Machine Learning Projects

*The **Same** and The **Different***

Bijay Gurung / @bglearning
Python User Group Nepal
Meetup #14

Outline

1 machine_learning() # What and Why

2 compare(normal_prog, ml_prog)

3 for perspective in perspectives:

```
for perspective in perspectives:
        for topic in perspective:
                discuss(topic.same)
                discuss(topic.different)
```

What and Why Machine Learning?

Programming

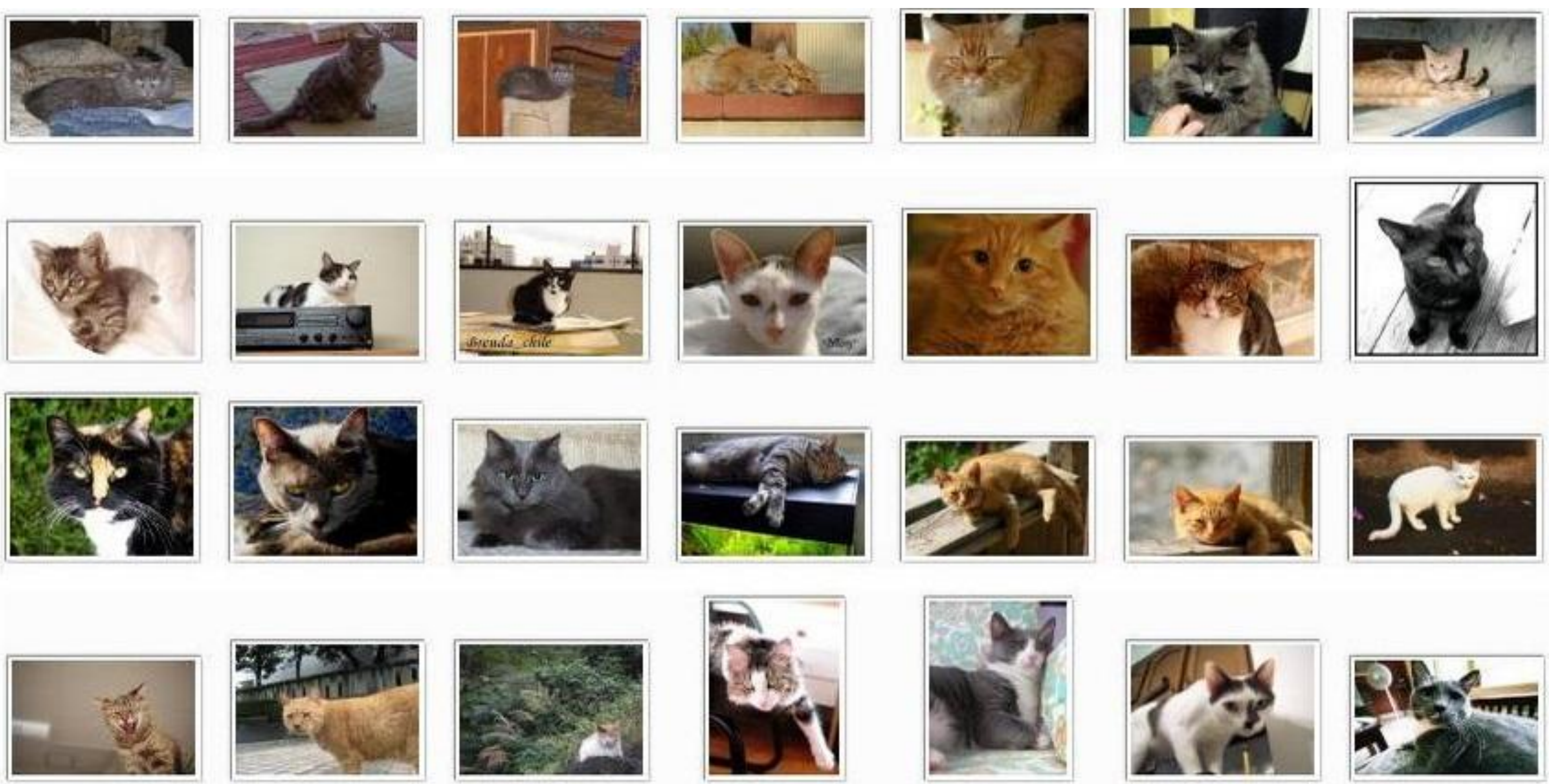
What and Why Machine Learning?

Cats!

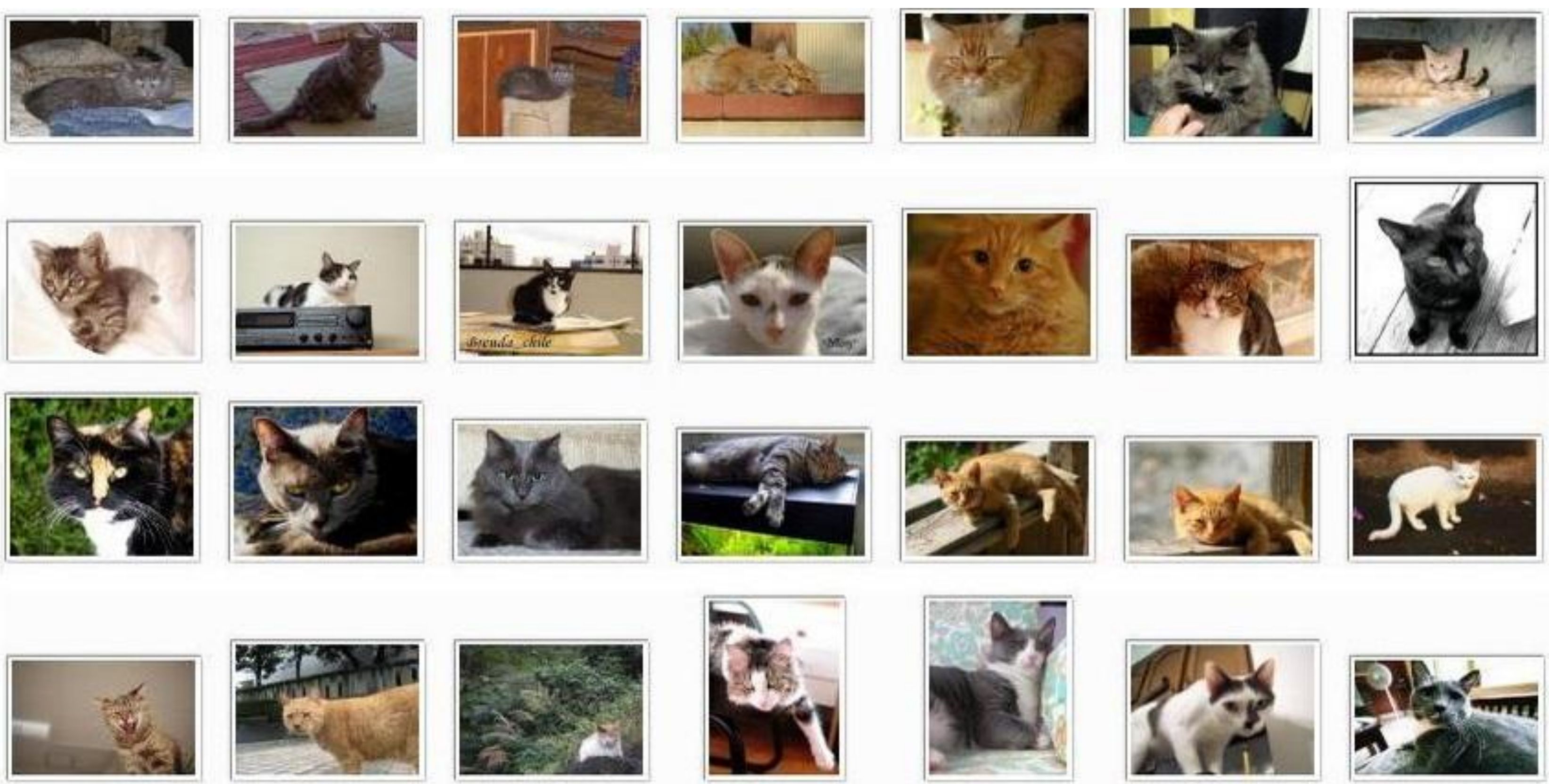
How can we detect cats in an image?

