Day 5

Welcome to GitHub

What is GitHub?

GitHub is a code hosting platform for **version control** and **collaboration**. It lets you and others work together on projects from anywhere

Git:

- The Git portion, is a low-level system created by Linus Torvalds to:
- Manage file version
- Easily add in patches and updates created by collaborators
- Backup files

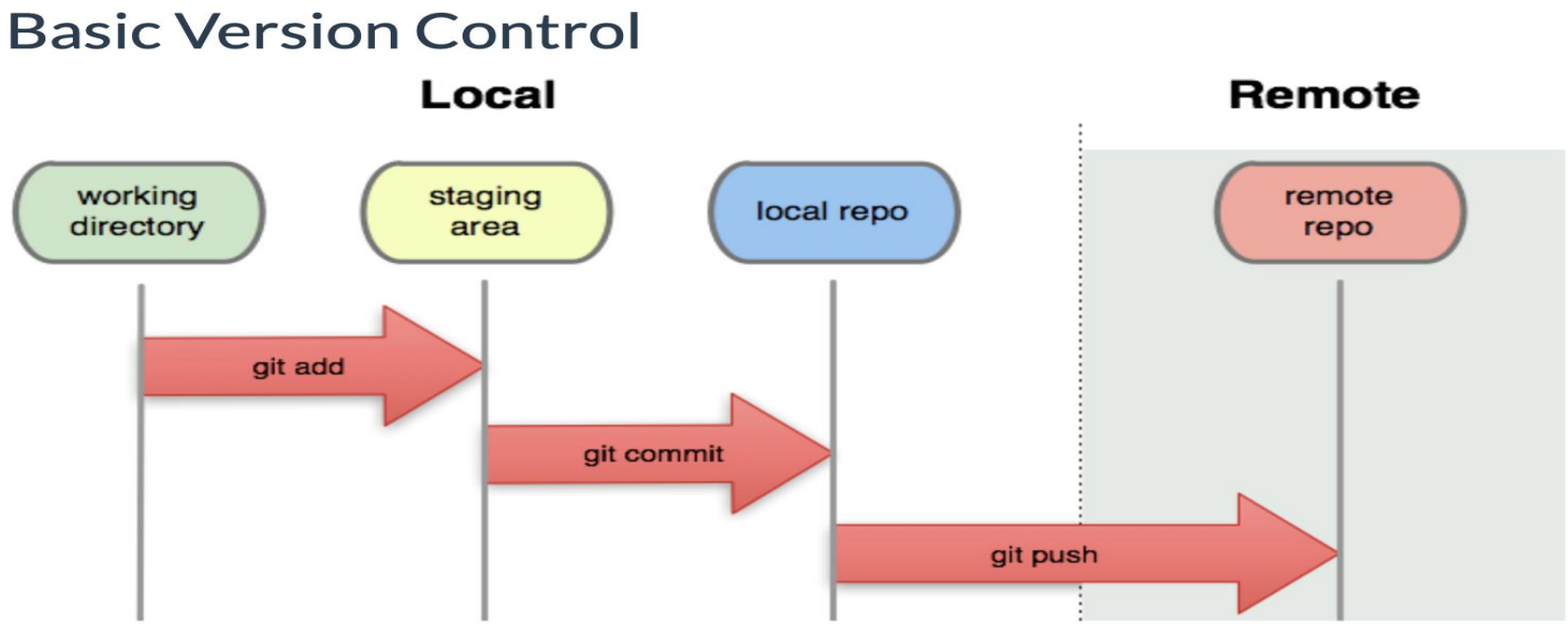
Hub:

- The Hub portion takes everything we do to a social level. GitHub is a web-based social network for users to host and share cod
- In GitHub, you can
  - Follow the work of other people
  - Share your work to the public
  - Contribute to other projects.

