Week 3: Data Definition

In the previous image, there is a very simple example of ambiguity in annotation jobs, this is called annotation inconsistency.
This week, we will dive into defining the data and best practices to label and organize the data well.

It is important to **standardize annotation** To avoid ambiguity.

Questions to ask for data definition:

- What is the input x?
    - Lightning? Contrast? Resolution?
    - What features need to be included?
- What is the target label y?
    - How can we ensure labelers give consistent labels?

Major types of data problems:

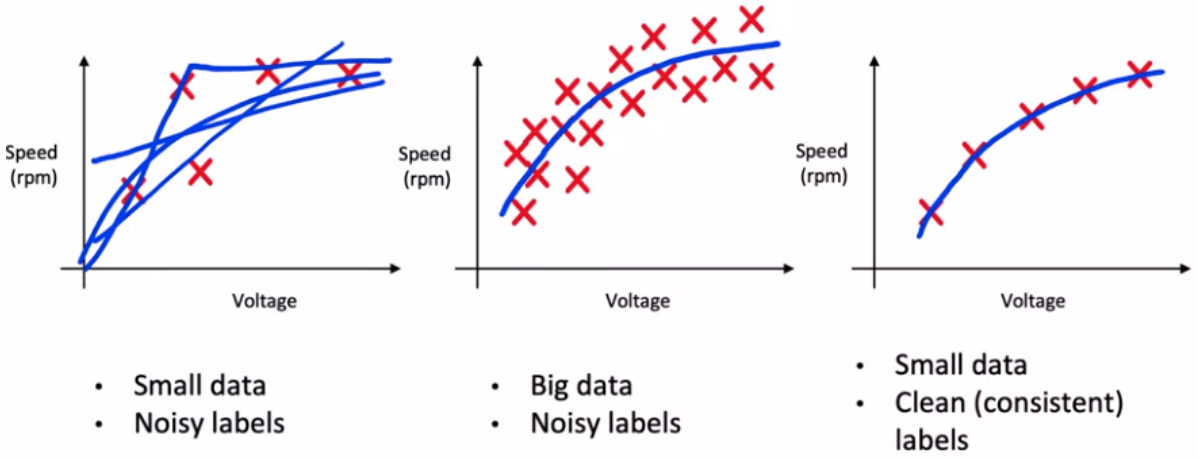
- Unstructured data: Here we can use humans to label data, Data augmentation, or synthesize new data.
    - Small data: <= 10.000 examples, Having clean labels are critical!
    - Big data: > 10.000 examples, Emphasis on data process.
- Structured data: Harder to synthesize or obtain more data, human labeling may not be possible (with some exceptions)
    - Small data: <= 10.000 examples,
    - Big data: > 10.000 examples, Emphasis on data process.

	Unstructured	Structured
Small data	Manufacturing visual inspection from 100 training examples	Housing price prediction based on square footage, etc. from 50 training examples
Big data	Speech recognition from 50 million training examples	Online shopping recommendations for 1 million users

≤ 10,000

> 10,000

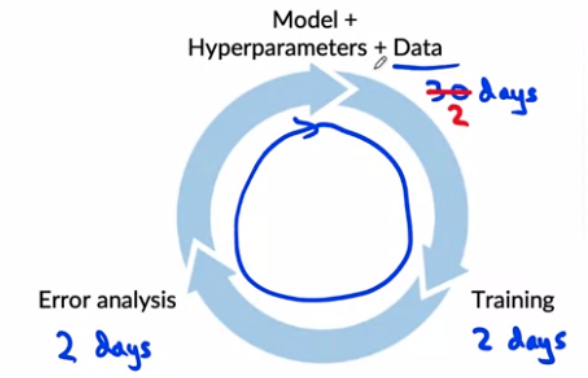
Small dataset and data consistency:



Measuring HLP is important, but inconsistently labeling instruction will low HLP.

How long should you spend obtaining data?

- Getting into this iteration loop as quickly as possible



- Instead of asking: How long it would take to obtain m examples? Ask: How much data can we obtain in K days.
- Exception: If you have worked on the problem before and from experience, you know you need m examples.

