# Intro to ML

- A training course for security researchers
- Alex Marks-Bluth

# Course Structure

**Problem Based Learning**

Each day (week) we will dive into a cybersecurity problem with a dataset and “learn from doing”

| | |
|---|---|
| **Day 0** | Self-study and preparations |
| **Day 1** | DNS tunneling |
| **Day 2** | Spam/not-spam |
| **Day 3** | L7DDoS - Volumetric Attack or Sale? |
| **Day 4** | Network Intrusion Detection |
| **Day 5** | TBD |

# Goals of the course
## “Think DS”

**Use ML as a tool**

- ML toolkit in Splunk
- Jupyter and standard python ML packages
- Know how to apply to cybersecurity datasets
- How to adapt existing models/pipelines to new data

**Familiarity with ML**

- Data science concepts
- Stages in researching and production
- Strategies in ML

**Know how to ask for help**

- How to get more specific help
  - What to ask
  - Who to ask
- How to work with DS consultants

# About Me

**Alex Marks-Bluth**

- Industry Data Scientist in cybersecurity industry > 10 years
  - Startups
  - Enterprise
  - Consulting
- Leads multidisciplinary teams of DS and security researchers in web security research in Akamai
- Work with large scale data and distributed systems, real time and offline data science systems


HOMEWORK
ASSIGNMENTS

# DNS tunneling

01

# DNS Tunneling

Housekeeping

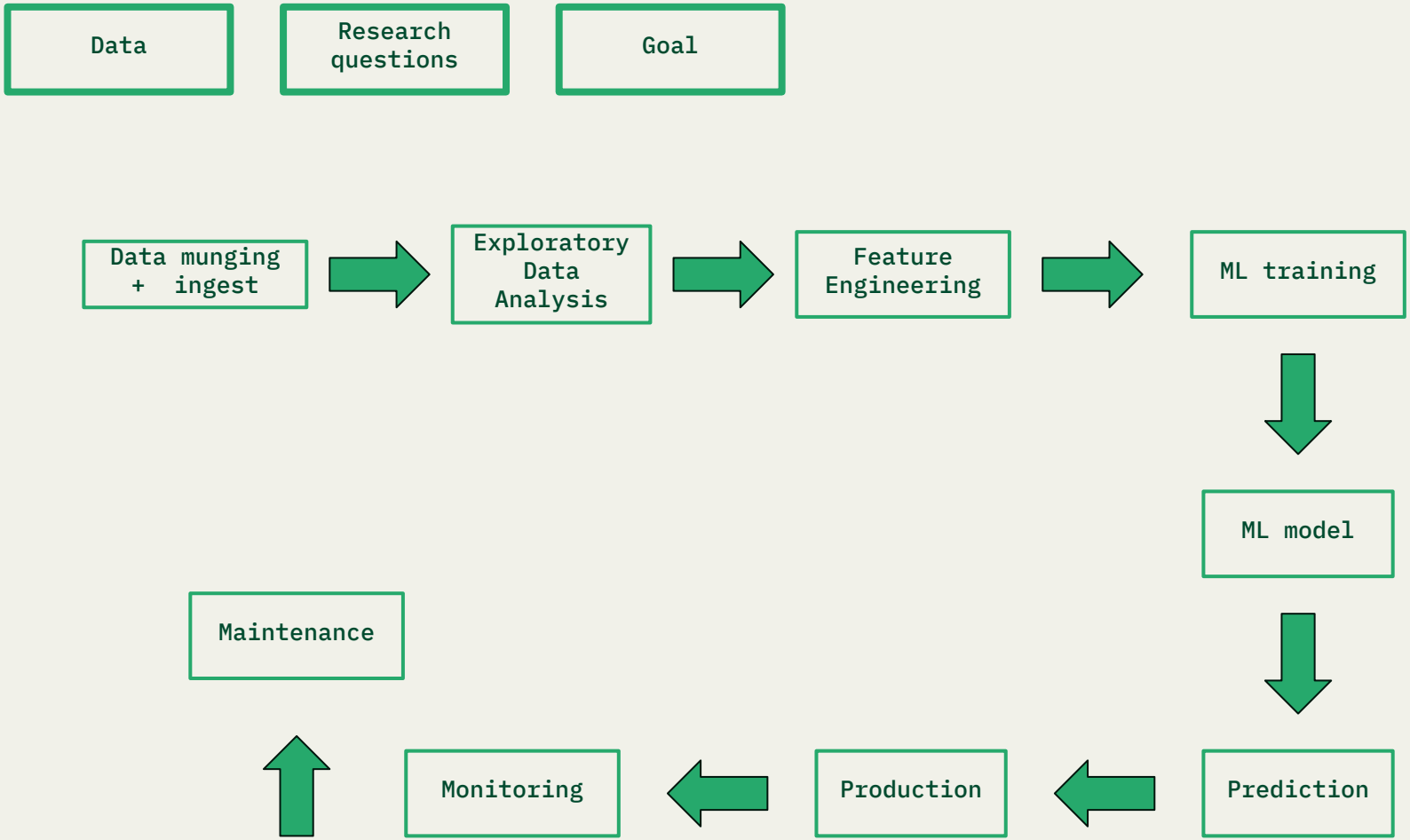
**Splunk**

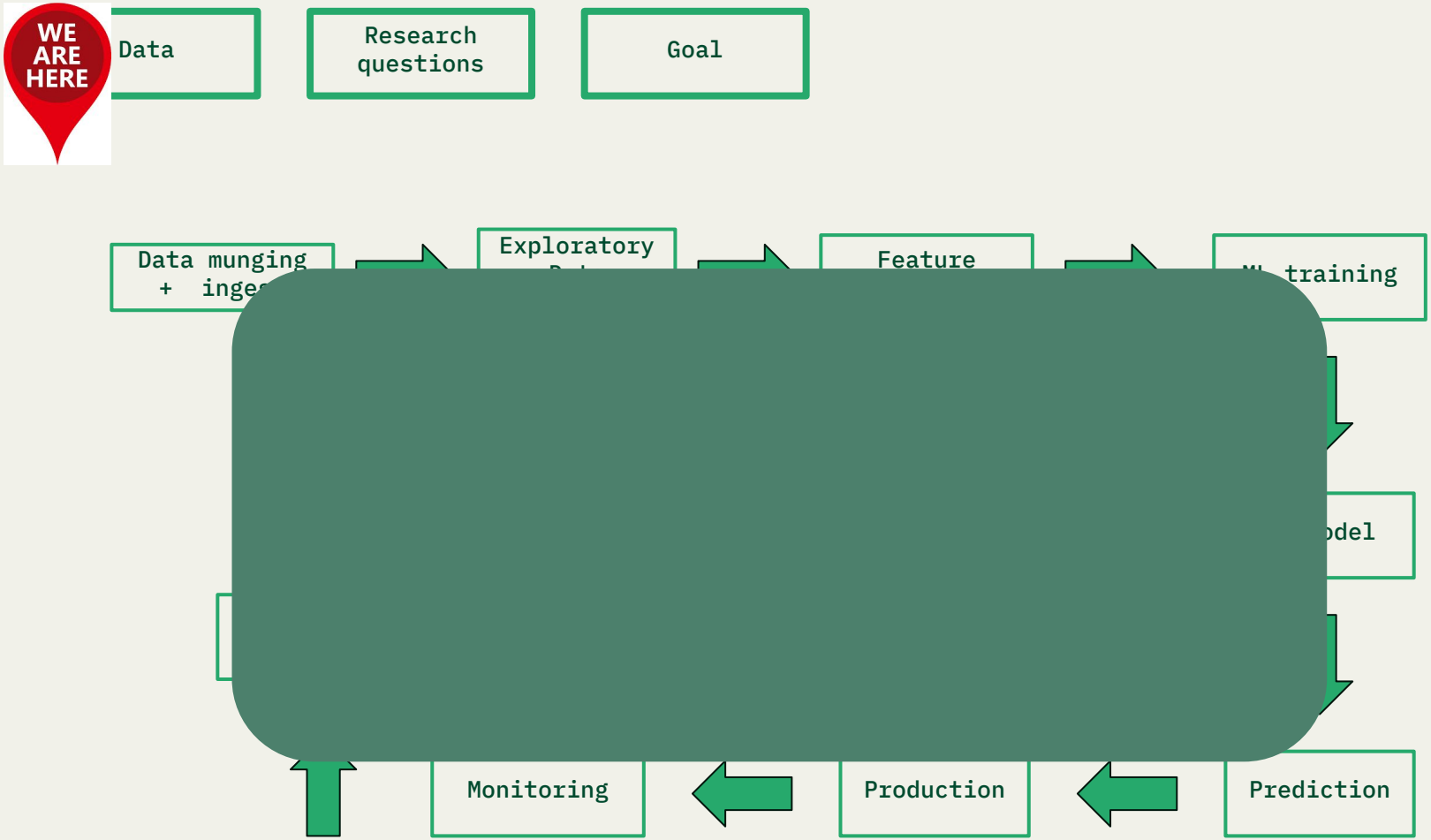
**Notebooks**

https://github.com/marksbluth/ML_course

**Data**

https://github.com/ggyggy666/DNS-Tunnel-Datasets/tree/main


Data
Research questions
Goal
Data munging + ingest
Exploratory Data Analysis
Feature Engineering
ML training
ML model
Prediction
Production
Monitoring
Maintenance