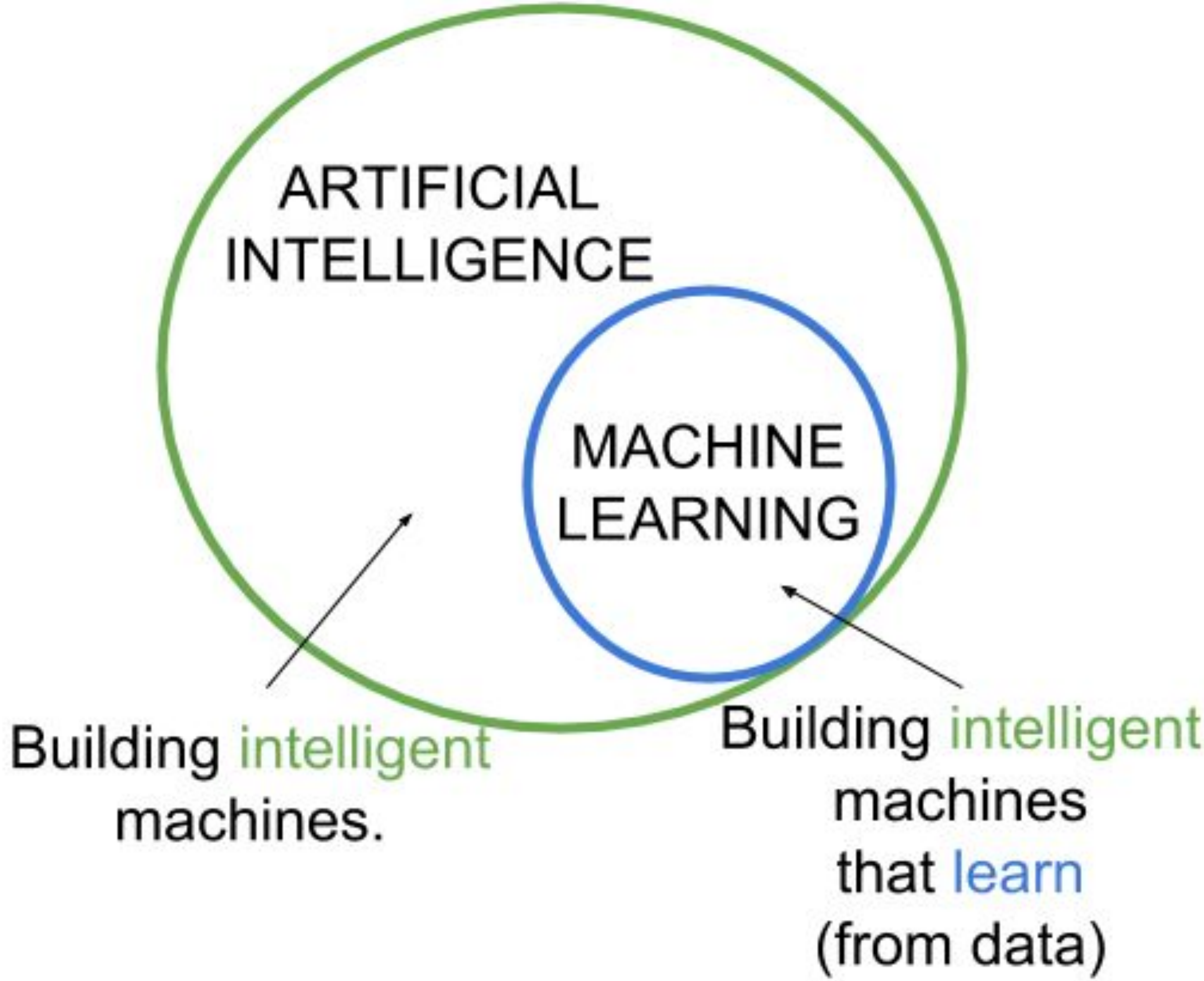
Cats!

“So many variations!” :|



ML:
“So much data!” :D

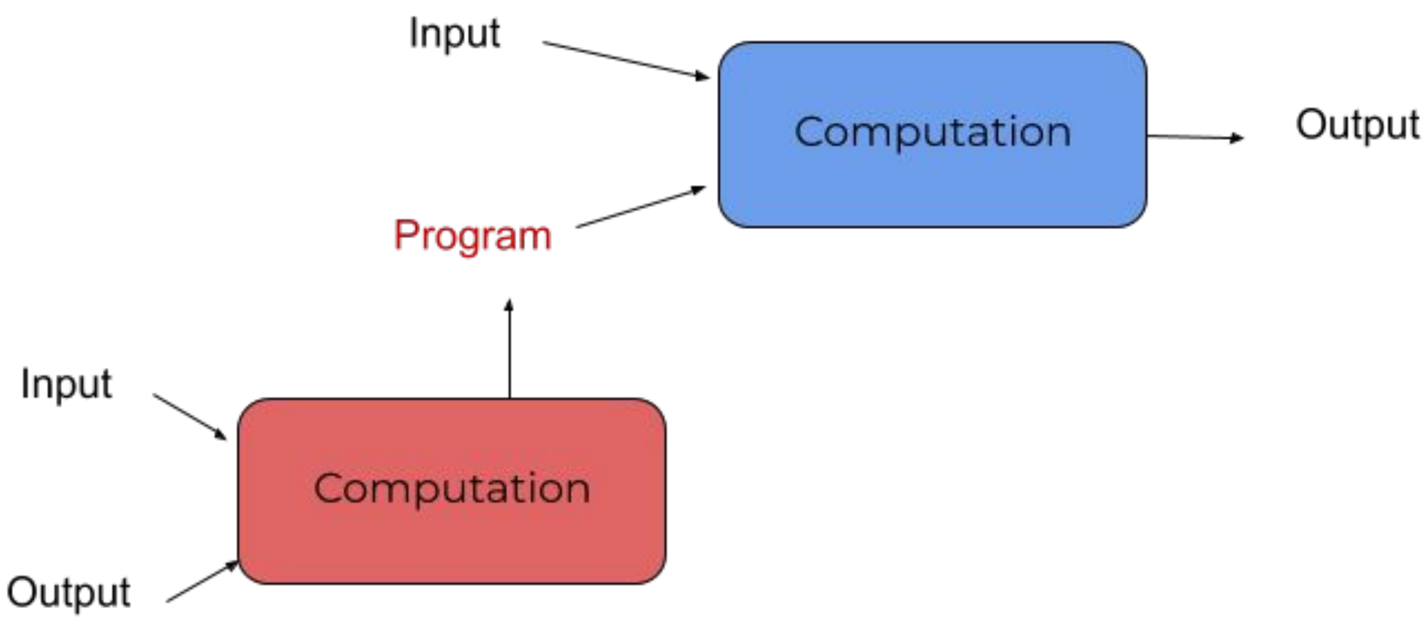



ARTIFICIAL
INTELLIGENCE
MACHINE
LEARNING
Building intelligent
machines.
Building intelligent
machines
that learn
(from data)

