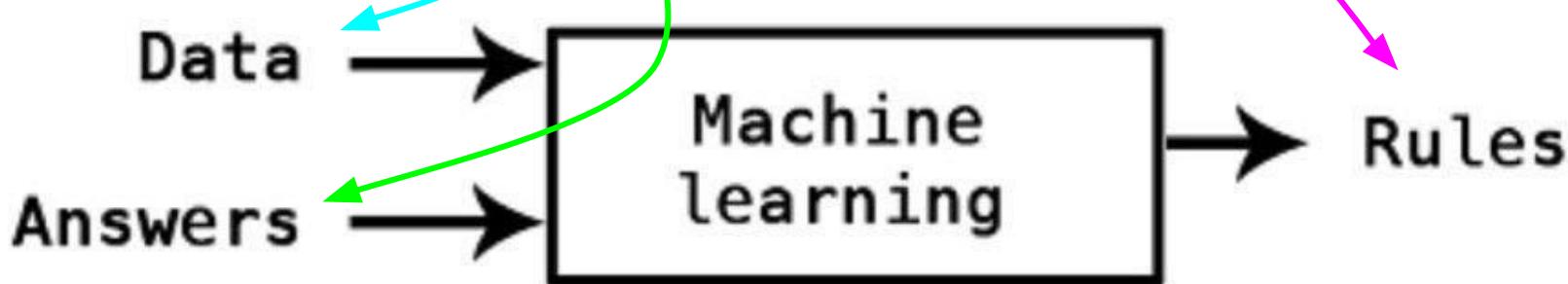


```
users = [  
    User(is_new=False, device="ios", os_major_version=13, referral_source="facebook", days_since_last_visit=10),  
    User(is_new=True, device="android", os_major_version=11, referral_source="search", days_since_last_visit=None),  
    User(is_new=False, device="android", os_major_version=10, 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_source="Facebook", user_id="1", user_last_visit=10),  
    User(is_new=True, device="android", os_major_version=11, referral_source="Instagram", user_id="2", user_last_visit=None),  
    User(is_new=False, device="android", os_major_version=12, referral_source="Twitter", user_id="3", user_last_visit=6),  
    ...  
]
```

```
conversions = [  
    True,  
    False,  
    False,  
    ...  
]
```

```
model = train(users,
```

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

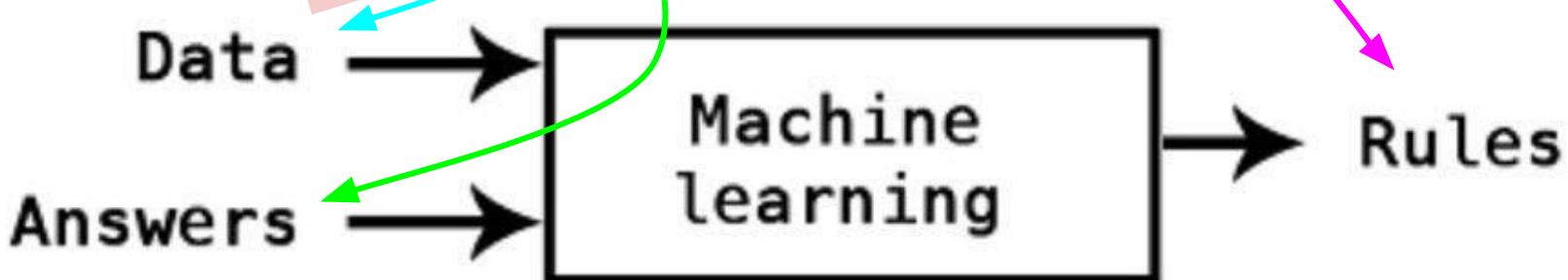
How do you learn the rules?



```
users = [  
    User(is_new=False, device="ios", os_major_version=13, referral_source="Facebook", device_last_visit=10),  
    User(is_new=True, device="android", os_major_version=11, referral_source="Twitter", device_last_visit=None),  
    User(is_new=False, device="android", os_major_version=12, referral_source="Instagram", device_last_visit=6),  
    ...  
]  
  
conversions = [  
    True,  
    False,  
    False,  
    ...  
]  
  
model = train(users, conversions)  
  
def send_coupon(user, conversion):  
    return model.predict([[user.device, user.os_major_version, conversion]])
```

How do you learn the rules?

MATH





```
users = [  
    User(is_new=False, device="ios", os_major_version=13, referral_source="Facebook",  
    User(is_new=True, device="android", os_major_version=11, ref...  
    User(is_new=False, device="android", os_major_version=12, ...  
    ...  
]
```

```
conversions = [  
    True,  
    False,  
    False,  
    ...  
]
```

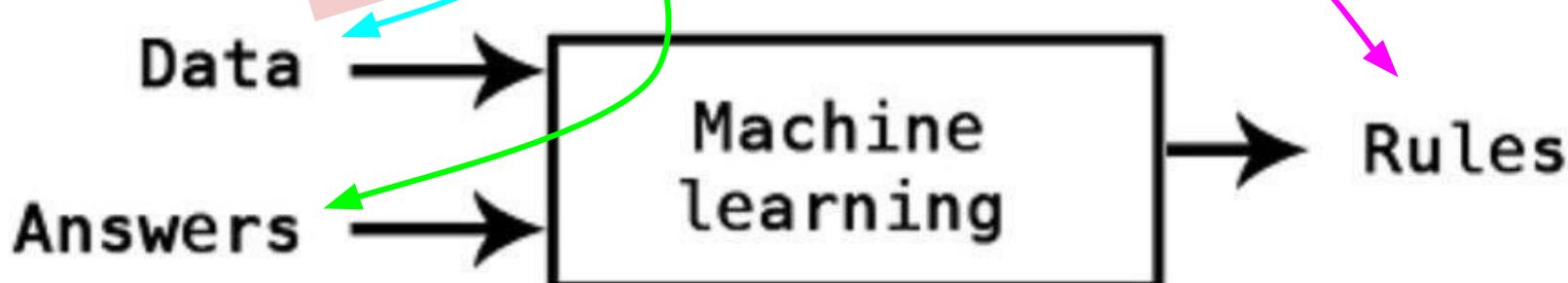
```
model = train(users,
```

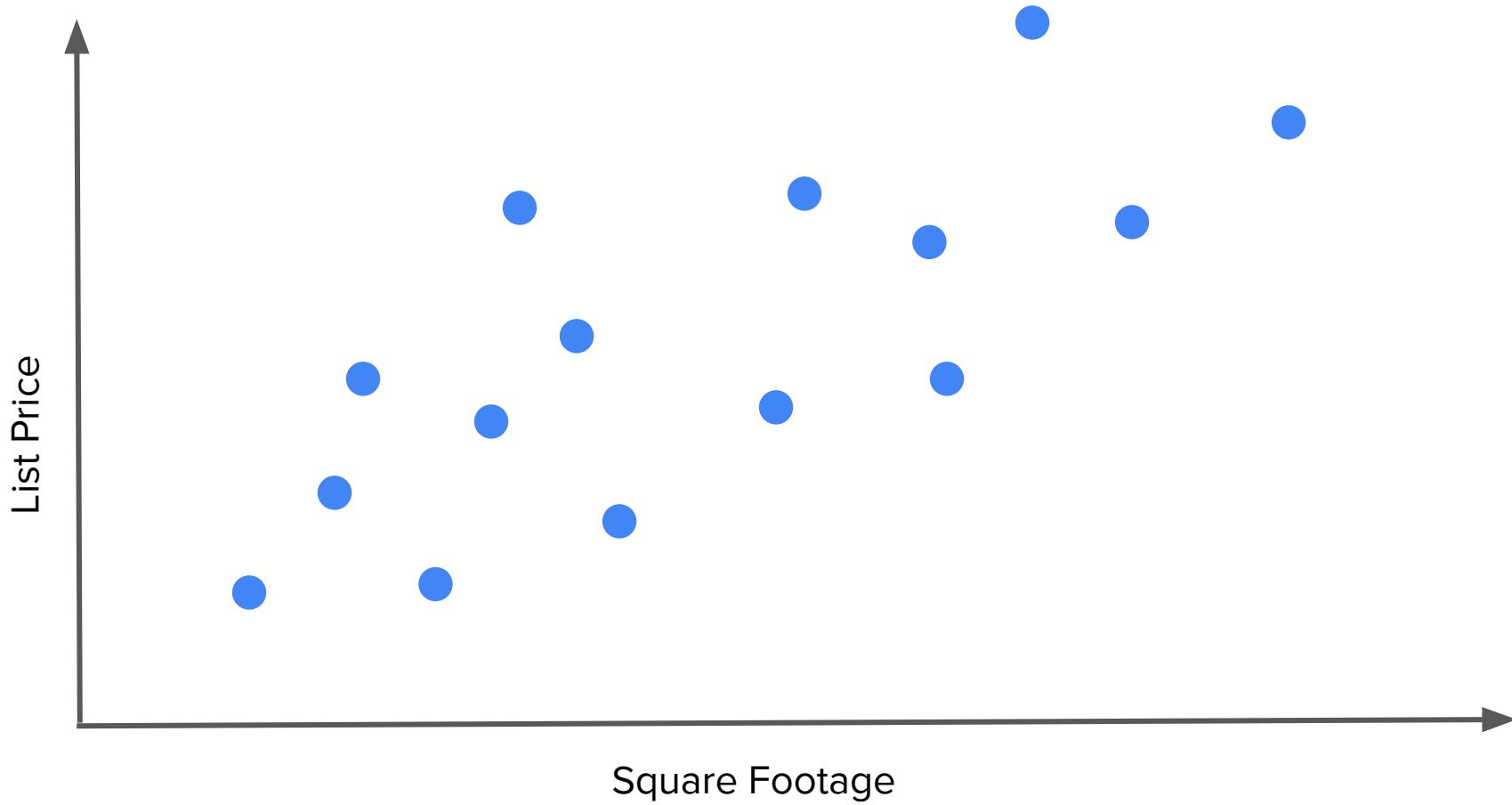
```
def send_coupon(user,  
    return model.predict(...)
```