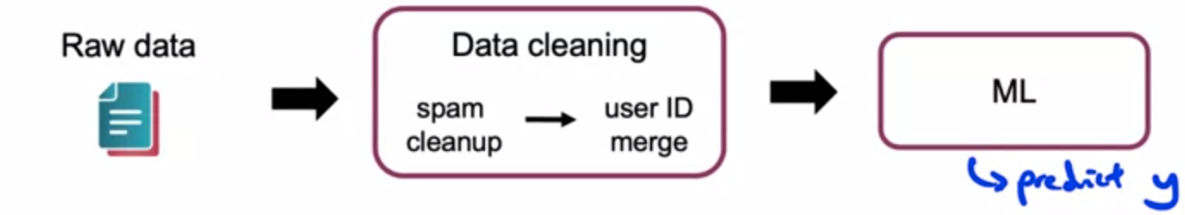
Inventory data, Obtaining data:

- Owned
- Crowdsource
- Pay for labels
- Purchase data

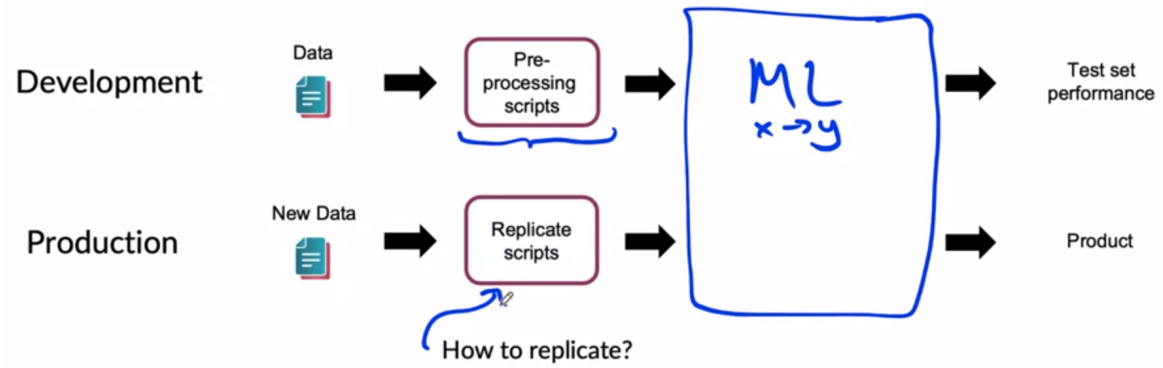
Labeling data:

- Options: In-house vs outsourced vs crowdsourced.
- Having MLEs label data is expensive. But doing this for just a few days is usually fine.
- Who is qualified to label?
    - Speech recognition - any reasonably fluent speaker
    - Factory inspection, medical image diagnosis - SME (subject matter expert)
    - Recommender systems - may be impossible to label well
- Don't increase data by more than 10x at a time.

Data Pipelines:



How to replicate scripts of data cleaning and preprocessing:



A POC (proof of concept) aims to decide if the application is workable and worth deploying.

- Focus on getting the prototype to work!
- it's ok if data preprocessing is manual. But take extensive notes/comments

Production phase:

- After project utility is established, use more sophisticated tools to make sure the data pipeline is replicable.

- E.g., Tensorflow Transform, Apache Beam, Airflow

Keep track of data provenance and lineage in the production system.

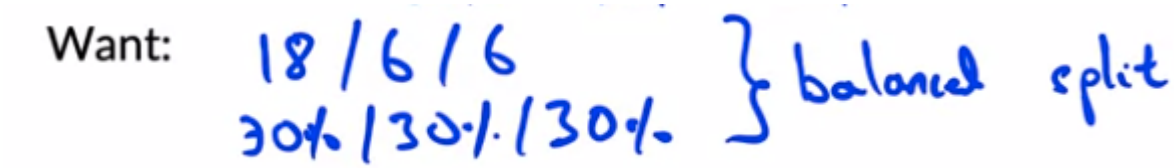
Is useful to use extensively Meta-data to keep track of our data. For example Time, Factory, line #, camera settings, phone model, inspector ID.

Meta-data useful for:

- Error analysis. Spotting unexpected effects.
- Keeping track of data provenance.

Balanced train/dev/test splits:

In the split, we want a balance of positives and negatives examples in our splits datasets. This makes our dataset more representative:



This is just needed for a small dataset because aimsAa random split will likely be representative or balanced for a large dataset.

Scoping:

How to pick a project to work on and planning out the scope of the project.

Example: Help eCommerce retailer looking to increase sales.

