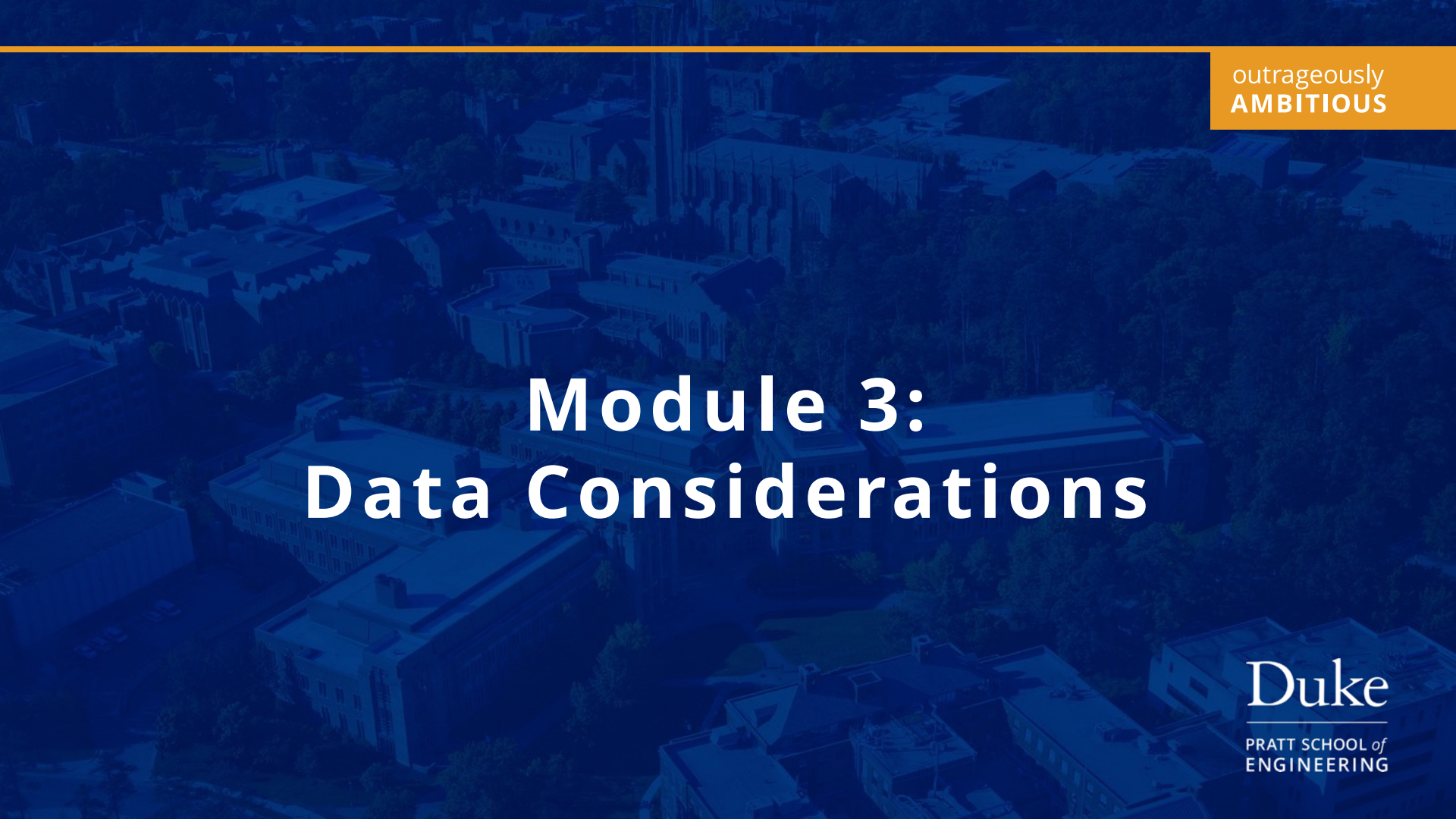