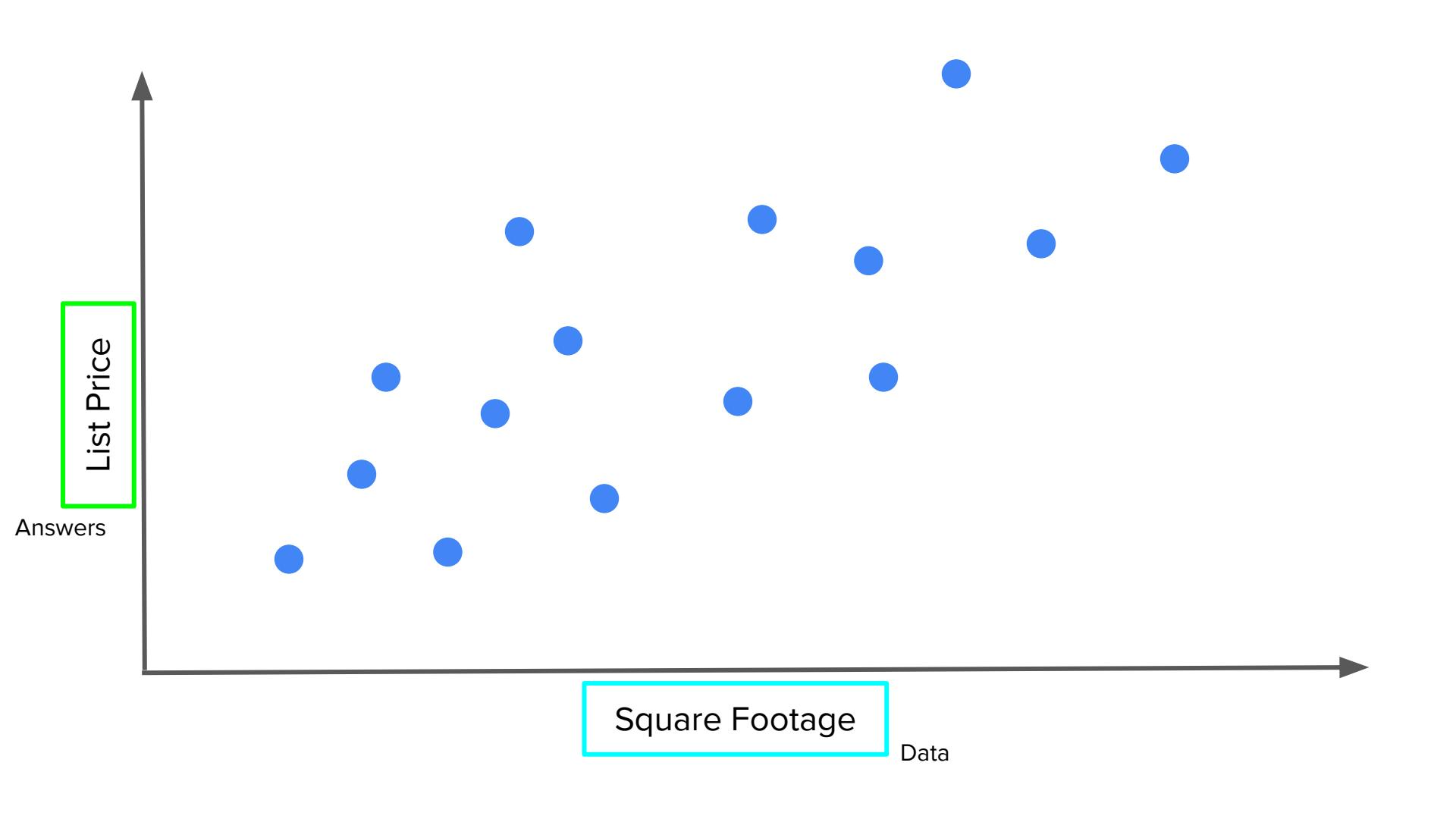
How do you learn the rules?

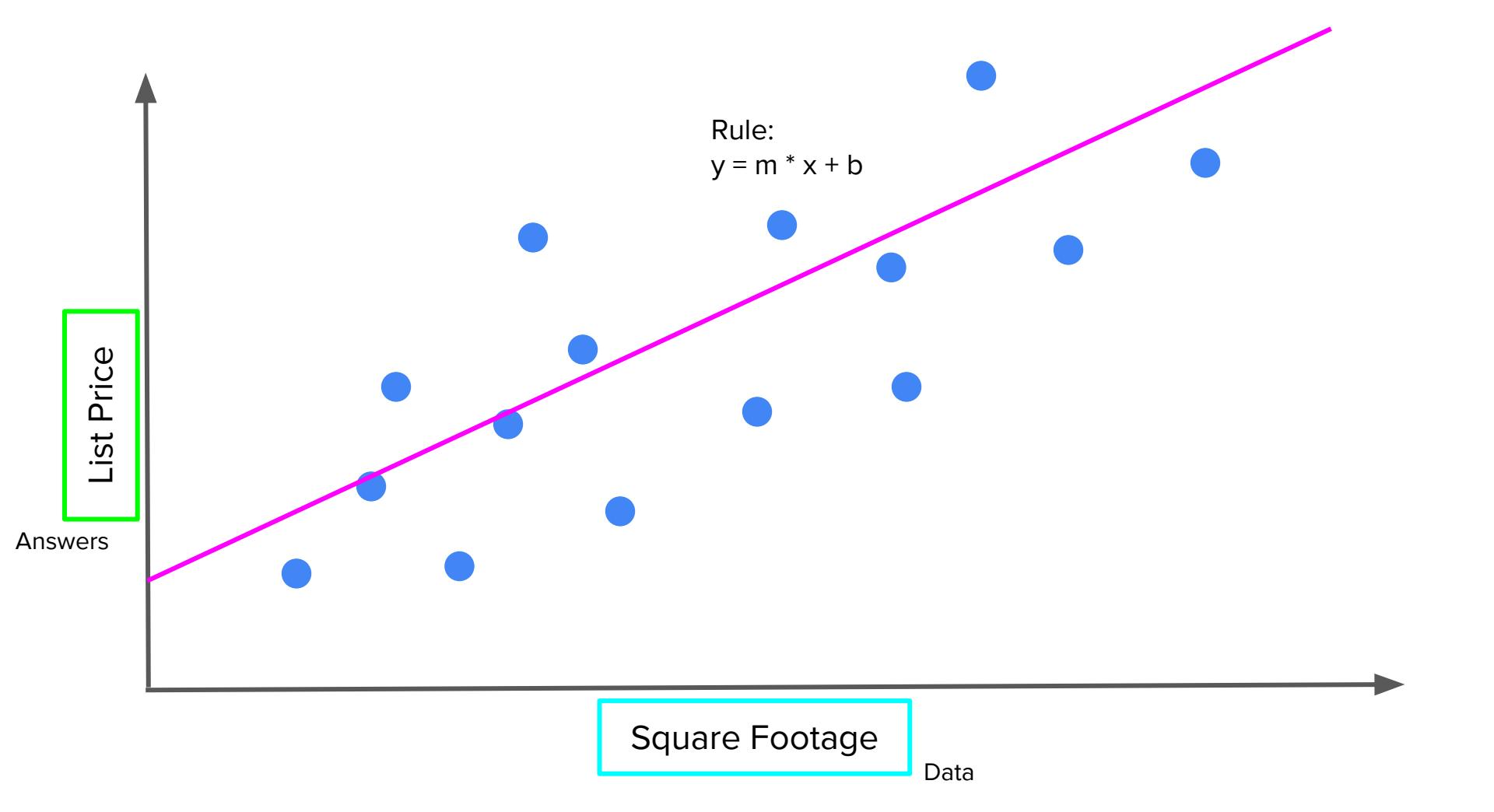
Free
Software!