- Better recommender system
- Better search
- Improve catalog data
- Inventory management
- Price optimization

Questions:

- What project should we work on?
- What are the metrics for success?
- What are the resources (data, time, people) needed?

Scoping process:

- Brainstorm business problems (not AI problems)
    - Increase conversion
    - Reduce inventory
    - Increase margin (profit per item)
- Brainstorm AI solutions to business problems.
- Assess the feasibility and value of potential solutions. Double-check if an AI solution is valid.
- Determine milestones.

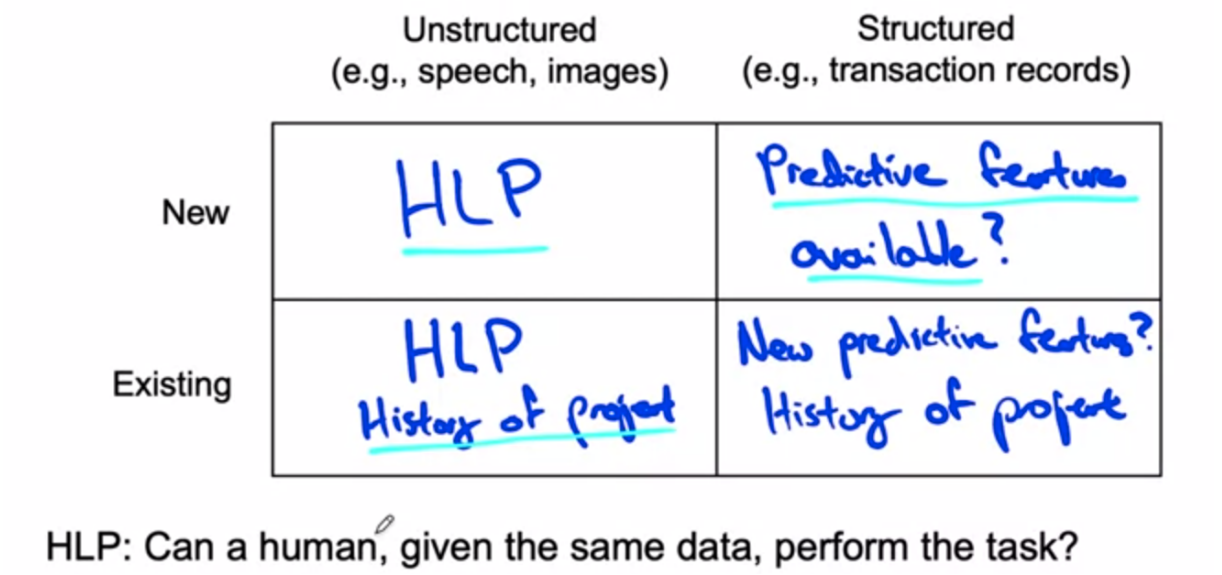
- Budget for resources.

For examples:

- Increase conversion:
  - Solution: Search, recommendations.
- Reduce inventory:
  - Demand prediction, marketing
- Increase margin (profit per item)
  - Optimizing what to sell (e.g., merchandising), recommend bundles

Feasibility: Is this project technically feasible?

Use external benchmarking (literature, other company, competitor)



Questions to ask:

- Do we have predictive features?
  - Given past purchases, predict future purchases. A good relationship mapping X->Y
  - Given the weather, predict shopping mall foot traffic. A good relationship mapping X->Y
  - Given DNA info, predict heart disease, this is not a good relationship, not doable.
  - Given social media chatter, predict demand for a clothing style, this is not feasible too.
  - Given the history of the stock's price, predict the future price of that stock, this is not feasible too or it needs more features than just the stock's price.

Diligence on value:

- MLE metrics
- Word-Level accuracy
- Query-level accuracy
- Search result quality
- User engagement

- Revenue
- Business metrics

Week 3: Data Definition and Baseline
If you wish to dive more deeply into the topics covered this week, feel free to check out these optional references. You won't have to read these to complete this week's practice quizzes.

Label ambiguity

Data pipelines

Data lineage

MLops

Geirhos, R., Janssen, D. H. J., Schutt, H. H., Rauber, J., Bethge, M., & Wichmann, F. A. (n.d.). Comparing deep neural networks against humans: object recognition when the signal gets weaker*. Retrieved May 7, 2021, from Arxiv.org website: https://arxiv.org/pdf/1706.06969.pdf