WE ARE HERE
Data
Research questions
Goal
Data munging + inges
Exploratory
Feature
training
odel
Monitoring
Production
Prediction

# Data Science in Industry

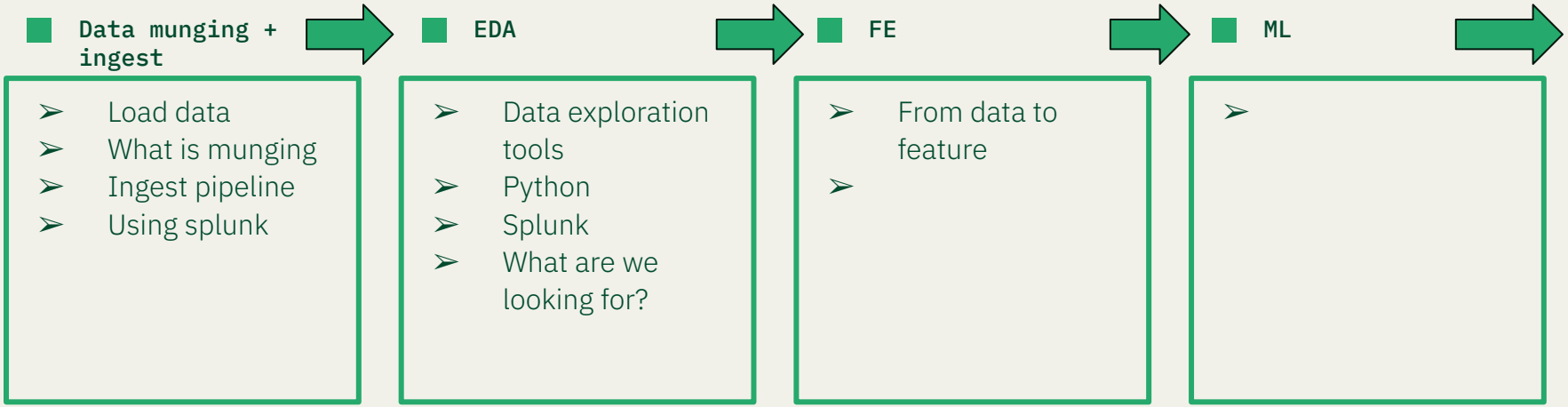
- `(What I do vs what my mother thinks I do)`

Hard parts are

- Framing a problem in a way that ML can solve
- Integrating ML solutions into a larger product
- Designing a ML system to work in production
- Testing a ML system
- Learning how to “smell” DS
- Just like the smell test in code.  Patterns, anti-patterns, assumptions

# e2e example on DNS data

## - some practical work :)



| | |
|---|---|
| **Getting Started** | 09:00 |
| **From data to insights** | 09:45 |
| **Coffee break** | 10:30 |
| **ML Inputs and outputs** | 10:45 |
| **e2e example on DNS data** | 11:30 |

| Getting Started | 09:00 |
| --- | --- |

**Tooling**

- databases
- notebooks
  - Jupyter, other
- experiment tracking + artifactory
  - MLFlow

| | |
|---|---|
| **Getting Started** | 09:00 |
| **From data to insights** | 09:45 |

- Data to dataframe
- Exploratory data analysis (EDA)
- how to frame a question in ML
  - intro to optimization functions
- How to structure a DS project

# From data to insights

- **Step-by-step:**

**Data to dataframe**

- Data = structured, unstructured, queries, ...
- Dataframe = rows and columns
  - Ordered
  - Distributed, single node
  - Pandas = standard, but pandas has some pointy edges
- “ML ready” dataframe = matrix of numbers
- (ML output = number(s) -> how to go back to relevant output?)

# From data to insights

## Step-by-step (continued)

**Exploratory data analysis (EDA)**

- Issues, errors, dirty
- Skew(s)
- Data sizing
- Data drift

**How to frame a question in ML**

- (intro to optimization functions)
- Classical optimization functions
- DL - what happens here?