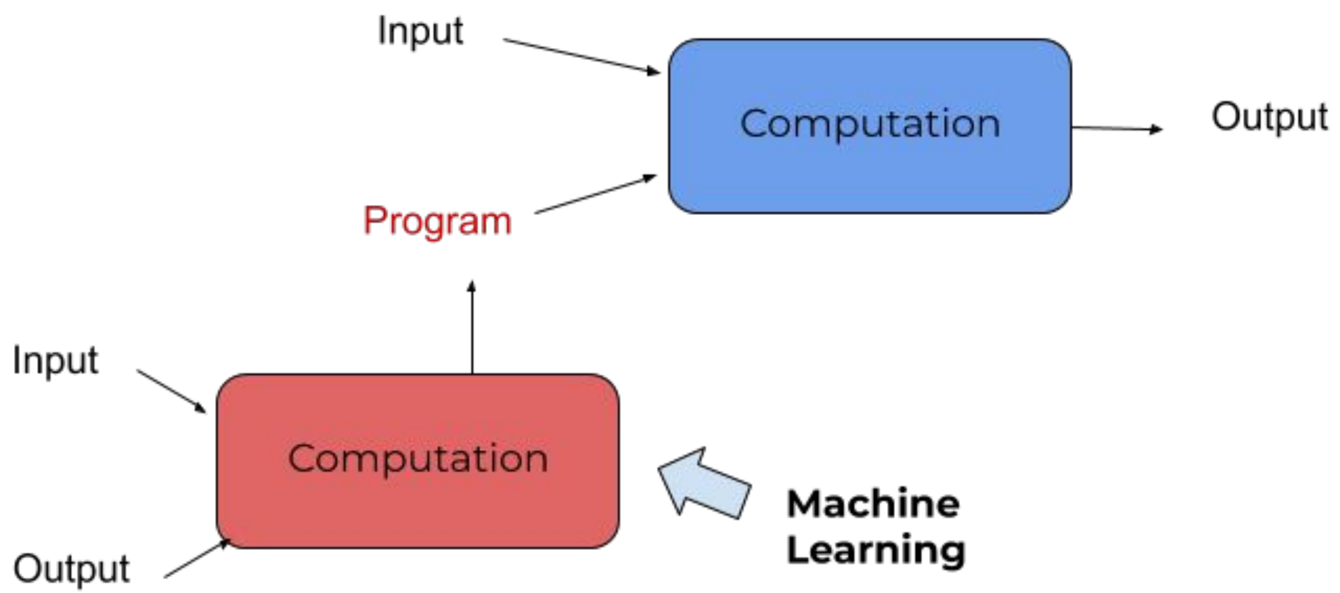
Why Machine Learning?

Because for a lot of problems we can’t explicitly define the solution.

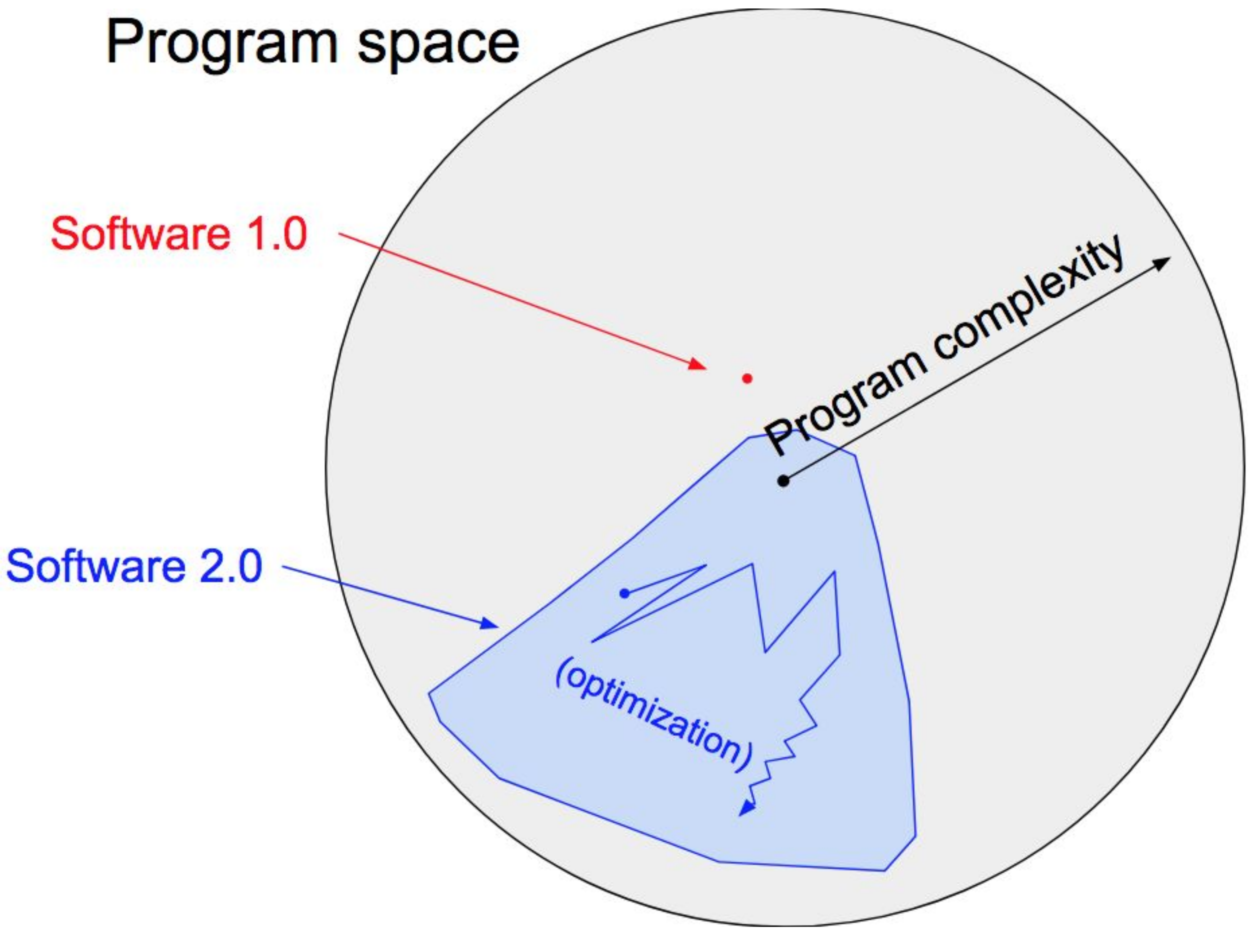
ML Programming



ML Programming



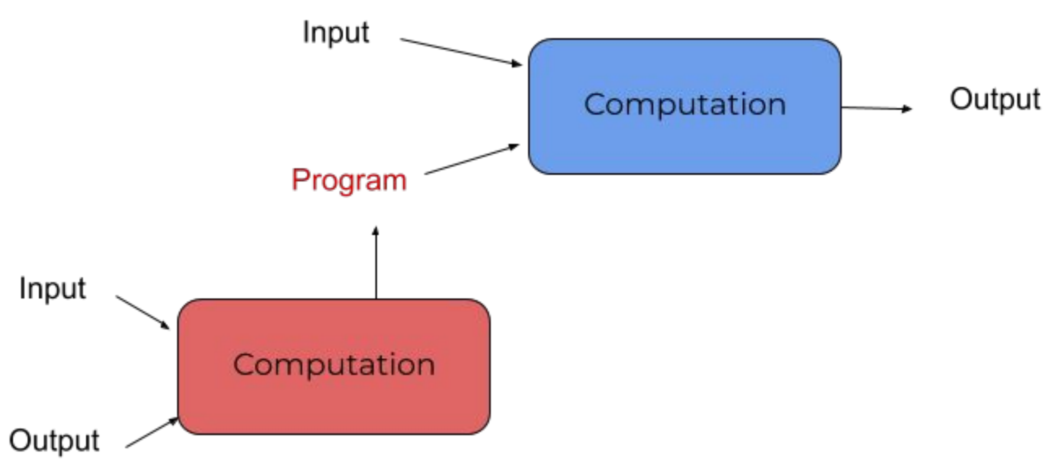
ML Programming



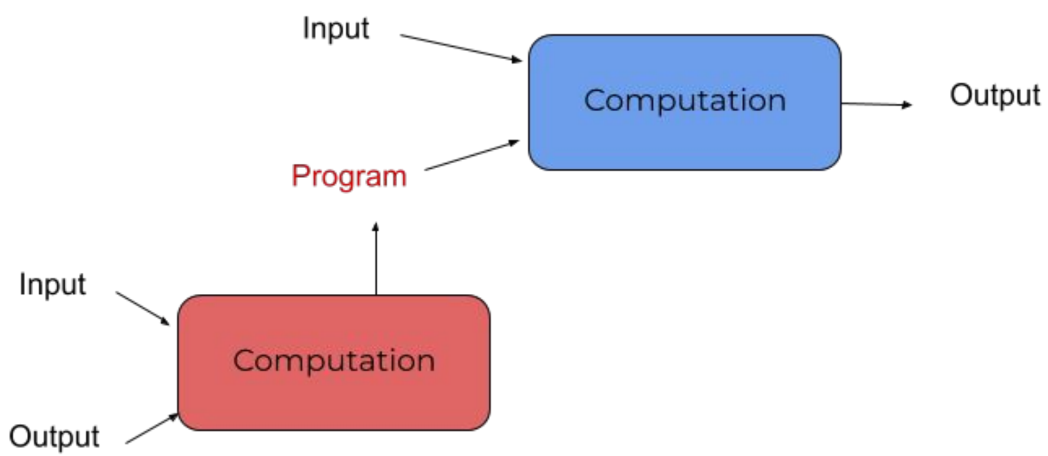
[Source: Software 2.0]