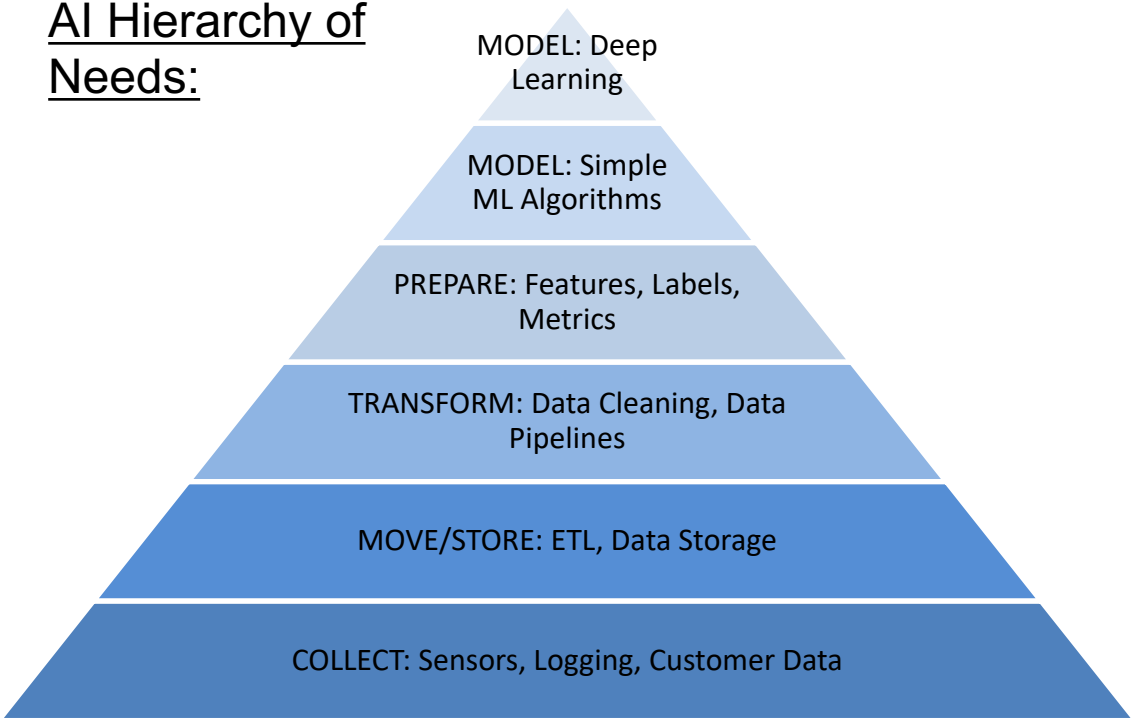