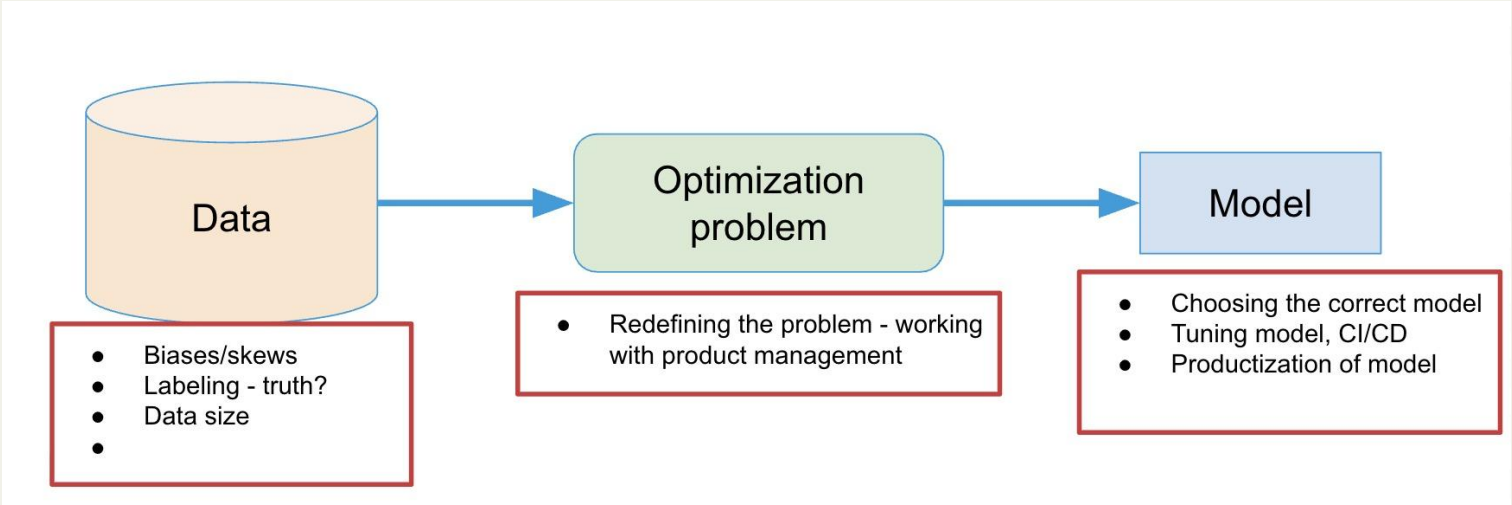
# ML as an optimization problem

## Linear Regression - how does it work

Linear regression is the simplest of “ML” models

- However is a good thought example of how more complicated models work
- Whiteboard example...
- What are some potential issues with linear regression?
    - Many dimensions
    - Skew
    - Large data
    - Entropy of data
    - How are these issues different for other models?

# ML as an optimization problem



# Types of optimizations

**Supervised learning**

= stick close to these labeled examples

**Unsupervised learning**

= find lowest energy

Semi-Supervised Learning

Reinforcement Learning

# Lots more models!

- Supervised learning (wikipedia)

  - Support-vector machines
  - Linear regression
  - Logistic regression
  - Naive Bayes
  - Linear discriminant analysis
  - Decision trees
  - K-nearest neighbor algorithm
  - Neural networks (Multilayer perceptron)
  - Similarity learning
  - Trees effectively allow for non-linear relationships while maintaining explainability:
  - Decision tree
  - Random forest
  - Gradient boosting, XGBoost, CatBoost

# Lots more models!

- Unsupervised Learning (wikipedia)

Clustering

- hierarchical clustering (eg Birch)
- K-means
- DBSCAN
- HDBscan

Anomaly detection

- Local Outlier Factor
- Isolation Forest
- Autoencoder

Latent variable models

| | |
|---|---|
| **Getting Started** | 09:00 |
| **From data to insights** | 09:45 |
| **Coffee break** | 10:30 |
| **ML Inputs and outputs** | 10:45 |
| **e2e example on DNS data** | 11:30 |

| | |
|---|---|
| **Getting Started** | 09:00 |
| **From data to insights** | 09:45 |
| **Coffee break** | 10:30 |
| **ML Inputs and outputs** | 10:45 |

**Taxonomy**:
types of data
- structured, unstructured, balanced, skewed

types of models
- supervised, unsupervised, semi-supervised, self-supervised, reinforcement

| | |
|---|---|
| **Getting Started** | 09:00 |
| **From data to insights** | 09:45 |
| **Coffee break** | 10:30 |
| **ML Inputs and outputs** | 10:45 |
| **e2e example on DNS data** | 11:30 |

**Tools**
sklearn package, Splunk Machine Learning Toolkit

**Discussion** - How to use ML without being an expert
Examples - linear regression vs LLMs
Building blocks - python vs machine code - levels of abstraction in DS

# Checks for EDA + FE processes

- Assumptions
- Repeatable
- Scaleable
- Testable

# Actual short courses that I recommend!

https://developers.google.com/machine-learning