Approaches

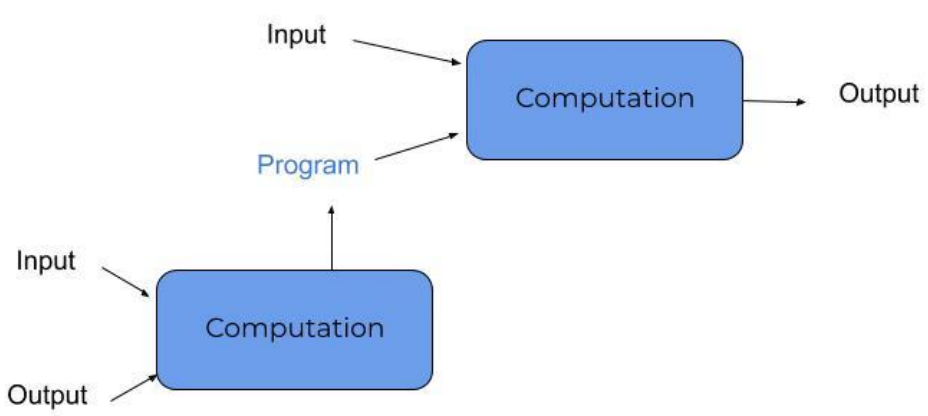

Input
Computation
Output
Program


Input
Computation
Output
Program
Input
Computation
Output

Approaches | Extremes

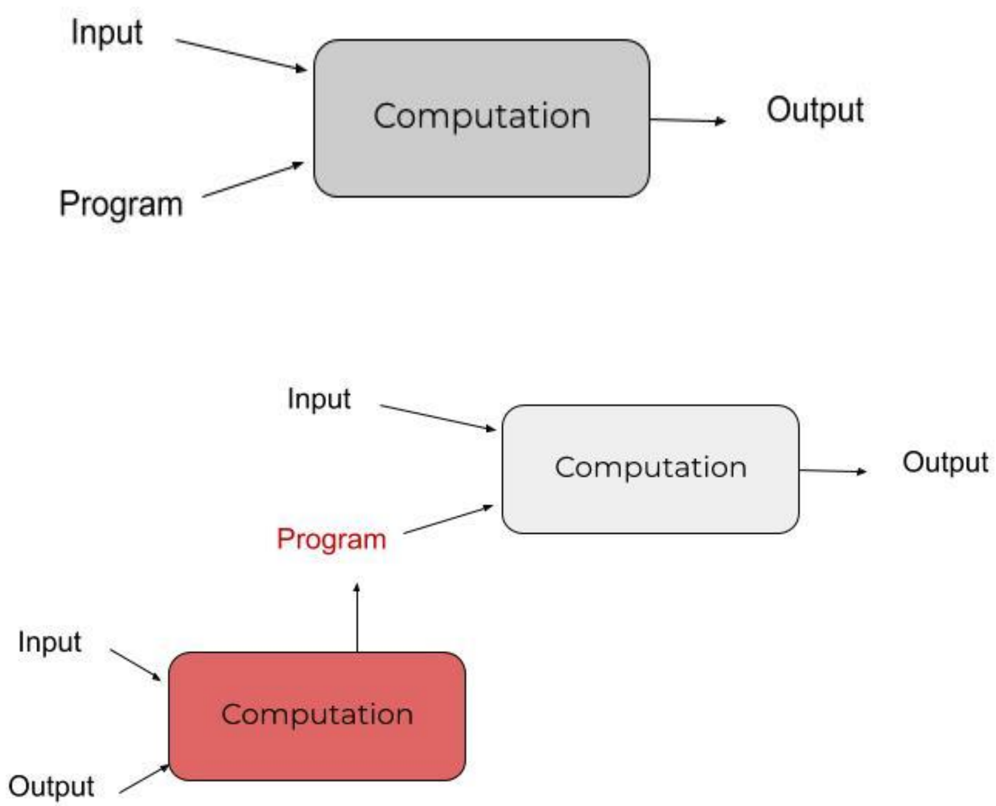


Approaches | Extremes



“They are doing the same thing: computation”

Approaches | Extremes



“ML is different; it’s experiments, it’s science!”

ML Project | Approaches

- The Software Engineering approach: Treat ML projects like any other software project.

- The Academic approach: Not think of ML projects as software projects.

ML Project | Approaches

- The Software Engineering approach: Treat ML projects like any other software project.
  > *Needs to pay heed to what's **different**.*

- The Academic approach: Not think of ML projects as software projects.

ML Project | Approaches

- The Software Engineering approach: Treat ML projects like any other software project.
  > *Needs to pay heed to what's* ***different***.

- The Academic approach: Not think of ML projects as software projects.
  > *Needs to pay heed to what's the* ***same***.

Normal vs ML Projects

Perspectives:

Project/ Product Owner	Project Manager
Developer	DevOps

Project/ Product Owner	Project Manager
Developer	DevOps

Requirements
| The Same

Both have the same goals:

Solve Problems, Add Value

Requirements | The Same

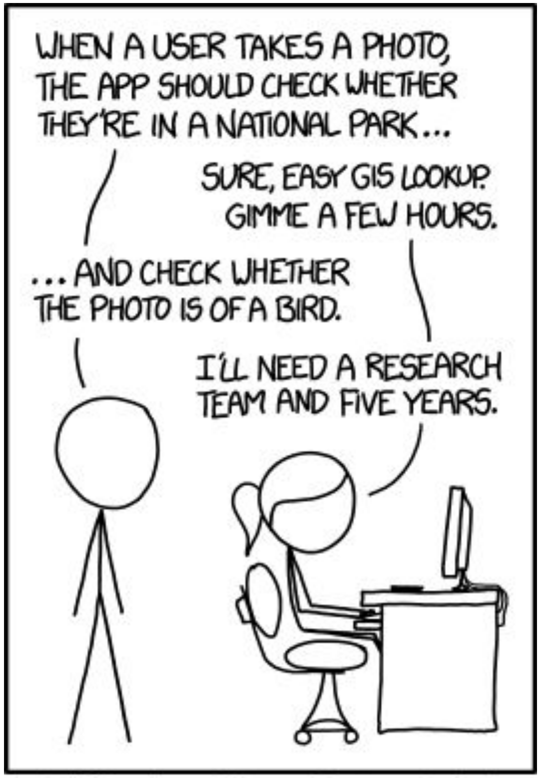
Bk

Requirements | The Different

More questions to ask:

- Is ML necessary?
- What's the expectation for the ML component?

Requirements | The Different



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

Feasibility
| The Same

Needs to be feasible from various perspectives:

- Technical
- Economic
- Legal
- Operational
- Schedule

Feasibility
| The Different

For ML, need to look at technical feasibility for the ML component:

- Do we have the necessary data? If not, can we acquire it?
- What's the state-of-the-art in this area?