outrageously
AMBITIOUS
Module 3:
Data Considerations
Duke
PRATT SCHOOL of
ENGINEERING

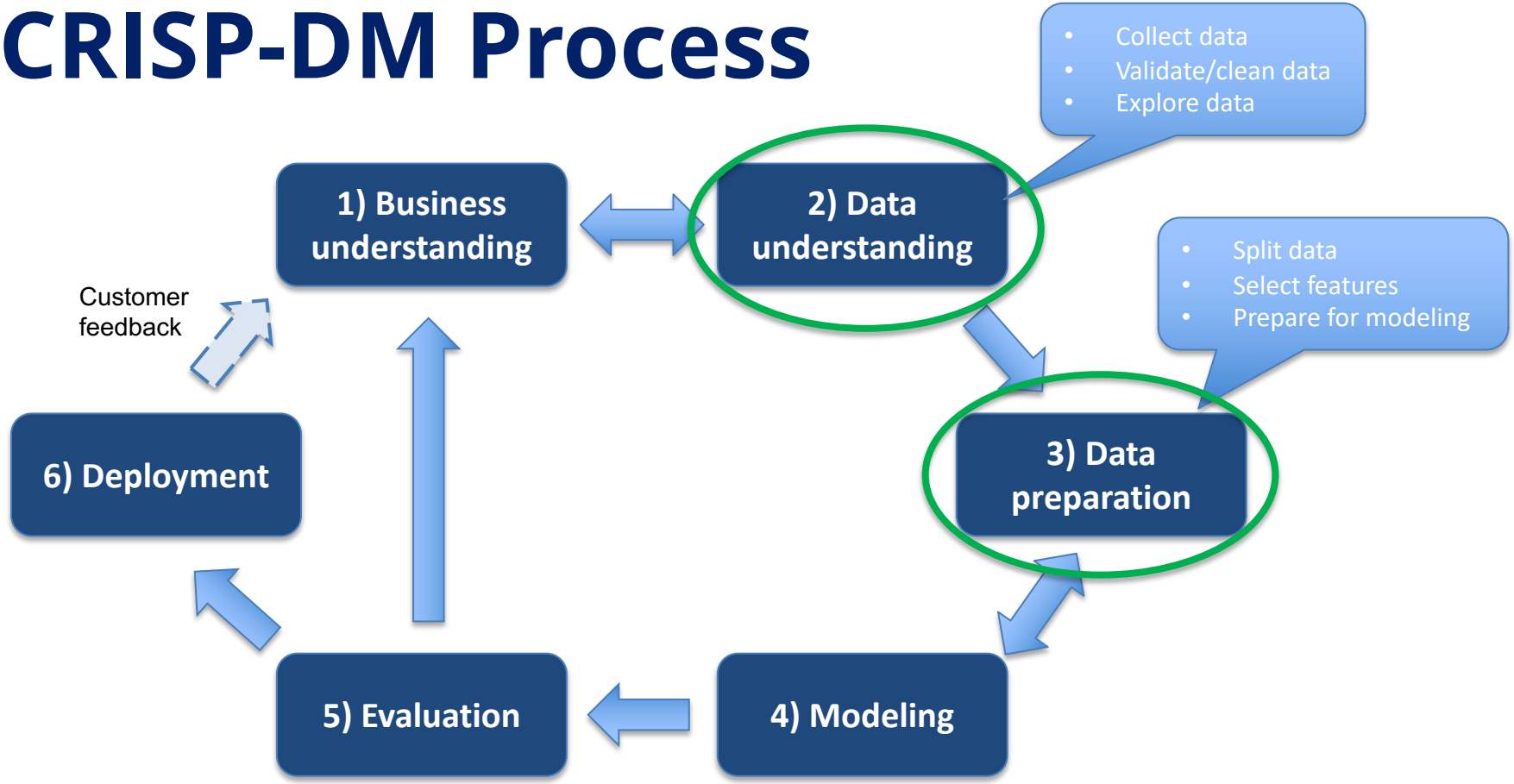
Importance of data

AI Hierarchy of Needs:



Inspired by Monica Rogati: https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007

CRISP-DM Process



Module 3 Objectives:

At the conclusion of this module, you should be able to:

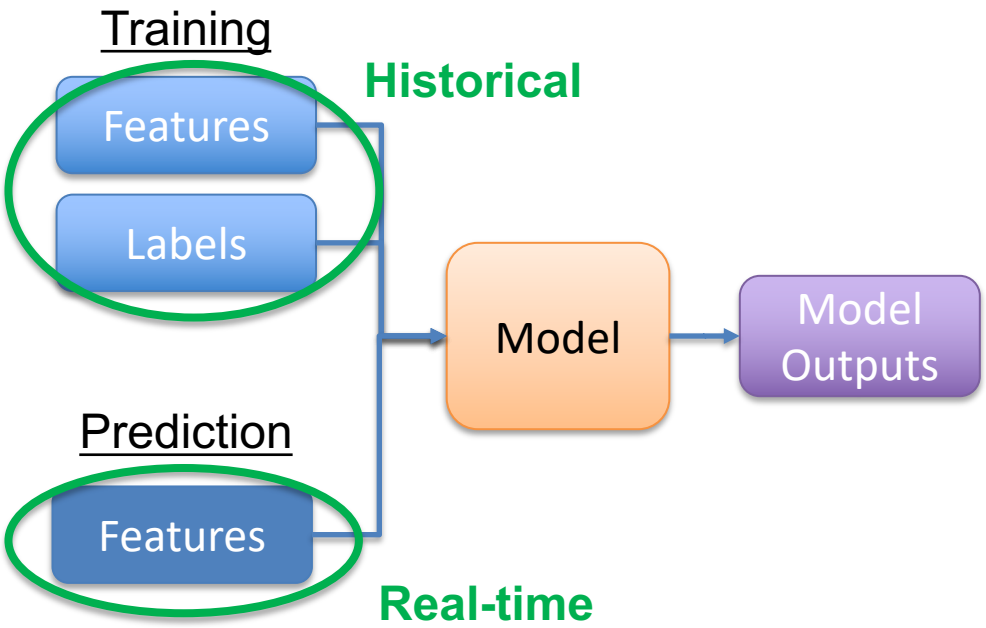
1) Evaluate data needs and sources of data
2) Identify strategies to collect data to support modeling
3) Explain the steps in building a data pipeline