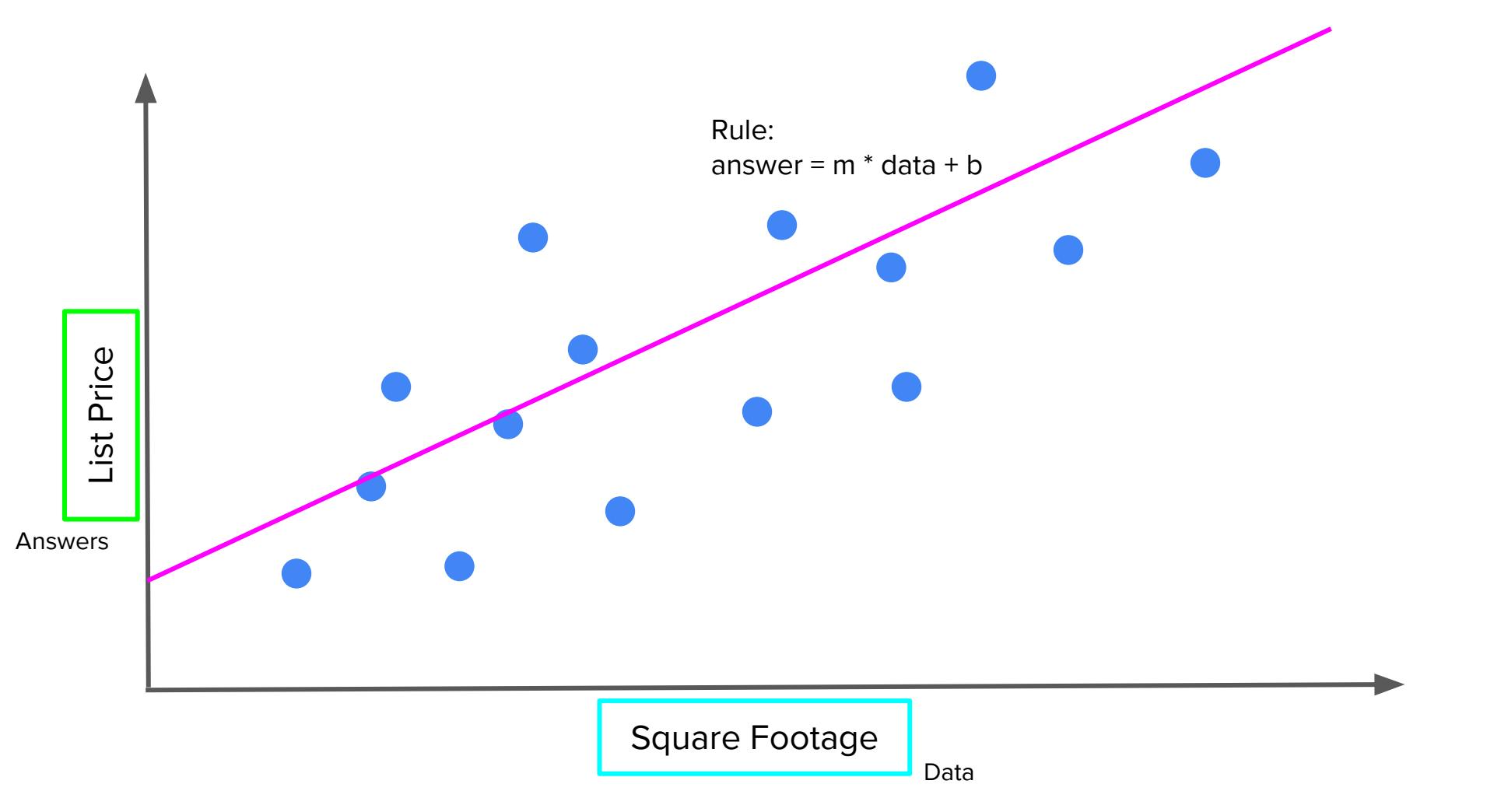
MATH





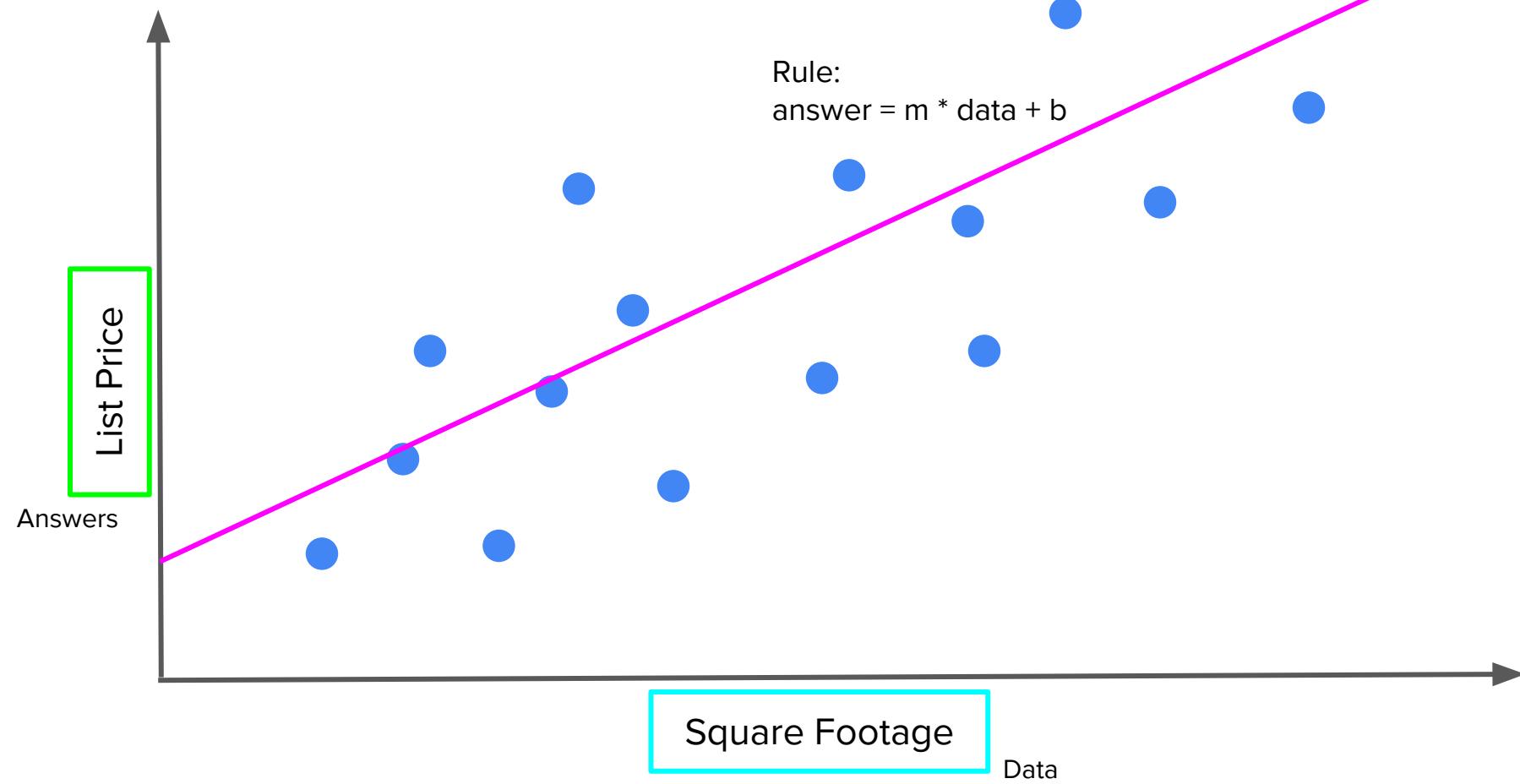






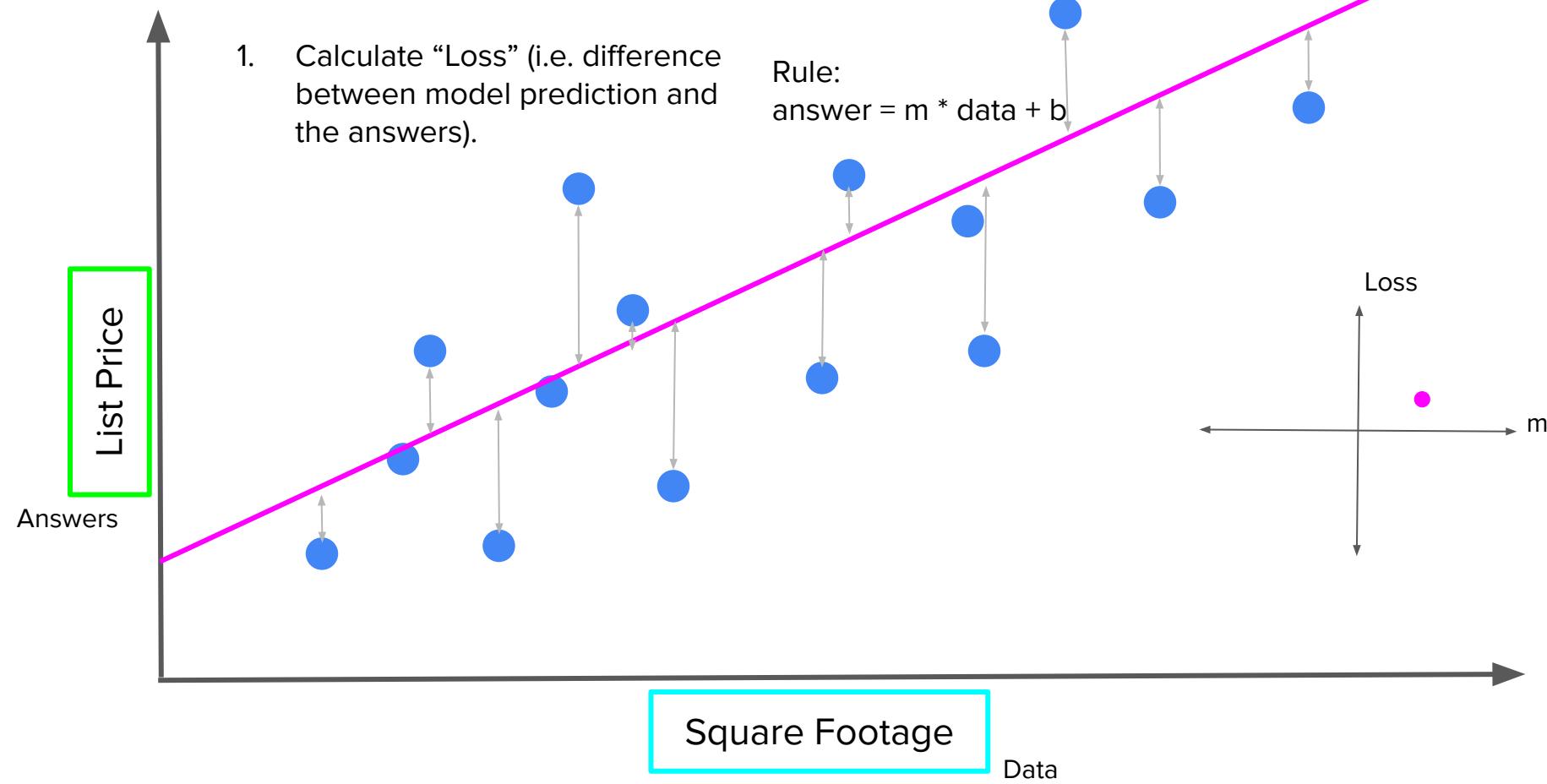
How do we learn the rules (m and b)?