Project/ Product Owner	**Project Manager**
Developer	DevOps

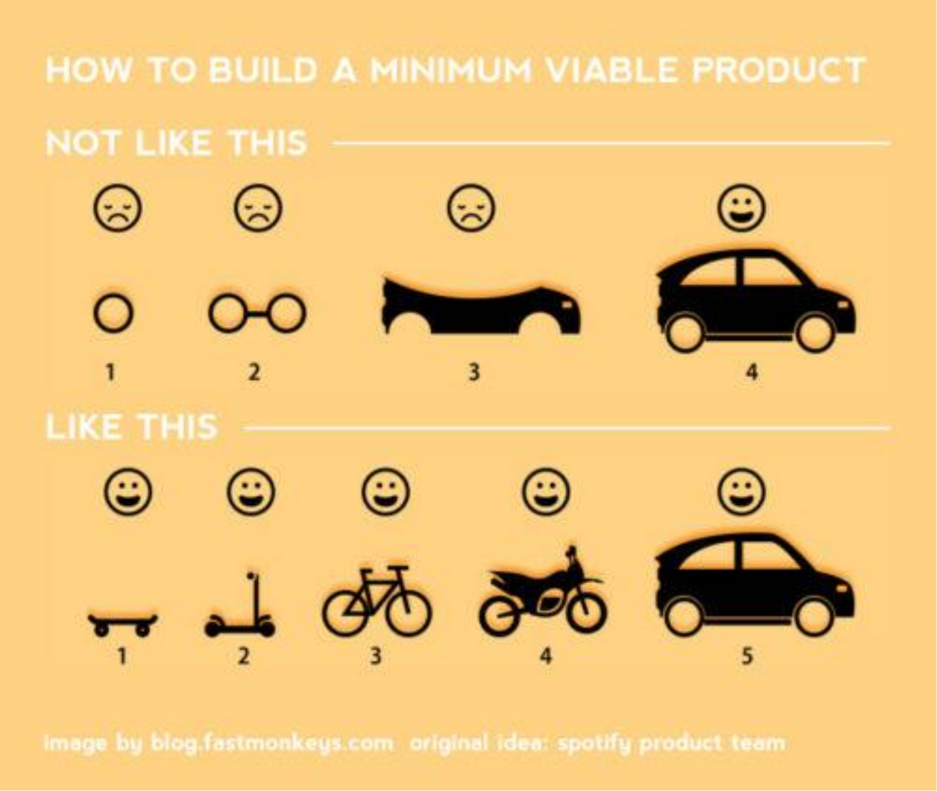
Iterative Approach | The Same

- Iterate! Iterate! Iterate!

[Source: ML Engineering Loop]

Iterative Approach | The Same

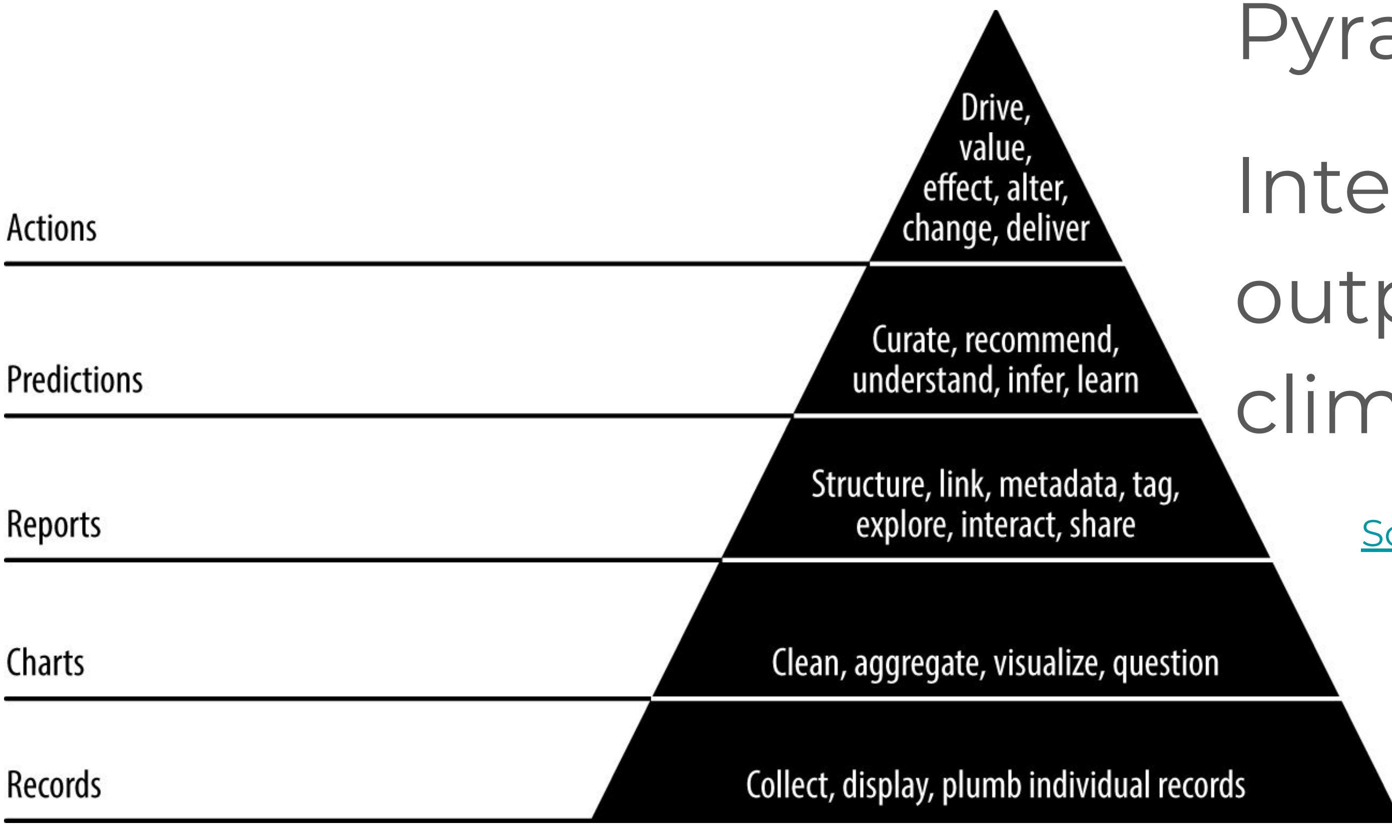
- Have intermediate outputs



Iterative Approach | The Different

- What are “intermediate outputs” in ML projects?

Iterative Approach | The **Different**



The Data Value Pyramid.

Intermediate outputs as we climb it.

[Source: Agile Data Science]

Task formulation & Planning | The Same

- Need to have some level of planning: a systematic approach
- Formulate tasks with clear what, why, and how

Task formulation & Planning | The Different

- Put in more areas for flexibility in the process.
  Because: ML projects have a big experimental component.

Task formulation & Planning | The Different

- Put in more areas for flexibility in the process.
  A lot of ML projects are experimental.

  Eg: Kanban vs Sprints

Task formulation & Planning
| The **Different**

- Put in more areas for flexibility in the process.
  A lot of ML projects are experimental.

  Do away with
  User Stories?