outrageously
AMBITIOUS

Data Needs

Duke
PRATT SCHOOL of ENGINEERING

What data do you need?



Training Data: Features

- **How to identify features?**
  - Subject matter experts
  - Customers
  - Temporal and geospatial characteristics
- **How many features?**
  - Start with a small set and establish a baseline
  - Add more and evaluate impact
  - Try everything logical – missing a key feature is much worse than having an extra

Training Data: Labels

- For supervised learning, we need labels/targets of what we are trying to predict
- We may be able to source these labels or may have to create them
- Our definition of the problem determines the form of the labels

How much data?

- Generally, the more the better
- Orders of magnitude more observations than number of features or labels
- Factors that influence data requirements:
  - Number of features
  - Complexity of the feature-target relationships
  - Data quality – missing and noisy data
  - Desired model performance

How much data?

Dataset	Size (observations)
Iris flower dataset	150
CheXpert chest xrays	224,000
ImageNet dataset	14 million
Google Gmail SmartReply	238 million
Google Translate	trillions


outrageously
AMBITIOUS
Data Collection
Duke
PRATT SCHOOL of
ENGINEERING

Sources of data

- **Internal data**
  - Log files / user data
  - Internal operations & machinery
- **Customer data**
  - Sensors
  - Operational systems & hardware
  - Web data – forms, votes, ratings
- **External data**
  - Weather, demographics, social media, etc.

Best practices in collecting data

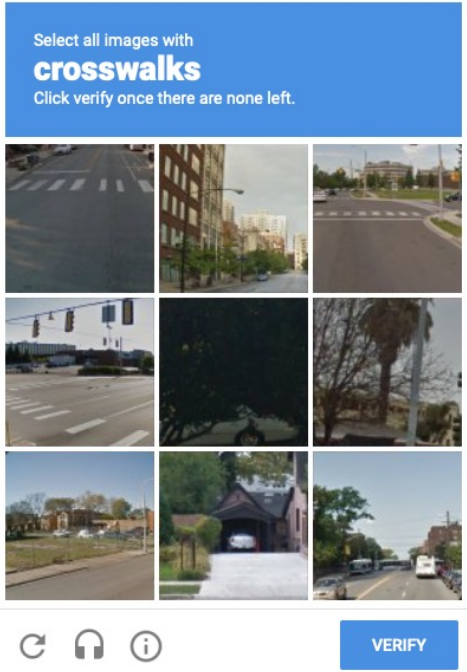
- **Collect data intentionally**
  - Only what you need
  - Beware of bias
  - Representative data
- **Update data as needed**
  - Environment changes
  - Re-train model periodically
- **Document**
  - Sources & metadata

Data labeling

- Sometimes label data can be collected, other times it must be created
- Possible sources of labels/targets
  - Log files
  - Customer records
  - Sensor readings
  - User input
- Methods of creating labels
  - Manual creation
  - Commercial data labeling services

Collecting user data

- Many options for collecting user data:
  - Forms
  - User behavior
  - Votes, rankings
- Ideally the data collection should:
  - Be an integrated part of the user's workflow
  - Provide the user some benefit
- Creative examples:
  - Google -> CAPTCHA
  - StitchFix -> Keeping vs. returning