Rule:
 $\text{answer} = m * \text{data} + b$



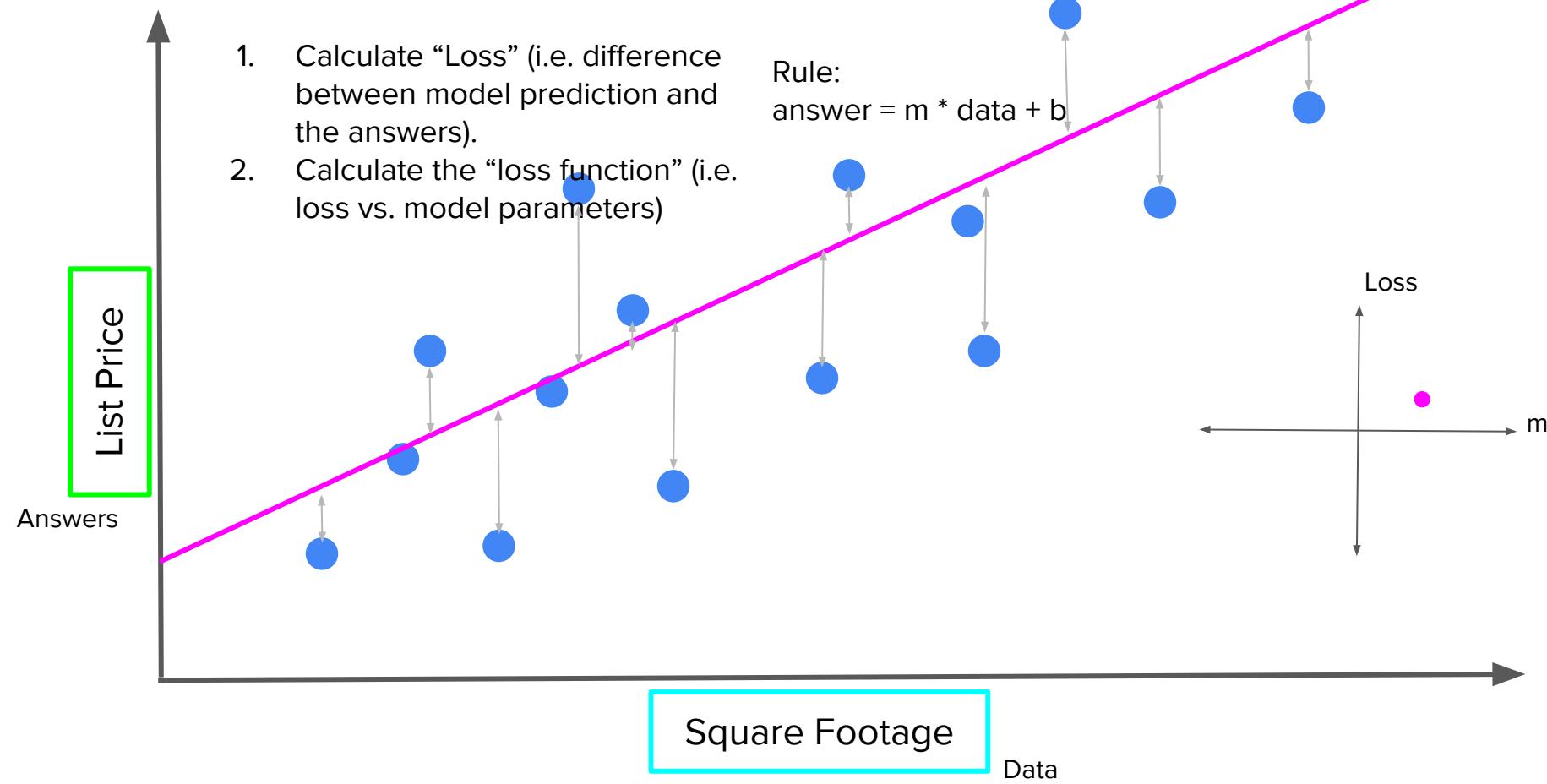
How do we learn the rules (m and b)?

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



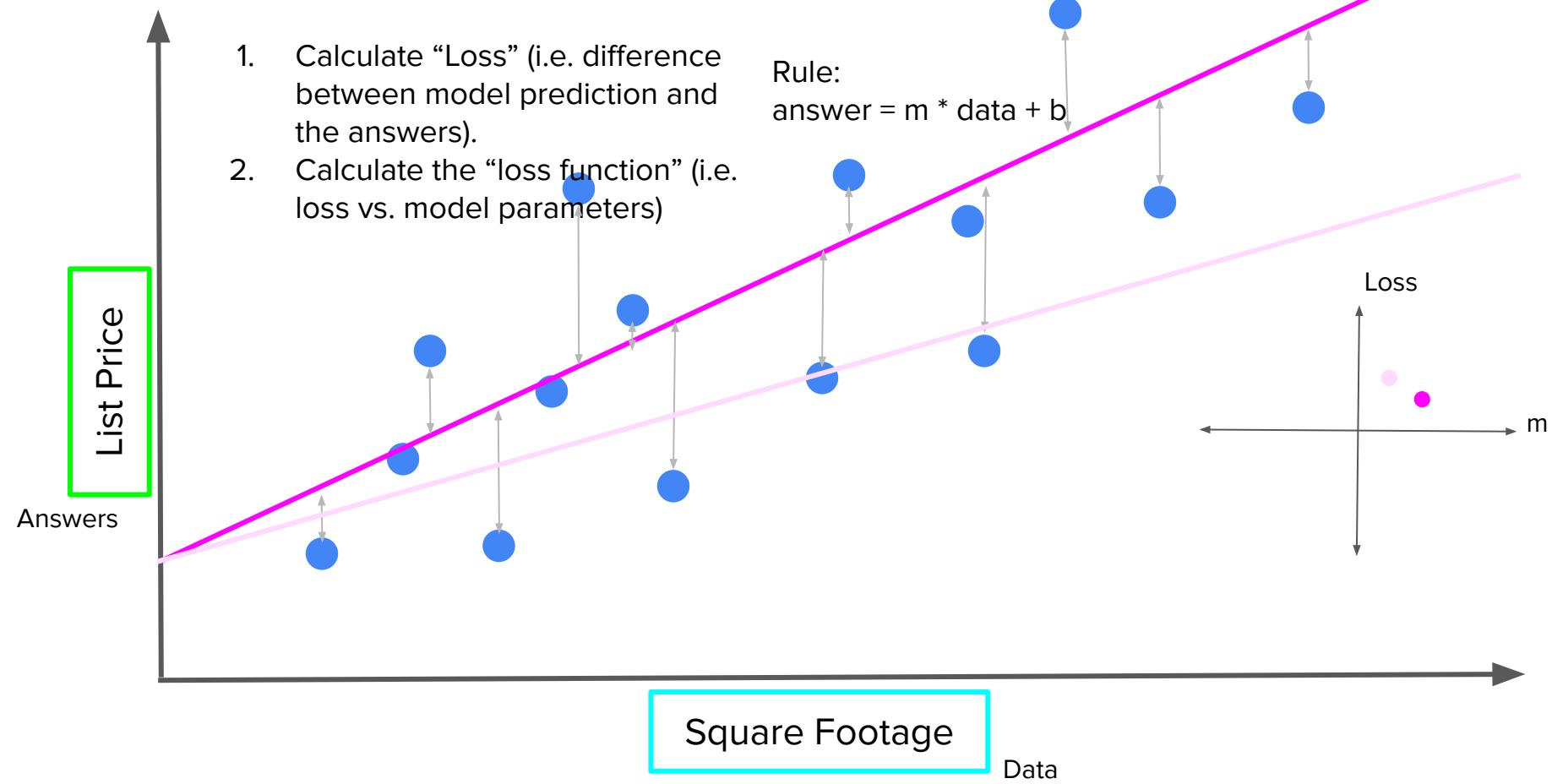
How do we learn the rules (m and b)?

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



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)