Task formulation & Planning | The Different

- Try to be more precise about what “Done” means

Task formulation & Planning
| The Different

- Try to be more precise about what "Done" means

  PM: "Is the task Done?"
  MLE:
  "Yes... umm... no. I've a feeling it needs more tuning... "

Project/ Product Owner	Project Manager
Developer	DevOps

Version Control | The **Same**

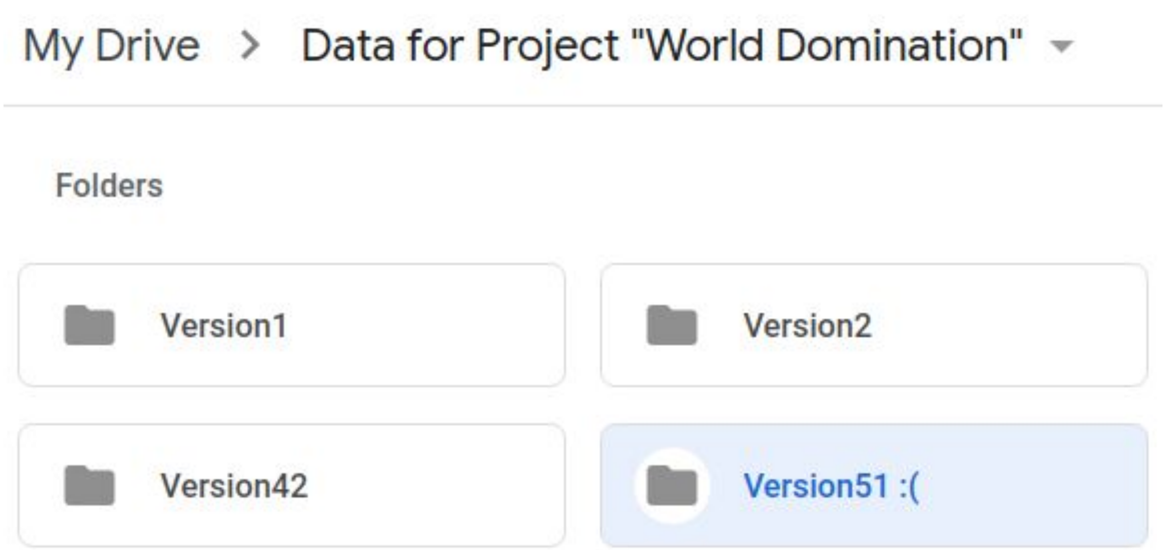
- Version Control your code!

Version Control | The Different

- Version Control your code!
  Also: data, model, configs.

Version Control | The Different

- Version Control your code!
  Also: data, model, configs.



Version Control | The Different

- Version Control your code!
  Also: data, model, configs.

Version Control | The Different

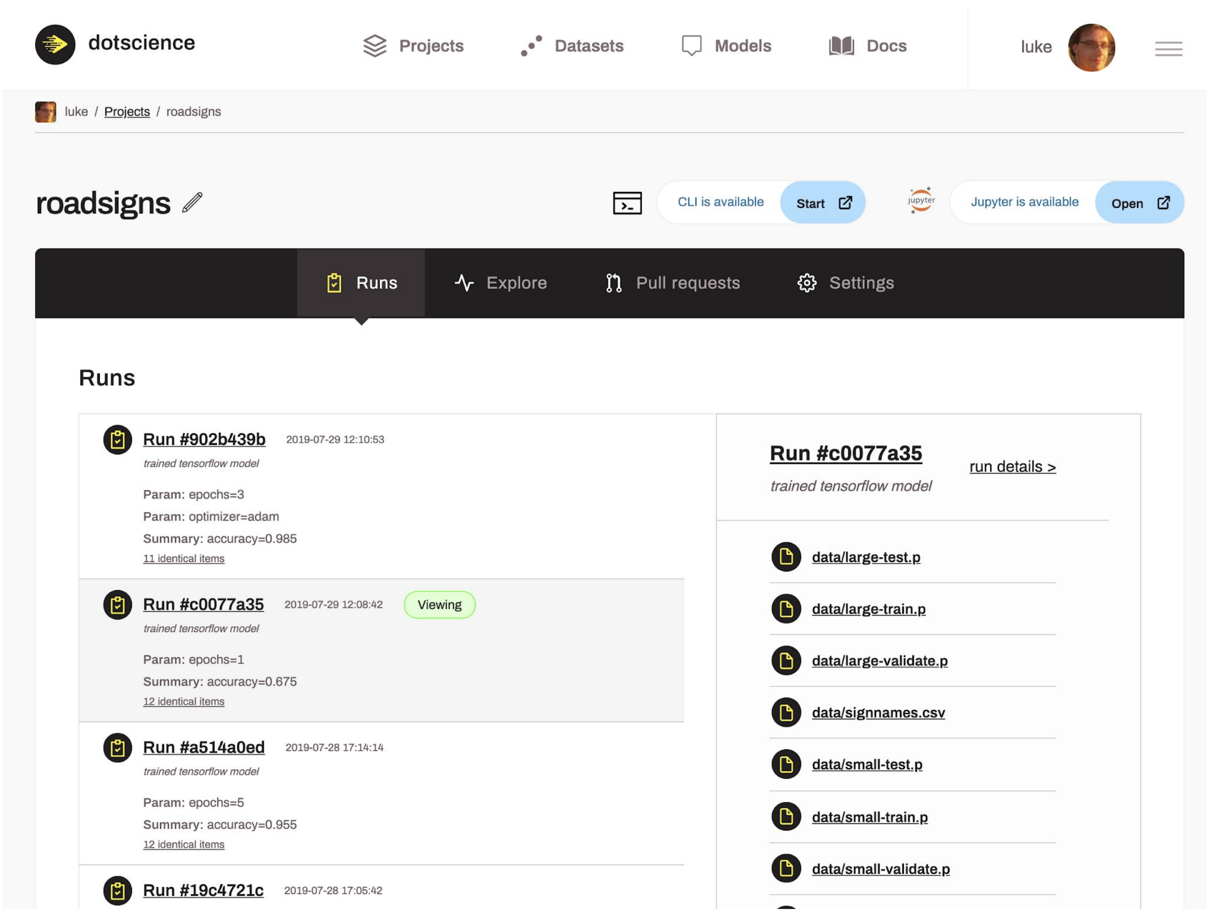
- Version Control your code!
  Also: data, model, configs.

DVC

comet

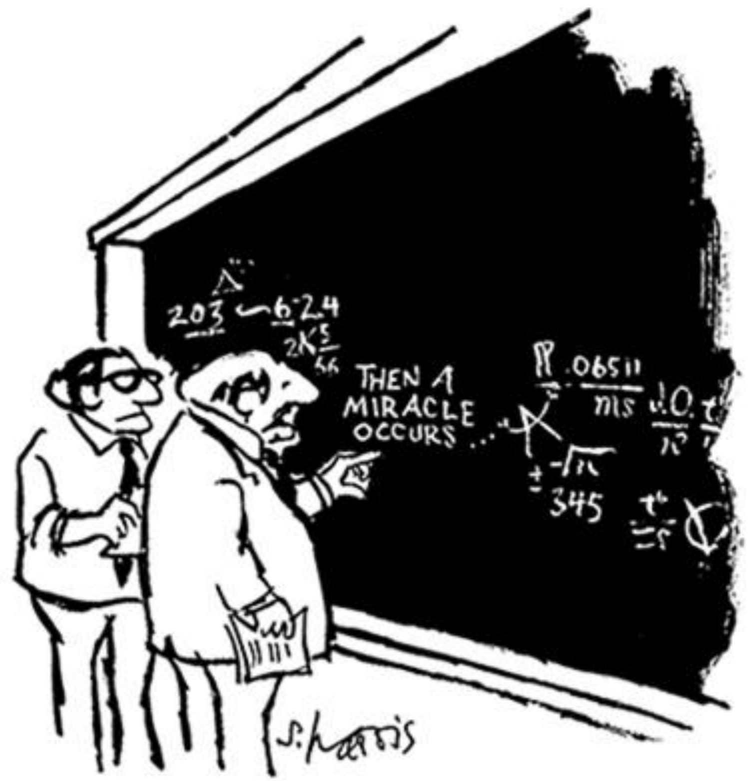
dotscience

Version Control | The Different



Version Control | The **Different**

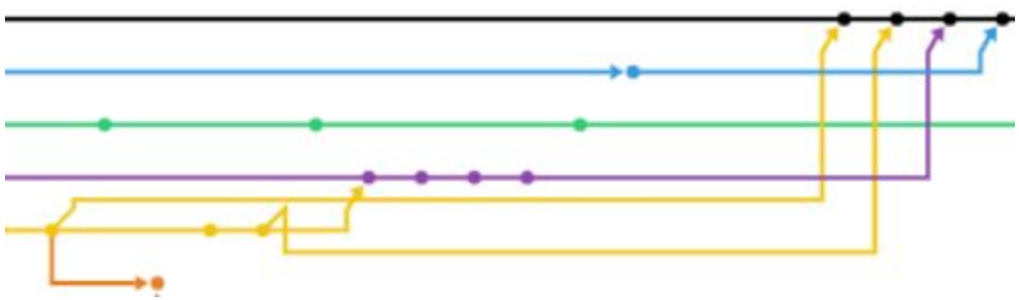
- Think about [reproducibility]



"I think you should be more explicit here in step two."