Flywheel effect

- Users generate data through interaction with an AI-enabled system
- Data can be used to strengthen the AI and open up further opportunities
- E.g. Amazon:
  - Searches/purchases -> Reorder listings
  - Purchases/ratings -> Personalized recommendations
  - Purchase records -> “Shoppers also bought” (Co-occurrence matrix)

Cold start problem

- If we are relying on user-supplied data for our model, we may initially not have enough to build a quality model
- This is particularly challenging with recommendation systems, where we face it with every new user
- We may consider starting with heuristics-based approaches, or adding a calibration step to gather data


outrageously
AMBITIOUS
Data Governance & Access
Duke
PRATT SCHOOL of
ENGINEERING

Dealing with data silos

- One of the key barriers to broader use of ML is that data is often siloed and inaccessible
  - Each department collects and manages its own data
  - Enterprise systems store data in different places using different schemas
- For a company just starting to use ML, it is wise to focus first on breaking down the silos

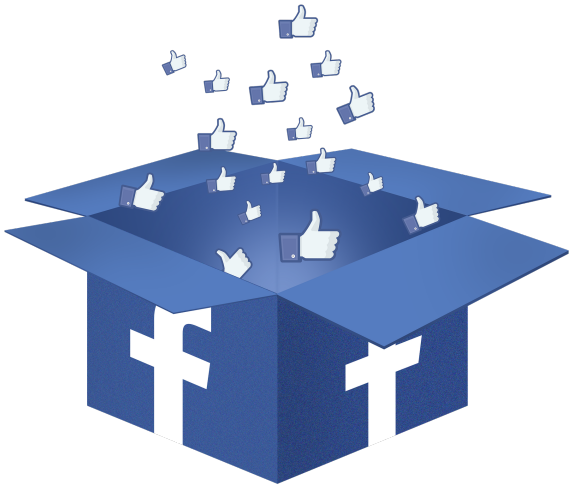
Dealing with data silos

Breaking down data silos requires:

1. **Cultural change**
   - Executive sponsor
   - Education and incentives
2. **Technology**
   - Centralized data warehouse
   - Data querying tools
3. **Data stewardship and access**
   - Responsibility for data stewardship
   - Make it easy to discover and access

Facebook: democratizing data

- From 2007 to 2010 the data collected by Facebook was exploding, and so were requests for it (10k/day)
- Transitioned to a Hadoop cluster which made access difficult
- Developed Hive, which allows users to query data using SQL
- Moved to self-service model and made training available to all employees
- Hackathons provided employees opportunities to find creative uses of data




outrageously
AMBITIOUS
Data Cleaning
Duke
PRATT SCHOOL of
ENGINEERING

Data cleaning

- Messy data can prevent the development of quality models
- Data can have several issues:
  - Missing data
  - Anomalous data
  - Incorrectly mapped data

Missing data

- Data can be missing for a number of reasons:
  - Users did not provide (web form)
  - Mistakes in data entry / mapping
  - Issues with sensors (power, comms, failures)

Types of missing data

	Missing Completely at Random	Missing at Random	Missing Not at Random
Description	No pattern in missing data or association to values of other attributes	Probability of missing-ness relates to another feature of the data	Probability of missing-ness relates to values of the feature itself
Example	Power outages of sensors	Males are less likely to answer survey questions about depression	Purchased item ratings skew towards people who hated the product
Potential for Bias	Low	High	High

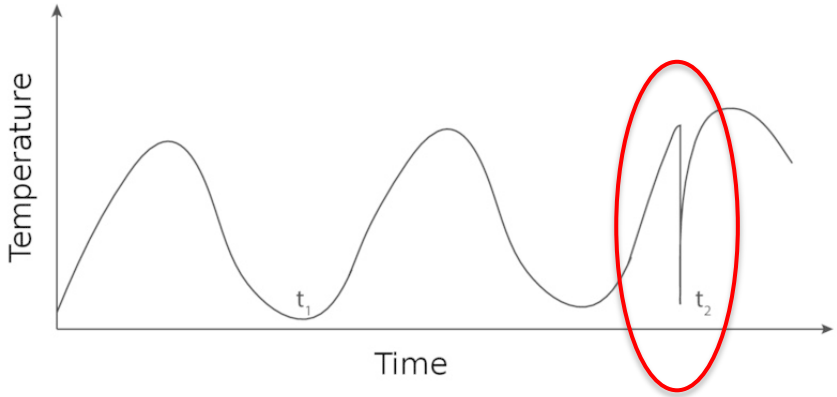
Dealing with missing data

There are multiple options for how to deal with missing data:

1. **Drop it:** remove rows or columns
2. **Flag it:** treat it as a special value of feature
3. **Replace with mean value/median**
4. **Backfill or forward-fill:** from previous or future values
5. **Infer it:** use simple model to predict it

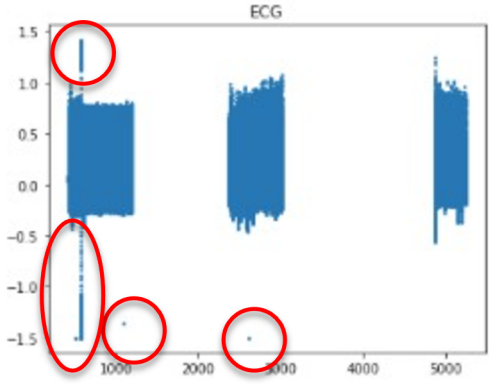
Outliers

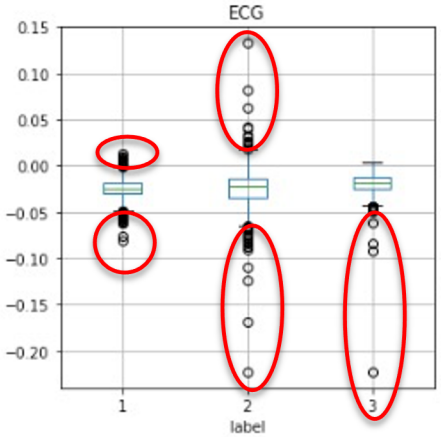
- Points which fall far from the rest, either in a feature value or the target value
- Outliers can overly influence your model
- Contextual outliers: not all outliers are extreme values



Detecting outliers

- We can use visualizations and statistical methods to identify outliers





Dealing with outliers

- Be careful not to automatically remove or adjust outliers
- Removing outliers may inhibit the model performance on extreme events
- First try to understand root cause – are they real or anomalous data?
- If anomalous, either remove or adjust to more likely value (fill with mean or back/forward fill)


outrageously
AMBITIOUS
Preparing Data for Modeling
Duke
PRATT SCHOOL of
ENGINEERING

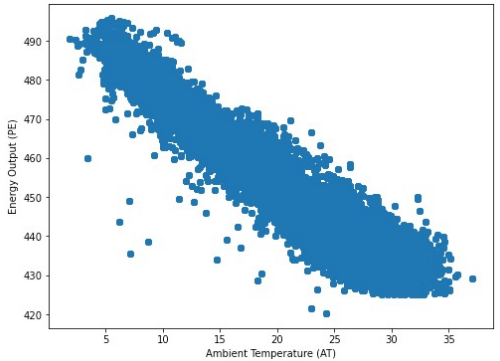
Preparing data for modeling

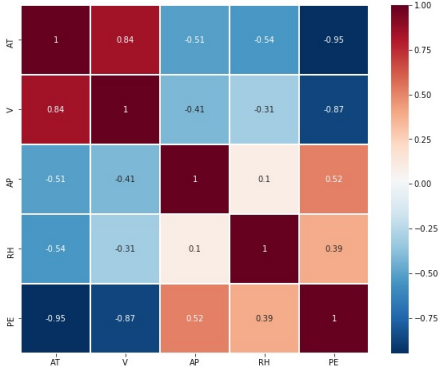
- Generally, collected raw data is not in a suitable form for training models
- Typically need to perform:
  - Data cleaning
  - Exploratory data analysis
  - Feature engineering & selection
  - Preparation for modeling

Exploratory data analysis (EDA)