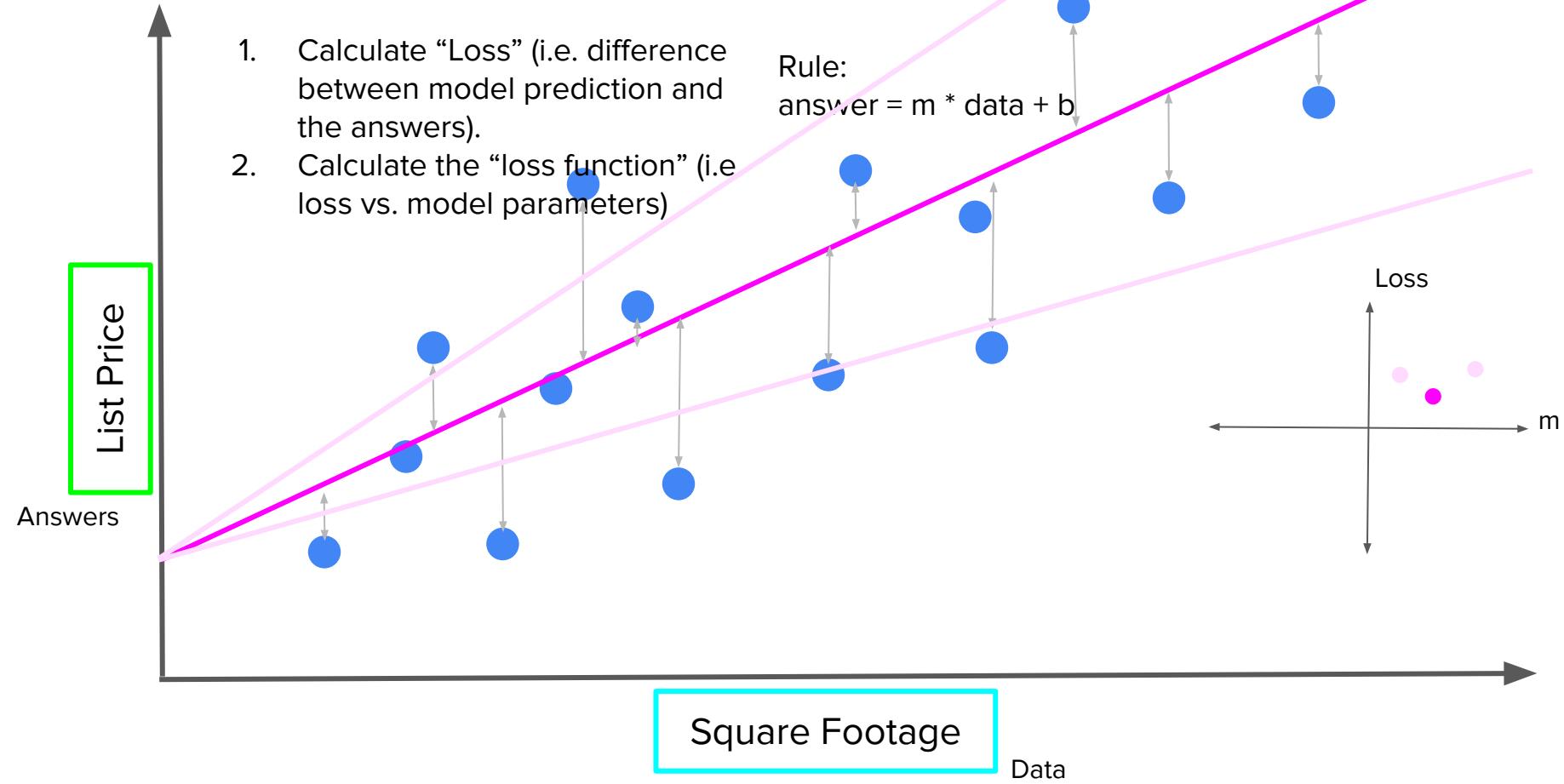


How do we learn the rules (m and b)?

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

Rule:

$$\text{answer} = m * \text{data} + b$$

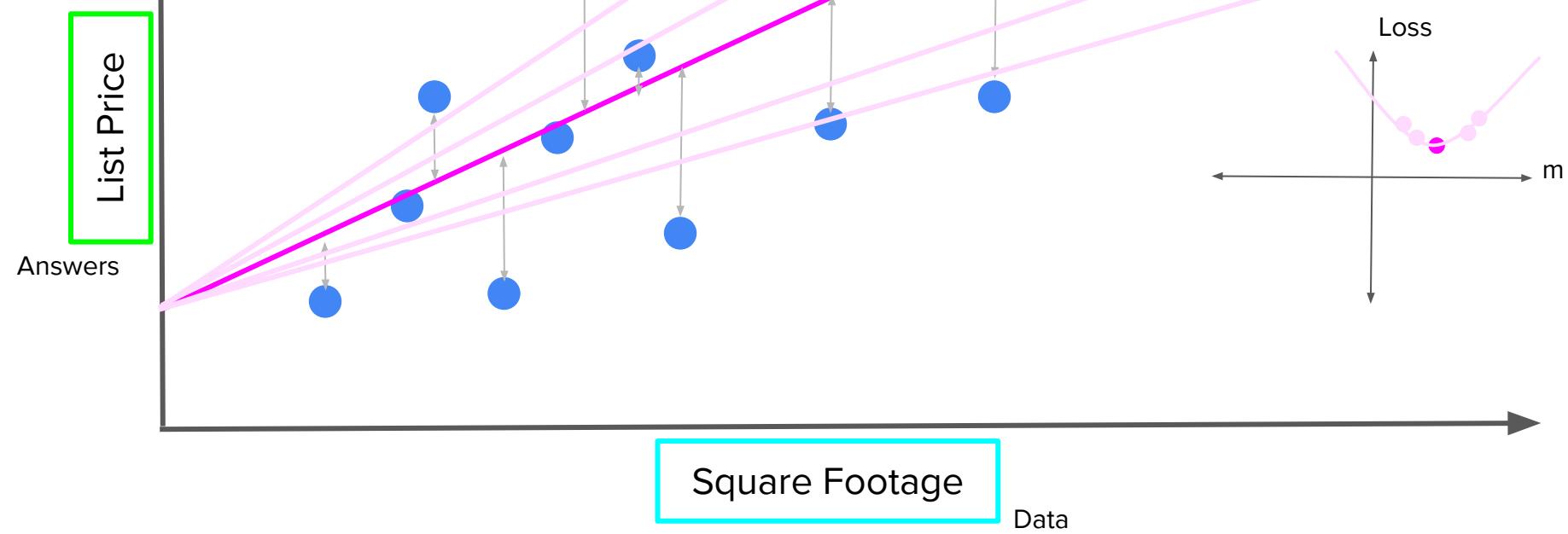


How do we learn the rules (m and b)?

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

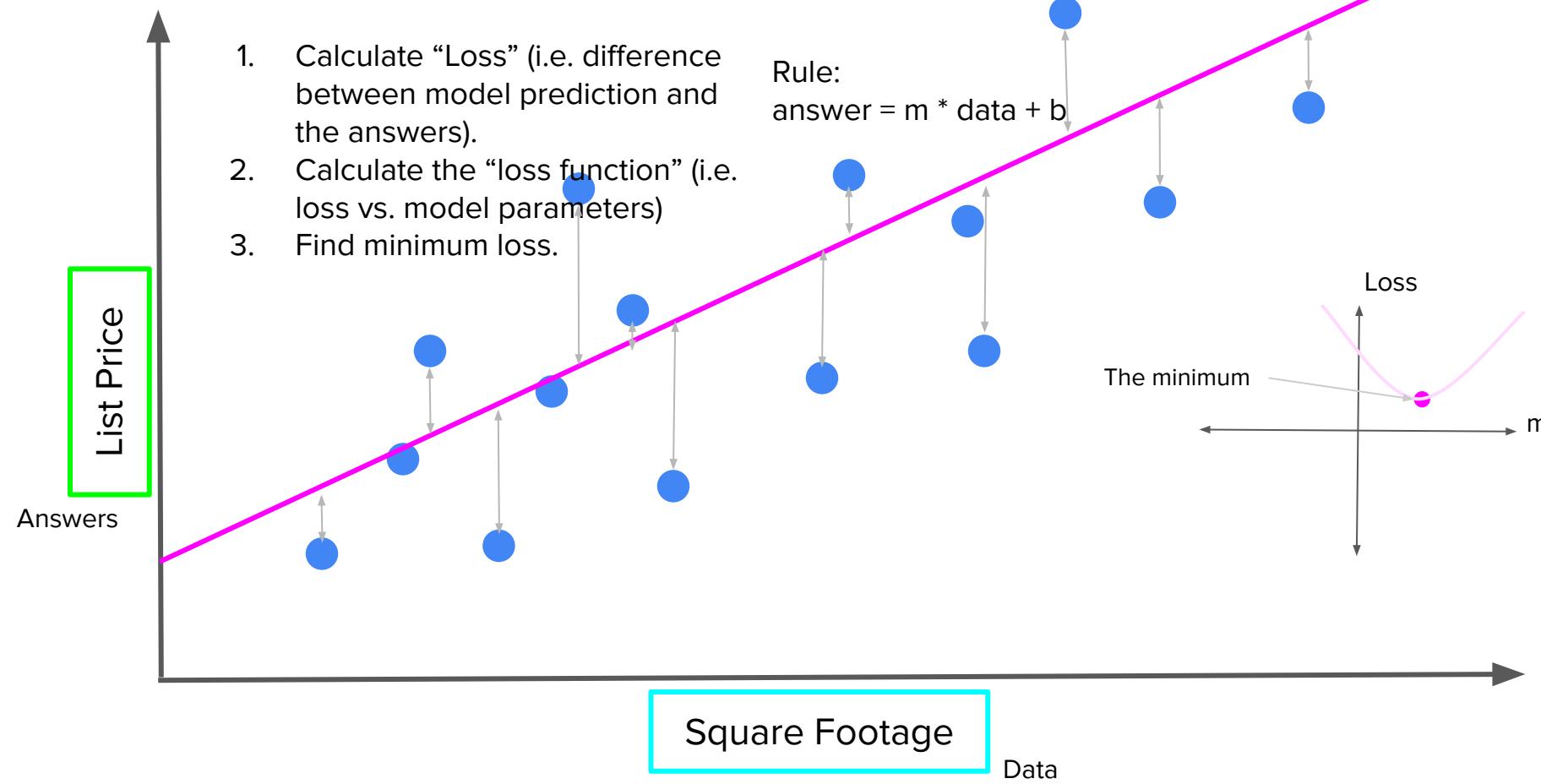
Rule:

$$\text{answer} = m * \text{data} + b$$



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.