Version Control | The Different

- Multi-experimental-branch Tendency



Formulate a policy

Flexibility | The Same

- Balance between future-proofing and getting things done

Flexibility | The Same

- Balance between future-proofing and getting things done

Flexibility
| The Different

- A lot of the flexibility is for experimentation rather than for changes in product features and implementation.

Flexibility
| The Different

- Which one should I do?

```
class MyModel(Model):
    def __init__(self):
        self.input_embedding = Embedding(100)
        self.encoder = LSTM(100, 200)
        self.my_novel_bits = ...
```

```
class MyModel(Model):
    def __init__(self,
                 input_embedding: TextFieldEmbedder,
                 encoder: Seq2SeqEncoder):
        self.input_embedding = input_embedding
        self.encoder = encoder
        self.my_novel_bits = ...
```

On the parts that aren't what you're focusing on, you start simple. Later add ELMo, etc., *without rewriting your code.*

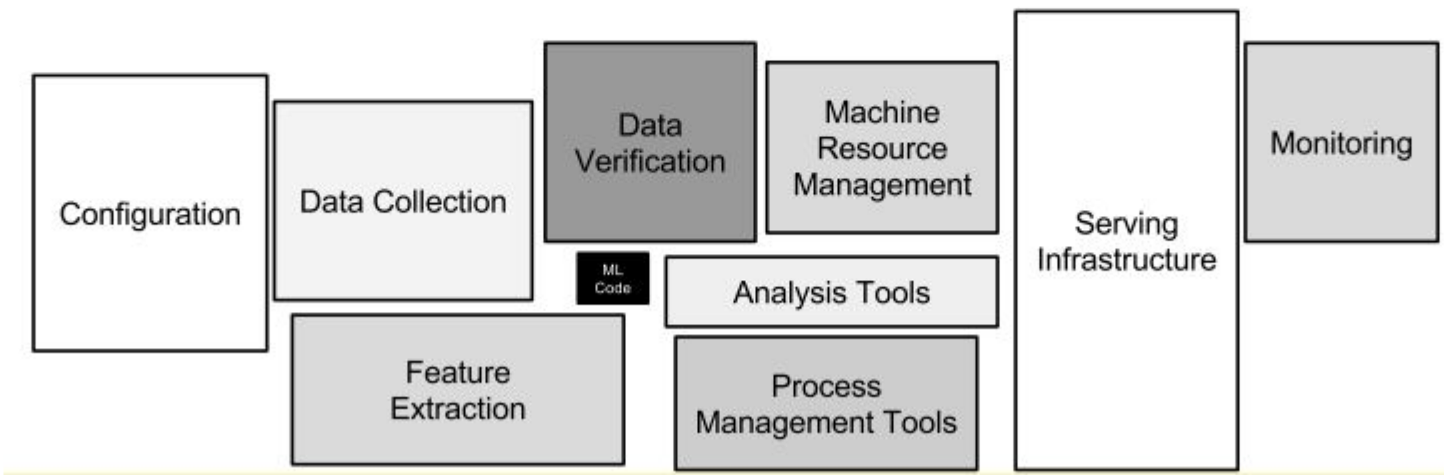
[Source: Writing Code for NLP Research, AllenNLP]()

Components & Abstractions | The Same

- Break into cohesive, de-coupled components

Components & Abstractions | The Same

- Break into cohesive, de-coupled components; ML is small part of a bigger whole



[Source: Hidden Technical Debt in Machine Learning]()

Components & Abstractions | The Different

- Components can “entangle” in weird ways.

Components & Abstractions
| The Different

- Components can “entangle” in weird ways.
  The CACE principle:
  Changing Anything Changes Everything

Components & Abstractions | The Different

- Components can “entangle” in weird ways.
  The CACE principle:
  Changing Anything Changes Everything
  Why?
  Dependence on both code and data

Testing
| The Same

- Standard tests (unit tests, integration tests, etc) important

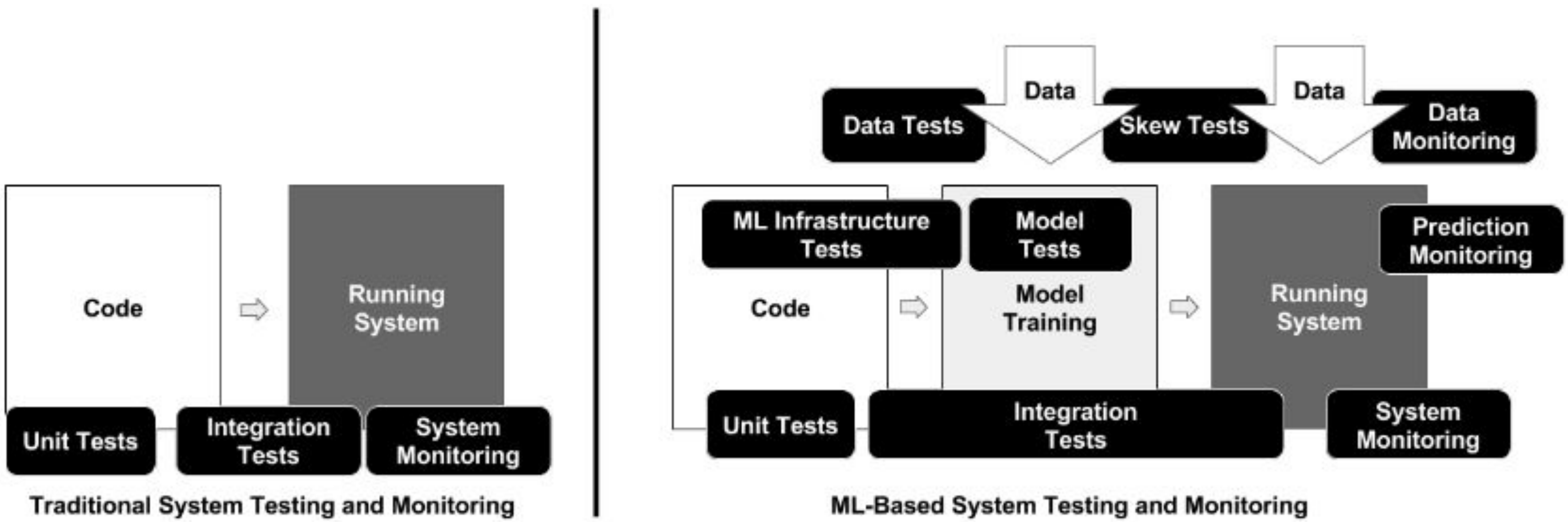
Testing
| The Same

- Standard tests (unit tests, integration tests, etc) important

- Focus on critical and/or riskier parts

Testing | The Different

More tests necessary!