- EDA helps us catch issues in our data and understand it better
- Involves both statistics and visualization
- We generally seek to understand distributions and relationships between our features with each other and the target

	AT	V	AP	RH	PE
count	47840.000000	47840.000000	47840.000000	47840.000000	47840.000000
mean	19.651231	54.305804	1013.259078	73.308978	454.365009
std	7.452162	12.707362	5.938535	14.599658	17.066281
min	1.810000	25.360000	992.890000	25.560000	420.260000
25%	13.510000	41.740000	1009.100000	63.327500	439.750000
50%	20.345000	52.080000	1012.940000	74.975000	451.550000
75%	25.720000	66.540000	1017.260000	84.830000	468.430000
max	37.110000	81.560000	1033.300000	100.160000	495.760000





Feature engineering

- We represent our data as a collection of features, or attributes
- Getting the right set of features is critical to modeling success
  - We can use sub-optimal models and still get good results
  - If we use the wrong features, regardless of model we will not achieve good results
- Features may be natural attributes of data or we may need to create them (e.g. text)

Example 1

Predicting daily energy consumption of an office building based on weather

- Which weather parameters?
  - Temperature, humidity, cloudiness, wind speed, etc?
- At what level of granularity?
  - Daily average? Business hours average? Hourly?
- Interactions between parameters
  - E.g. minutes of sunshine and season of year

Example 2

Predicting hourly demand for bikes for a bikeshare network

- Which weather parameters?
  - Temperature, humidity, cloudiness, wind speed, etc?
- How do we represent time?
  - Hour of day? Day of week?
  - Business day vs weekend? Holiday?
  - Month / season of year?

Feature selection

- Reducing feature set has many benefits:
  - Reduce complexity / risk of overfitting
  - Reduces training time
  - Improves interpretability
- However, missing a feature can be disastrous for modeling
- We perform **feature selection** to downsize our possible features to an optimal set

Feature selection methods

	Filter Methods	Wrapper Methods	Embedded Methods
Description	• Statistical tests which rely on characteristics of the data only	• Train model on subsets of features	• Extracts features which contribute most to training of a model
Pros & Cons	• Computationally inexpensive • Often used before modeling to remove irrelevant features	• Computationally very expensive • Often unfeasible for real-world modeling	• Leverage model training with minimal additional computation

Transform data for modeling

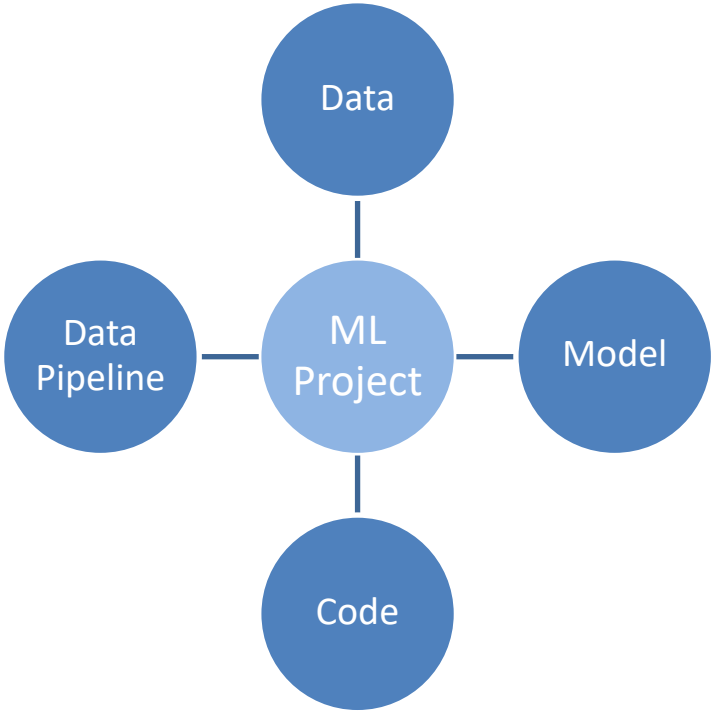
- Final step is to prepare data in a format to be ingested to a model
- This often involves:
  - Scaling data to put values of different features into the same order of magnitude
  - Encoding categorical variables – converting string variables into numerical codes


outrageously
AMBITIOUS
Reproducibility & Versioning
Duke
PRATT SCHOOL of
ENGINEERING