The ML Recipe

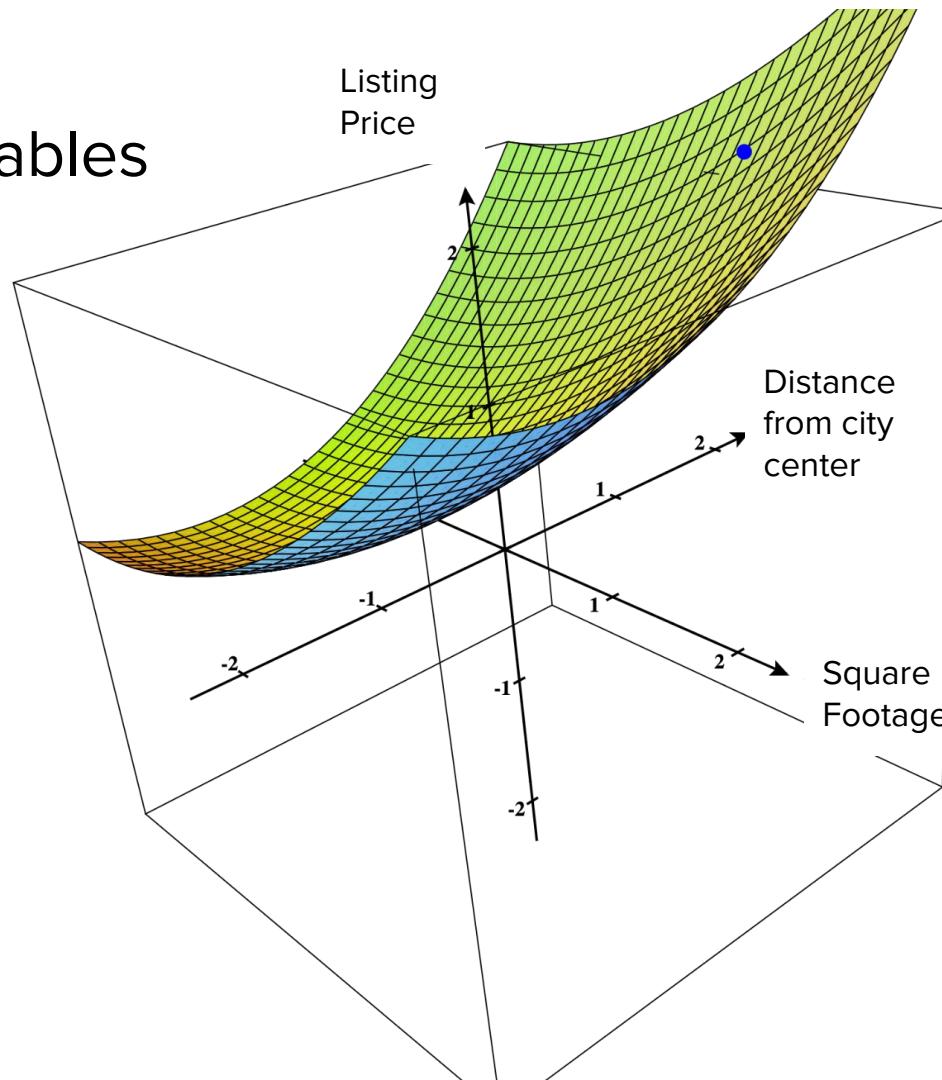
1. Think up some model
2. Feed **data** into the model and make predictions.
3. Calculate the loss between predictions and true values.
4. Determine the model parameters that produce the minimum loss.

The ML Recipe

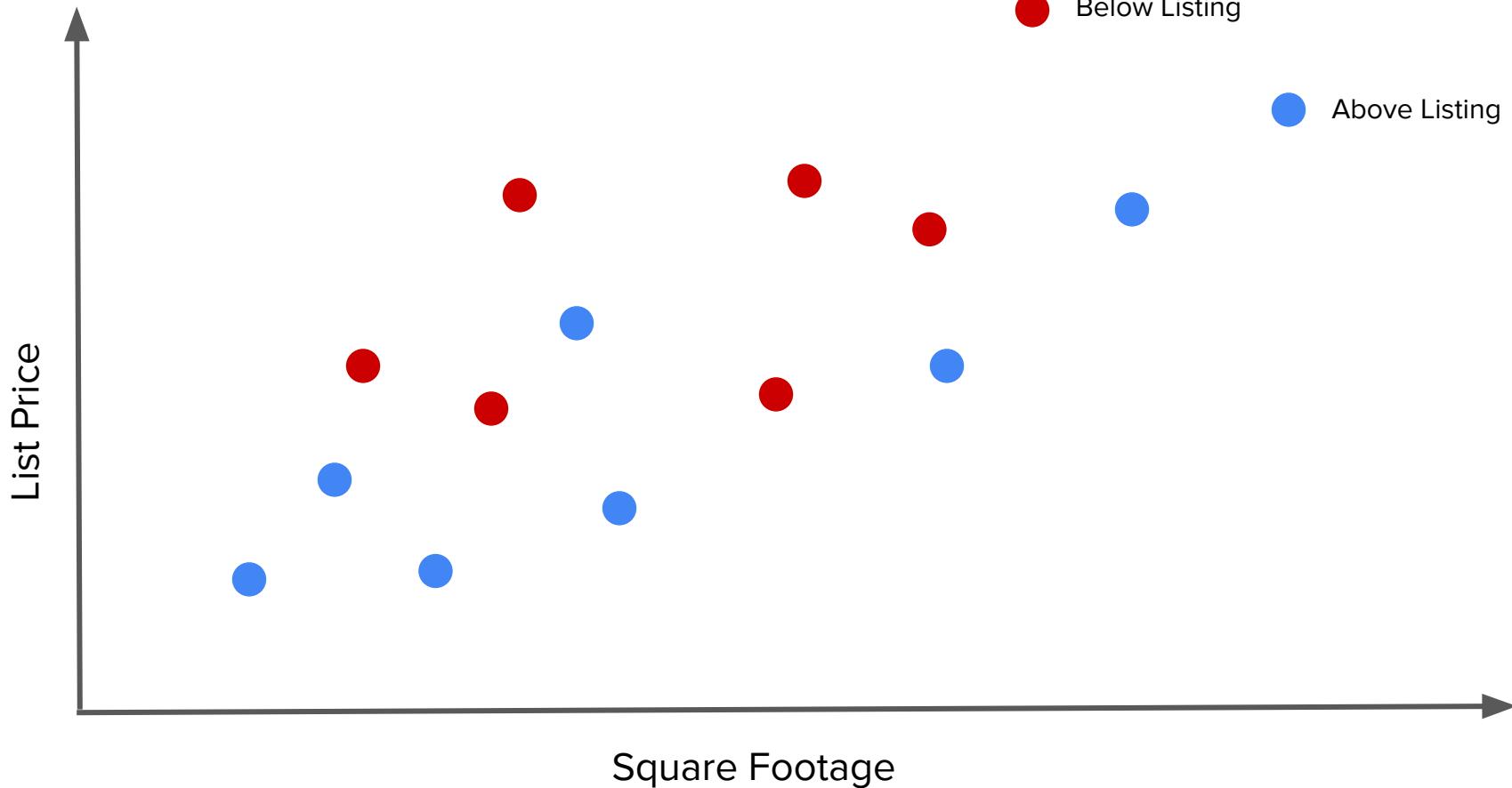
1. Think up some model
2. Feed **data** into the model and make predictions.
3. Calculate the loss between predictions and true values.
4. Determine the model parameters that produce the minimum loss.
 - a. Take the derivative of your loss function with respect to the model parameters.
 - b. Set it equal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

Multiple Variables

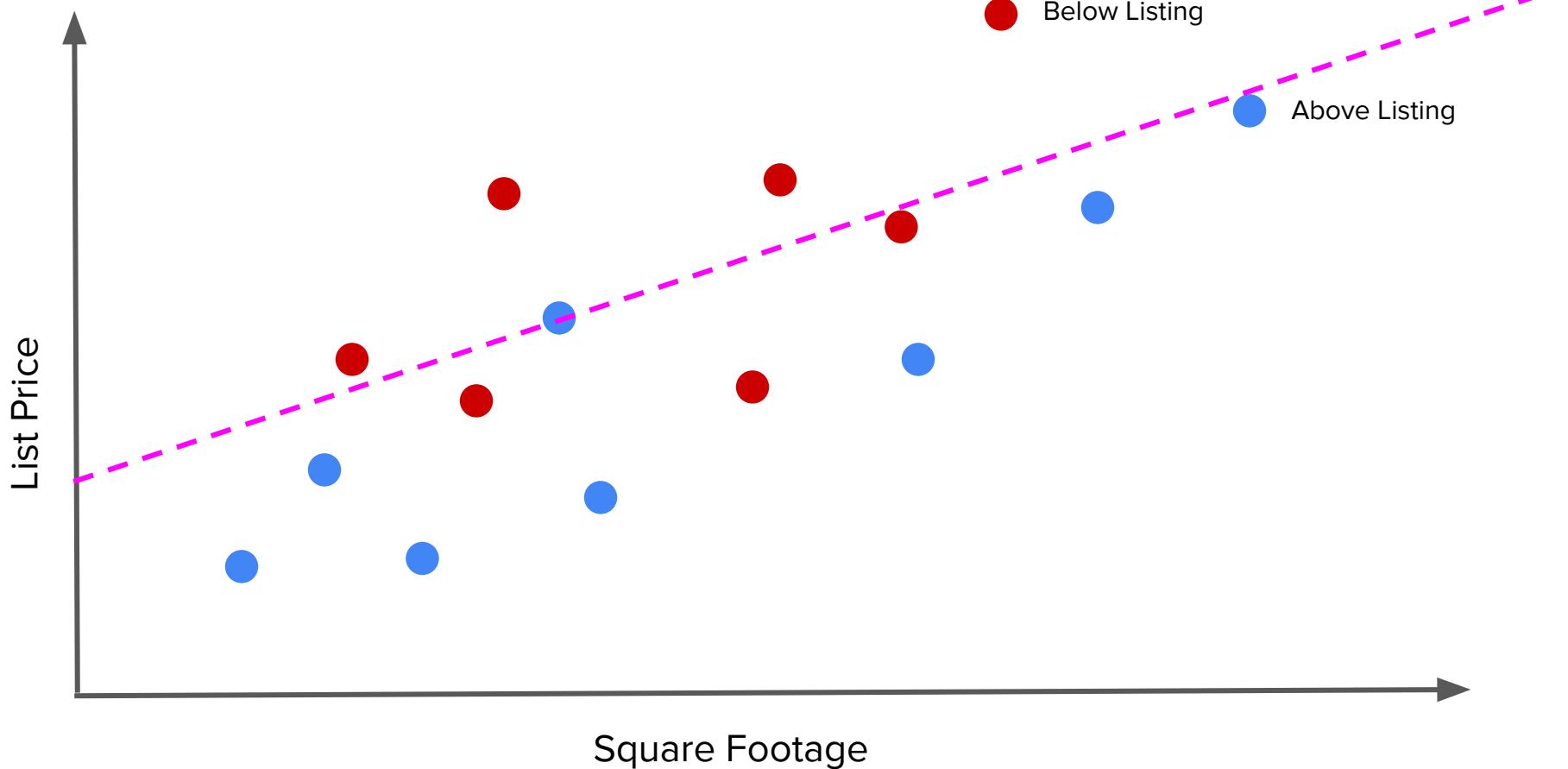
$$y = m_1 * x_1 + m_2 * x_2 + b$$



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

Data Input

(x's)