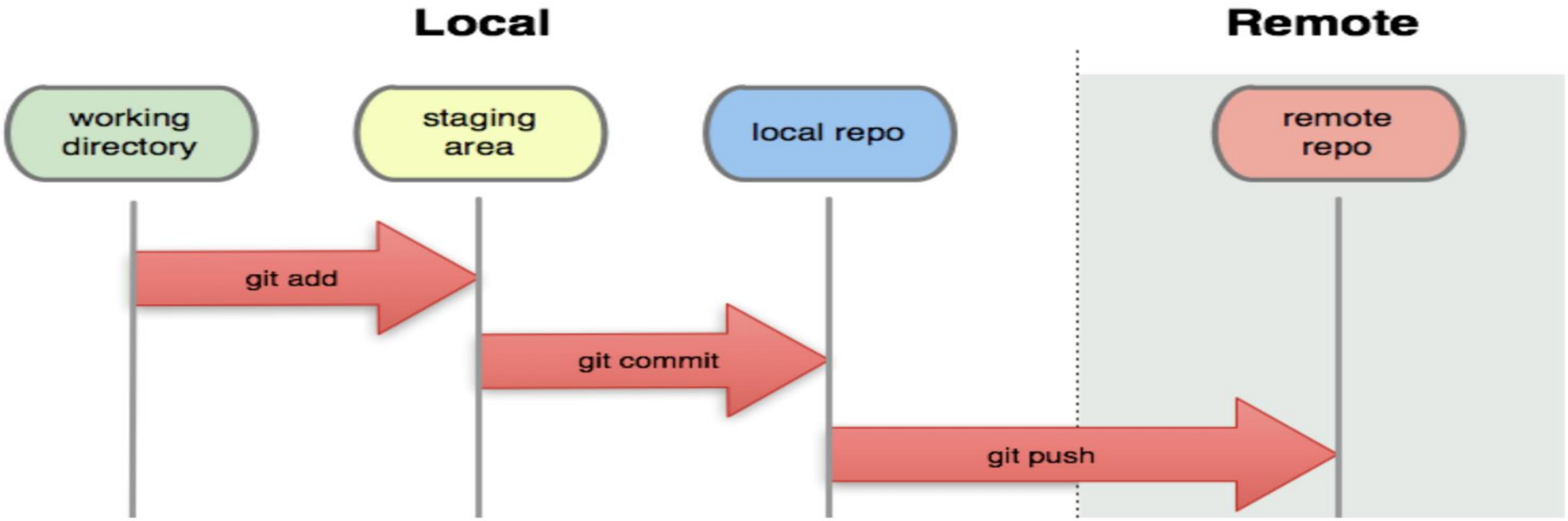
What is Version Control?

The secret sauce of how GitHub is able to keep your repository history are snapshots called **Commits**:

- Pictures of the state of your folder that you take and give names
- When you finish making edits to the repo, you make a commit to save the current state



Basic Version Control



Working Directory

- This is the repository or folder that you are currently making your changes in. However, you need to directly tell git that you want to track certain files - otherwise, git won't take snapshots of those changes

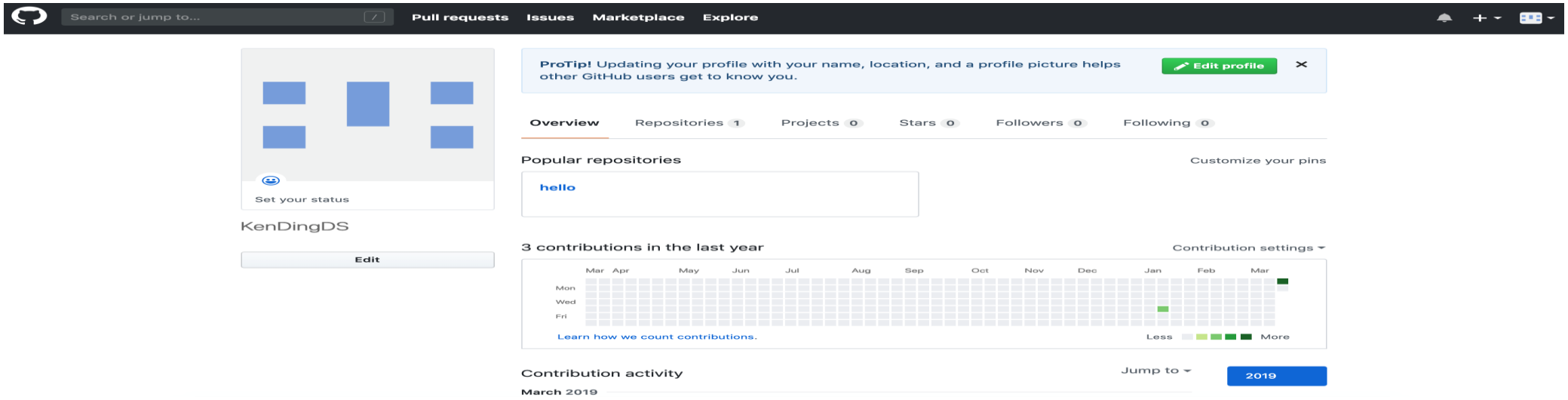
Staging Area

- The staging area contains all the tracked(added) files and information about what will go into your next commit. Think of this as the purgatory between snapshots. When you finally "Commit", it saves the snapshot and your staging area will be clear