Reproducibility

- Ability to reproduce results is a major issue in ML projects
- Reproducibility is important because:
  - Helps debug future issues
  - Employees can leave
  - Handoffs between teams
  - Peer reviews establish credibility
- Reproducibility best practices
  - Documentation – functionality, dependencies
  - Data lineage
  - Model, code & data versioning

Versioning

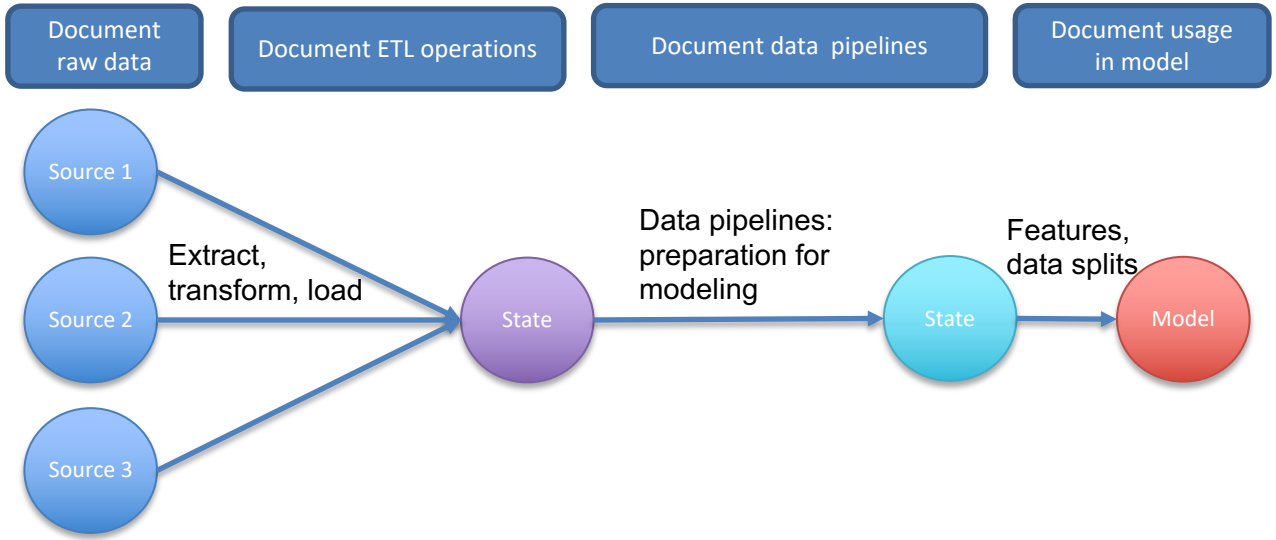


Data lineage

- Data lineage involves tracking of data from source to consumption – how it was transformed and where it moved
- Benefits:
  - Enables debugging
  - Simplifies data migrations
  - Inspires trust in data
  - Meets compliance requirements
- Options for data lineage
  - Commercial data lineage systems
  - Spreadsheet / graph software

Data lineage

- Common to visualize data lineage as a set of maps at different levels
- Record information on sources, characteristics, relationships, transformations and locations



Model versioning

- Along with data lineage and code versioning, model versioning is critical
- Benefits:
  - Track modeling experiments: code, data, model config, results
  - Track production model and revert if necessary
  - Run champion/challenger model tests
- Options for model versioning
  - Commercial model versioning (Weights & Biases)
  - ML platform-as-a-service (H2O)
  - Manual (log) or open source (MLFlow, DVC)


outrageously
AMBITIOUS
Wrap-up
Duke
PRATT SCHOOL of
ENGINEERING

Wrap Up

- Collecting sufficient data, with good quality and the right features, is the most important factor in successful ML
- Data should be collected intentionally and updated as things change
- Prior to investing in ML organizations should focus on clean & accessible data
- Collaboration and reproducibility tools and methods are critical to track progress through experimentation