Square
Footage



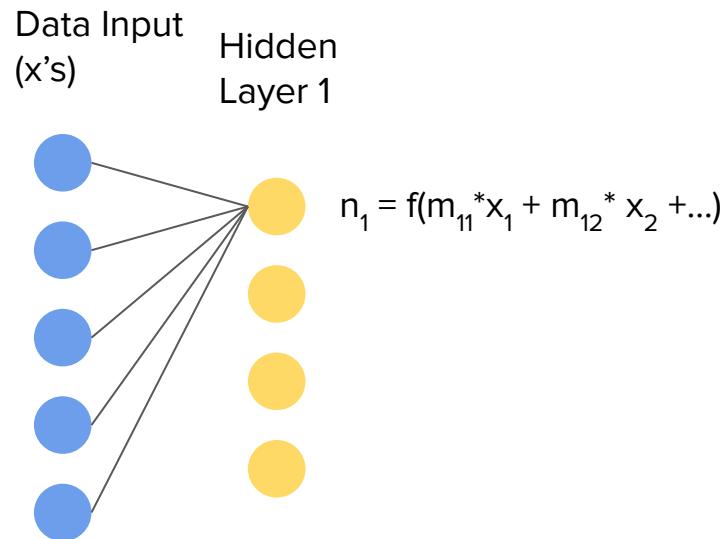
Distance



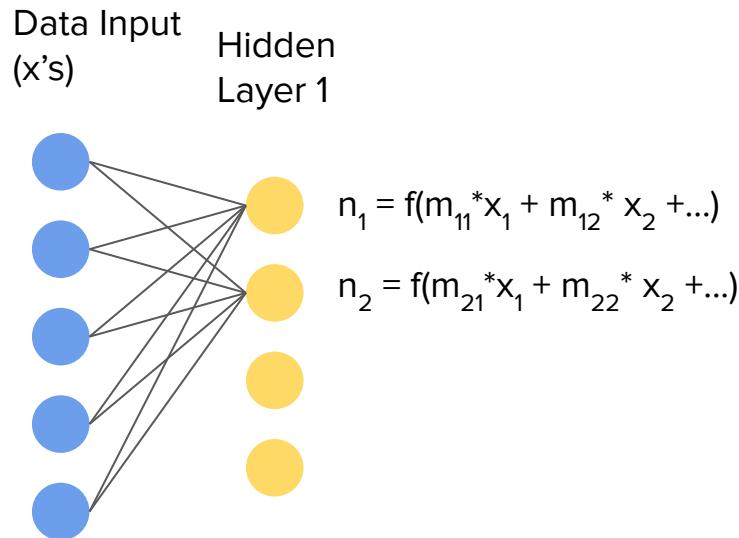
Bedrooms



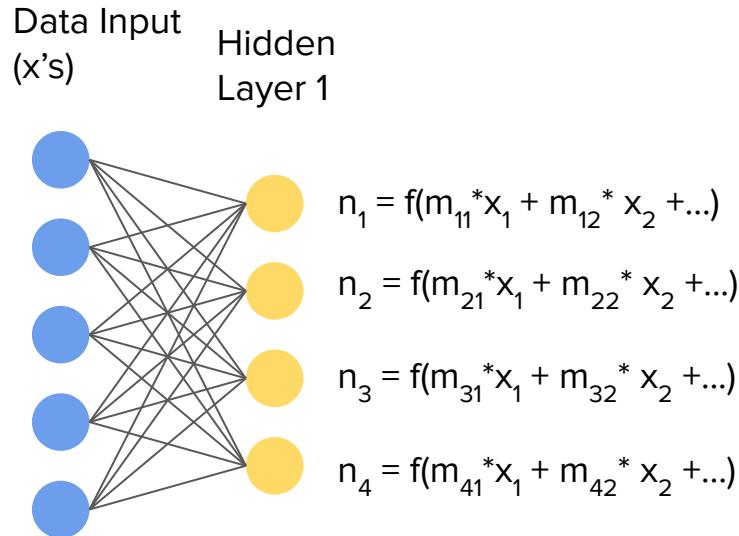
Deep Learning is Just a Fancy Model



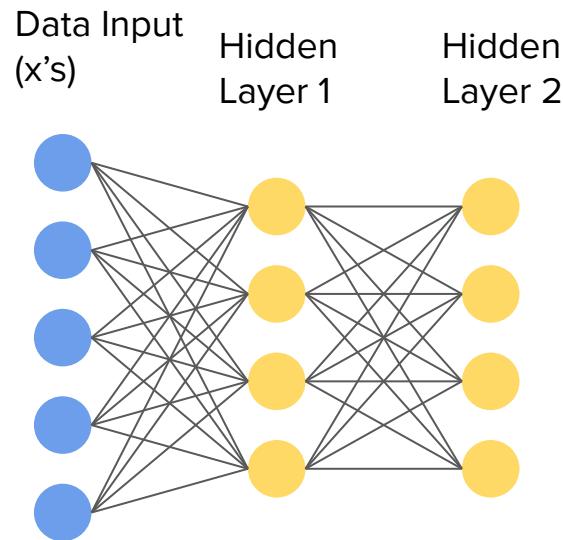
Deep Learning is Just a Fancy Model



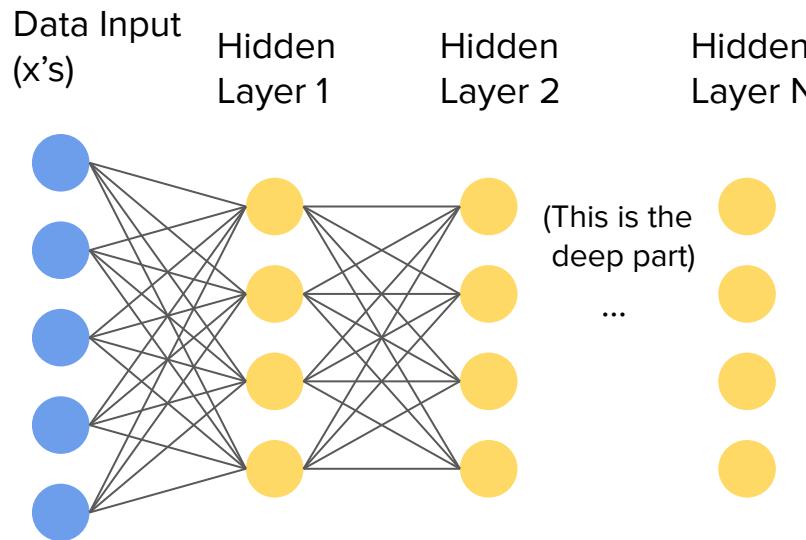
Deep Learning is Just a Fancy Model



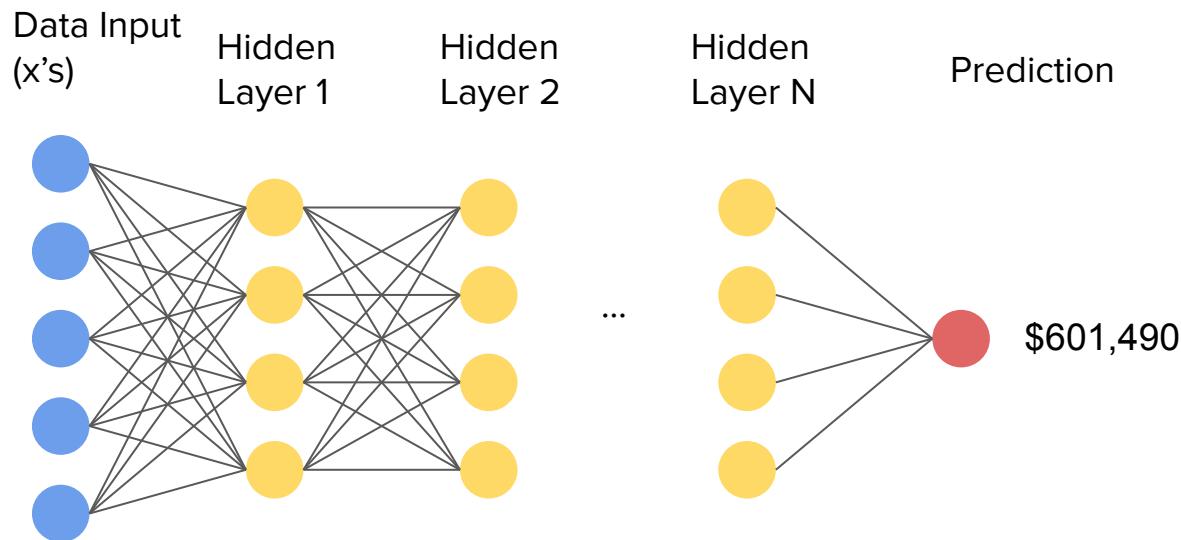
Deep Learning is Just a Fancy Model



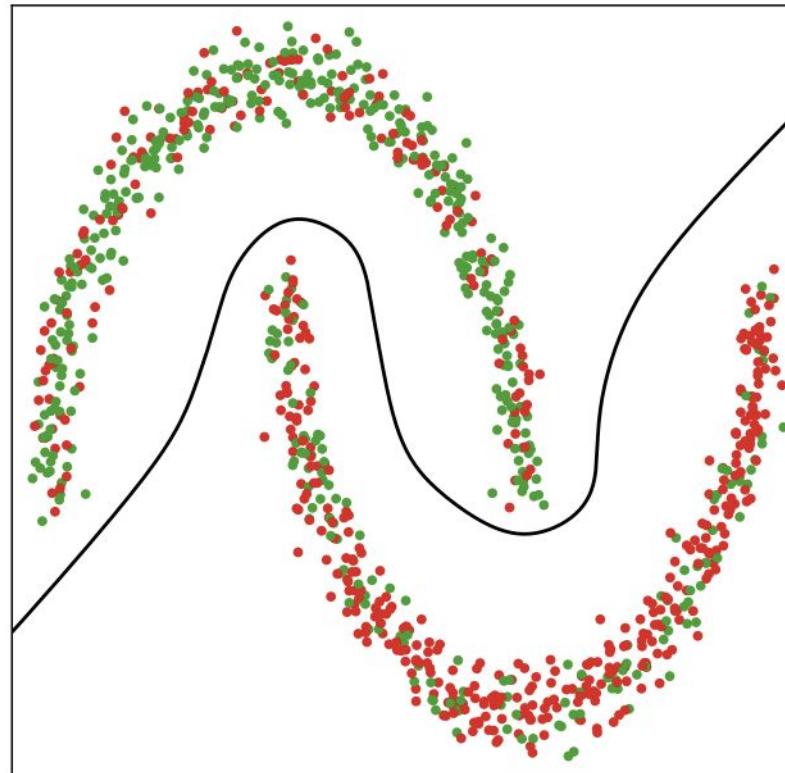
Deep Learning is Just a Fancy Model



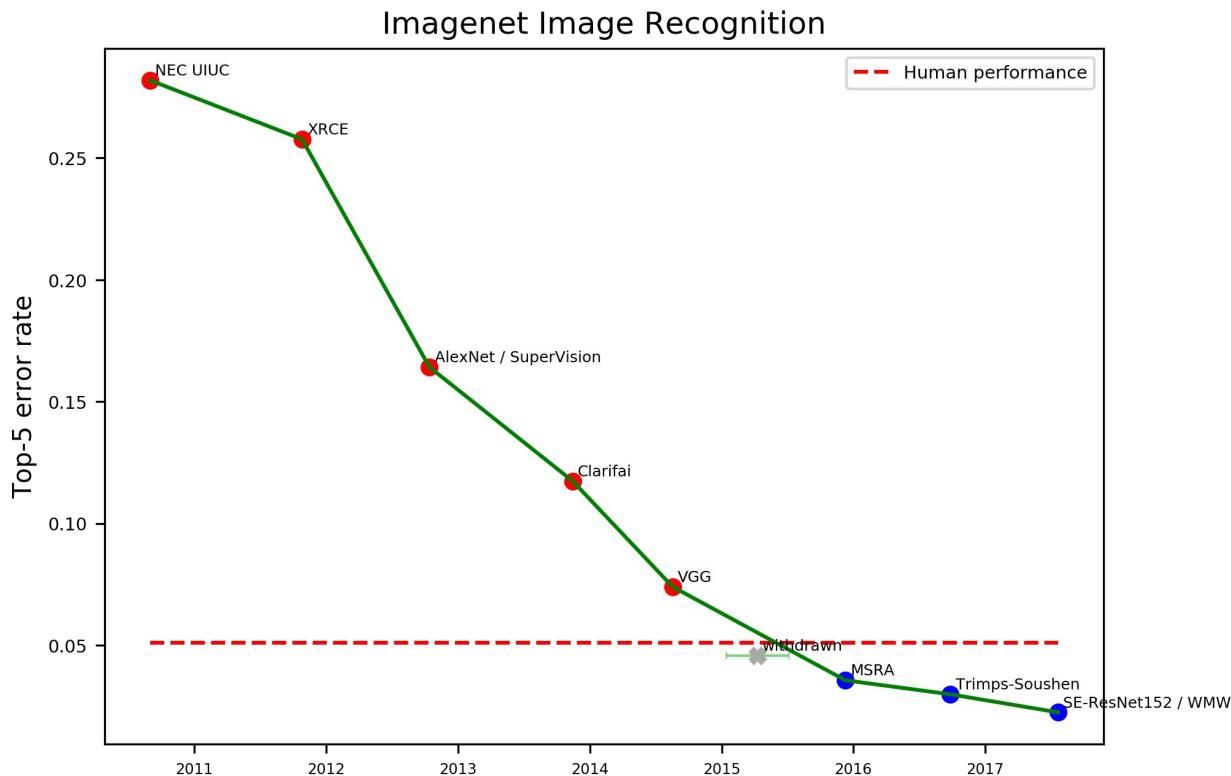
Deep Learning is Just a Fancy Model



Why Do Deep Learning? Handle Nonlinearities



Why Do Deep Learning? It Works.



Computer Setup

Computer Setup

1. Git
2. Installing Python
3. Virtual Environments

A Note on Operating Systems

- Python is notorious for presenting difficulties to Windows users. If you have a Windows computer, we highly recommend using the [Windows Subsystem for Linux](#) (WSL) to run Linux on your Windows computer.
- If you choose to use “regular” Windows, then please be warned that your professors will not be very helpful with debugging issues on your computer.
- If you are on a Mac, then we recommend installing packages with the Homebrew package manager whenever possible.
 - See <https://brew.sh> for installation instructions

Git and GitHub

- **Git** is a “version control system” that allows multiple people to collaborate and work on the same code.
- **GitHub** is a website that “hosts” git. In practice, GitHub is a website where you can upload and download code.
- Git and GitHub are wildly confusing but *essential* to modern software and ML.

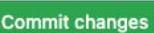
Setting Up Git

1. Your primary interaction with **git** will be as a program that you run from your terminal.
 - a. You should get comfortable using the terminal in this class!
 - b. Force yourself to use it instead of navigating via a window.
2. Go to <https://git-scm.com/downloads> and install Git
 - a. For Macs, we recommend installing with Homebrew.
3. If you type `which git` then you should see a path to the location where git was installed, such as `/opt/homebrew/bin/git`
4. Set your Git username and email by running the following in your terminal
 - a. `git config --global user.name "Your Name"`
 - i. Check if this was done correctly on a Mac or Linux by running the following, and you should see your user name print out:
 - ii. `git config --list | grep user.name`
 - b. `git config --global user.email "your@email.com"`

Setting Up GitHub

1. Go to <http://github.com> and create an account if you have not already.
2. Go to <https://docs.github.com/en/get-started/quickstart/create-a-repo> and follow the first 6 steps to create a repository.
 - a. A repository is like a folder that contains the files for a project.
3. Now, *clone* (aka download) your new repository to your laptop.