[Source: A rubric for ML production readiness]

Good Practices
| The Same

- Use of linters, style-guides, etc
- Readable, well-commented code
- Documentation

Good Practices | The Different

- Data Documentation

Table	Table Notes	Field	Definition	Example Value	Field Notes
website_actions	Tracks user actions on the website; updated nightly	user_id	User ID	55555	Tie to user table
		action	User action on website	watched_video	Values: clicked_button, watched_video, signed_up_for_thing
		object_id	Object a user interacted with	222	Tie to object table
		timestamp	Date and time of user interaction	2018-01-02 1:01:01	Data appears from 1/1/18 to present

Source: Field Notes: Building Data Dictionaries

Good Practices | The Different

- Experimental logs

Experimenter	git SHA	Background Search Method	Model	Dataset	Train Acc	Validation Acc	Notes
Pradeep	fc8d6ca3	Lucene	QAMNS (50d)	Intermediate	0.3114	0.3045	patience=20
Pradeep	fc8d6ca3	Lucene	QAMNS (300d)	Intermediate	0.8317	0.3864	patience=20
Pradeep	fc8d6ca3	BOW-LSH question+answers Glove 50d	QAMNS (50d)	Intermediate	0.3008	0.35	patience=20
Pradeep	fc8d6ca3	BOW-LSH question+answers Glove 50d	QAMNS (300d)	Intermediate	0.7466	0.4227	patience=20
Pradeep	fc8d6ca3	BOW-LSH question+answers Glove 300d	QAMNS (50d)	Intermediate	0.3946	0.3591	patience=20
Pradeep	fc8d6ca3	BOW-LSH question+answers Glove 300d	QAMNS (300d)	Intermediate	0.7311	0.4227	patience=20
Pradeep	fc8d6ca3	BOW-LSH+IDF question+answers Glove 300d	QAMNS (300d)	Intermediate	0.7446	0.4227	patience=20
Pradeep	fc8d6ca3	BOW-LSH+IDF question+answers Paragram 300d	QAMNS (300d)	Intermediate	0.7853	0.3955	patience=20
Pradeep	fc8d6ca3	Lucene	QAMNS (300d)	SciQ	0.5551	0.571	patience=6
Pradeep	fc8d6ca3	BOW-LSH question+answers Glove 300d	QAMNS (300d)	SciQ	0.5434	0.524	patience=6

Source: Writing Code for NLP Research, AllenNLP

Project/ Product Owner	Project Manager
Developer	**DevOps**

Deployment & Maintenance | The Same

- Need Continuous Integration, Build Automation, Monitoring, etc

Deployment & Maintenance | The Different

- Also need to monitor data, model performance, etc.

Deployment & Maintenance | The Different

How is MLOps different from DevOps?

- Data/model versioning != code versioning - how to version data sets as the schema and origin data change
- Digital audit trail requirements change when dealing with code + (potentially customer) data
- Model reuse is different than software reuse, as models must be tuned based on input data / scenario.
- To reuse a model you may need to fine-tune / transfer learn on it (meaning you need the training pipeline)
- Models tend to decay over time & you need the ability to retrain them on demand to ensure they remain useful in a production context.

[Source: MLOps, Microsoft]

Deployment & Maintenance | The Different

Essence:

Not just code artefacts, overall application performance

Also, data, model artefacts, and their operational behavior

Thinking Meta

Where is the difference coming from?

ML projects:

- Depend on/tightly coupled with data (in addition to code & infrastructure)
- Can be thought of as being “Two phase”
- Is usually highly experimental

Where is the similarity coming from?

ML Projects:

- Have ML component as just one part of a bigger system that’s meant to solve a problem.

Wrapping Up

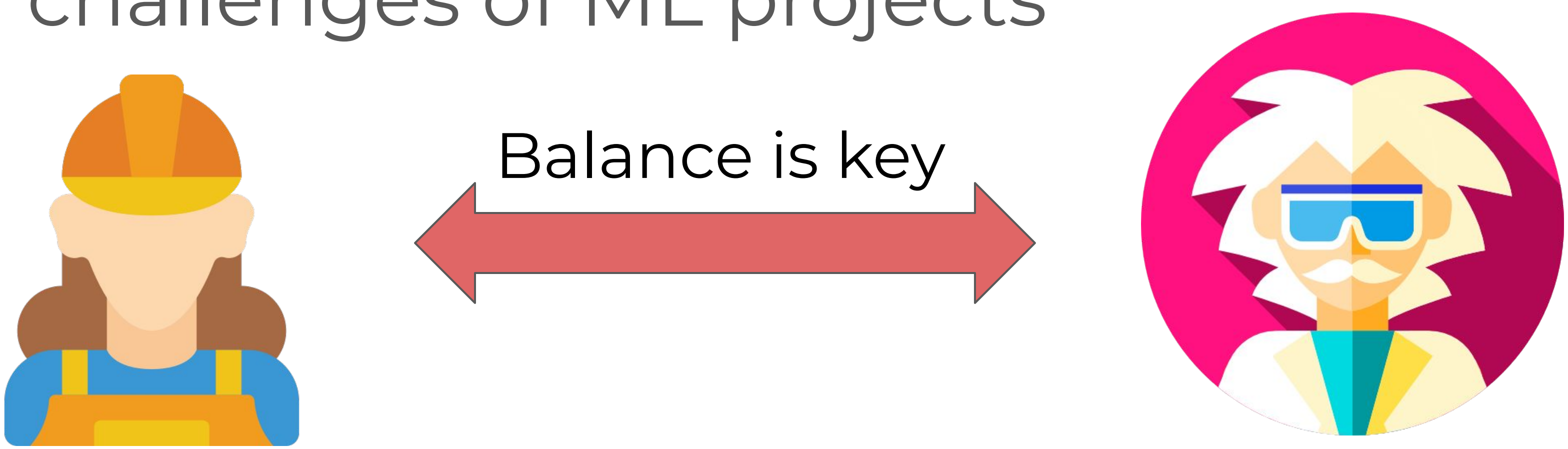
What?

Machine Learning Projects are both same as and different than “normal” software projects.

So What?

When doing ML projects, it's best to:

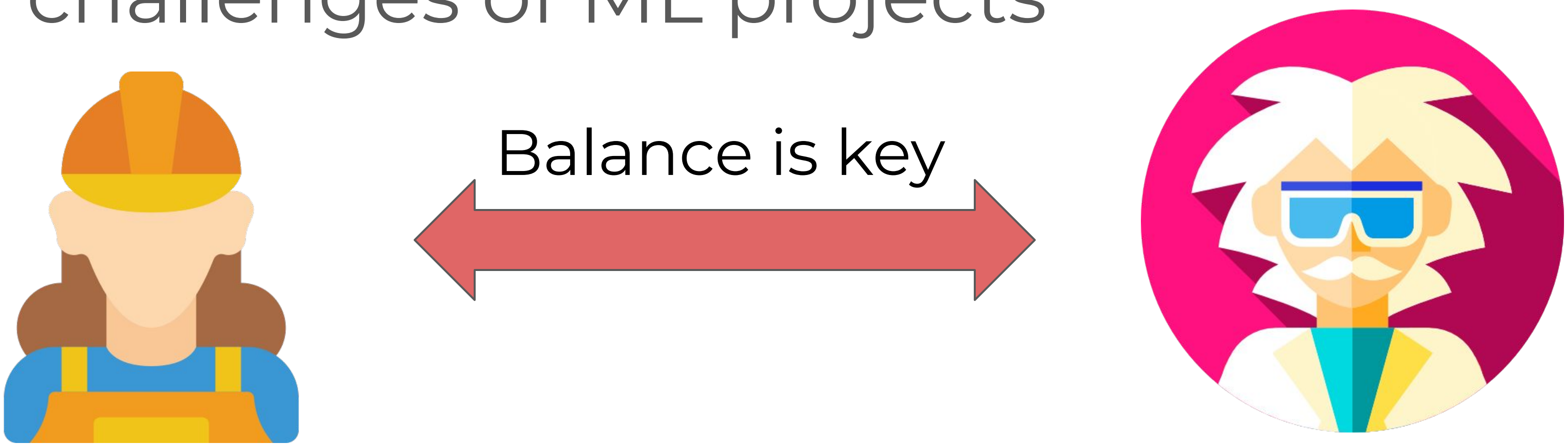
- Use Software Engineering principles
- And also think about the unique challenges of ML projects



So What?

When doing ML projects, it's best to:

- Use Software Engineering principles
- And also think about the unique challenges of ML projects



Now What? Up to you :)