 - a. Go to the main page for your new repository and click the Green “Code” button.
 - b. Click HTTPS
 - c. Copy the url that’s underneath
 - d. From your terminal, run the following where you replace \$THE_URL with the URL that you copied.
 - i. `git clone $THE_URL`
 - e. The repo should now be on your laptop (type ‘ls’ to see if it’s in your current directory).

Pulling and Pushing

1. Through GitHub, you can modify your code online.
 - a. Go to your hello-world repo, and click on README.md
 - b. Above the file's contents, click 
 - c. On the **Edit file** tab, add some new text to the file.
 - d. Under **Commit changes** add a *commit message* where it says Update README.md in grey letters.
 - i. A “commit” is a logical change to your codebase. This could consist of multiple changes to multiple files, adding a file, removing a file, etc... The idea is that this is a collection of changes that are related to each other.
 - ii. Each commit gets a “commit message”. This is a small piece of text to explain what the commit does (e.g. “Allow users to login from Facebook”. “Remove old files”.)
 - e. Click the green  button to make your changes “official”.

Pulling and Pushing

1. You have now changed your code on GitHub.
2. On your laptop, navigate to your hello-world directory and run
 - a. `git pull`
3. This will *pull* the latest changes from GitHub to your laptop. You should see that your *local* copy of README.md has been updated.

Pulling and Pushing

1. You can now go through a similar process where you change your code on your laptop and **push** the changes to GitHub.
2. Open README.md in a text editor, modify it, and save it.
 - a. If you have no strong opinions on text editors, then we recommend using VSCode <https://code.visualstudio.com/>
 - b. Once you've installed VSCode, you can run the following to open README.md in VSCode
 - i. `code README.md`
3. We now must *commit* the changes to README.md
4. In your terminal, run
 - a. `git add README.md`
5. Then, run the following where you type your own commit message.
 - a. `git commit -m "Some commit message"`
6. You have now committed your code to your laptop. Run the following to push your changes from your laptop to GitHub
 - a. `git push origin main`

The FRE 7773 Repo

- This course has a GitHub repository at
<https://github.com/jacopotaagliabue/MLSys-NYU-2022>
- Clone this repo to your laptop.
- We will update this repo throughout the course. You can pull the latest changes to the repo onto your laptop by running
 - a. git pull

Computer Setup

1. ~~Git~~
2. Installing Python
3. Virtual Environments

Python is the
second-best language
for everything

The worst part about
Python is installing it
and its libraries

Installing Python

1. Go to the Week 1 folder of the class repo

<https://github.com/jacopotaagliabue/MLSys-NYU-2022>

2. Follow the steps for
 - a. Installing Python
 - b. Virtual Environments
 - c. Demo