GitHub Repository

- You send and receive snapshots to and from the GitHub Server through pushing (to send your changes) and pulling (to pull changes from collaborators). By doing so, you are able to:
- Host your work publicly
- Keep file versions managed with multiple people

Explore the GitHub Page



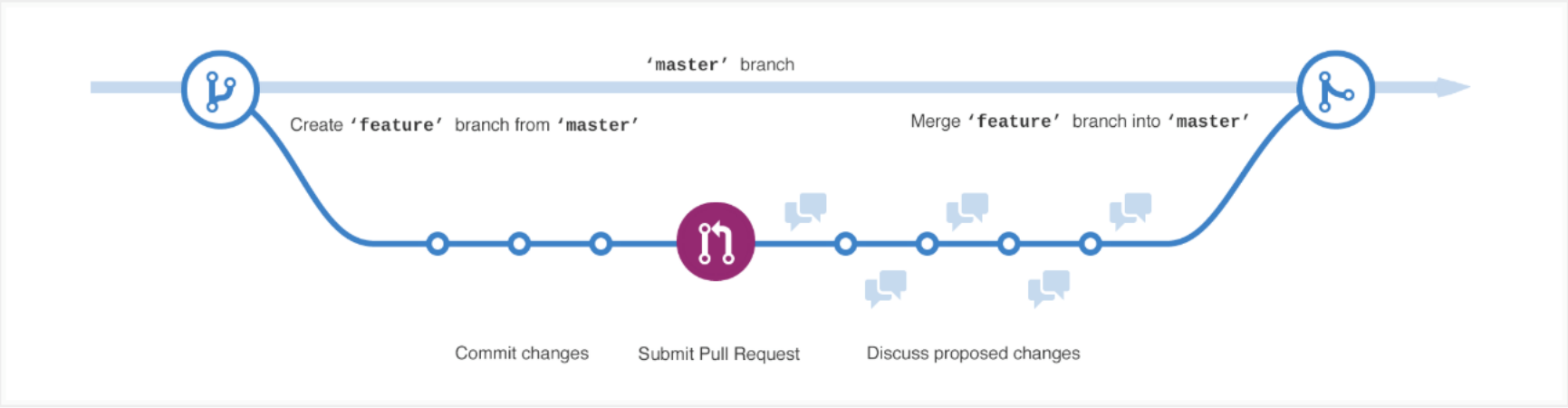
Category	What it means
Repositories	Project pages
Code	Files inside projects
Commits	Notes users have made to explain code changes they've made
Issues	Problems or comments users have made about a repository
Wikis	User-editable wiki pages for repositories
Users	People's profiles

Term	Definition
Code	The thing that we care about controlling the version of! This could be anything from plain text to Python to HTML, it doesn't matter.
Commit	A single, tracked, change to the code base. It comes with a message explaining what the change is. The history of a repository is made up of a chain of many commits.
Issue	A way to keep track of tasks, enhancements, and bugs for your projects. It's kind of like starting a thread on a forum where you can have a wider, public discussion about a relevant topic with the repository owners and other users.
Watch	Get notifications when things change with the repo.
Star	Mark the repo as a favorite!
Fork	Think "fork in the river". This copies the repo to another organization, usually yours, so you can do stuff with it that won't affect the main "stream" (I hope you like this metaphor because this isn't going to be the last time). This is pretty common among open source projects.
Pull Request	A formalized mechanism for incorporating commits into the main code base of a repository. More later.
Project	Think "project management". It gives you a pseudo kanban board and lets you organize your issues and pull requests with milestones, due dates, and other things that may be useful for productivity management.
Wiki	It's a wiki. Usually it has more meta-level details or instructions and is a good way to encourage users to participate or share best practices.
Insights	Meta-level analytics.
Branch	If the river is your primary code base, a branch is an offshoot that contains work (commits) that might be exploratory, or in progress, or used for testing, or whatever. Branches can eventually flow back into the main code base via a Pull Request, or they can just stay there, kind of like a separate, parallel project with a lot of shared history.
Clone	Cloning refers to the process of copying a repository on GitHub to a local computer. They call it cloning because there's no difference between a repository on GitHub or a repository on your computer, but I wish they would have stuck to the river theme.
README	A plain-text document that introduces the project and explains how to use it. On GitHub, these are written in markdown for text formatting, because of it's simplicity.

Create a Branch

Branching is the way to work on different versions of a repository at one time.

- By default your repository has one branch named master which is considered to be the definitive branch.
- We use branches to experiment and make edits before committing them to master.
- When you create a branch off the master branch, you're making a copy, or snapshot, of master as it was at that point in time. If someone else made changes to the master branch while you were working on your branch, you could pull in those

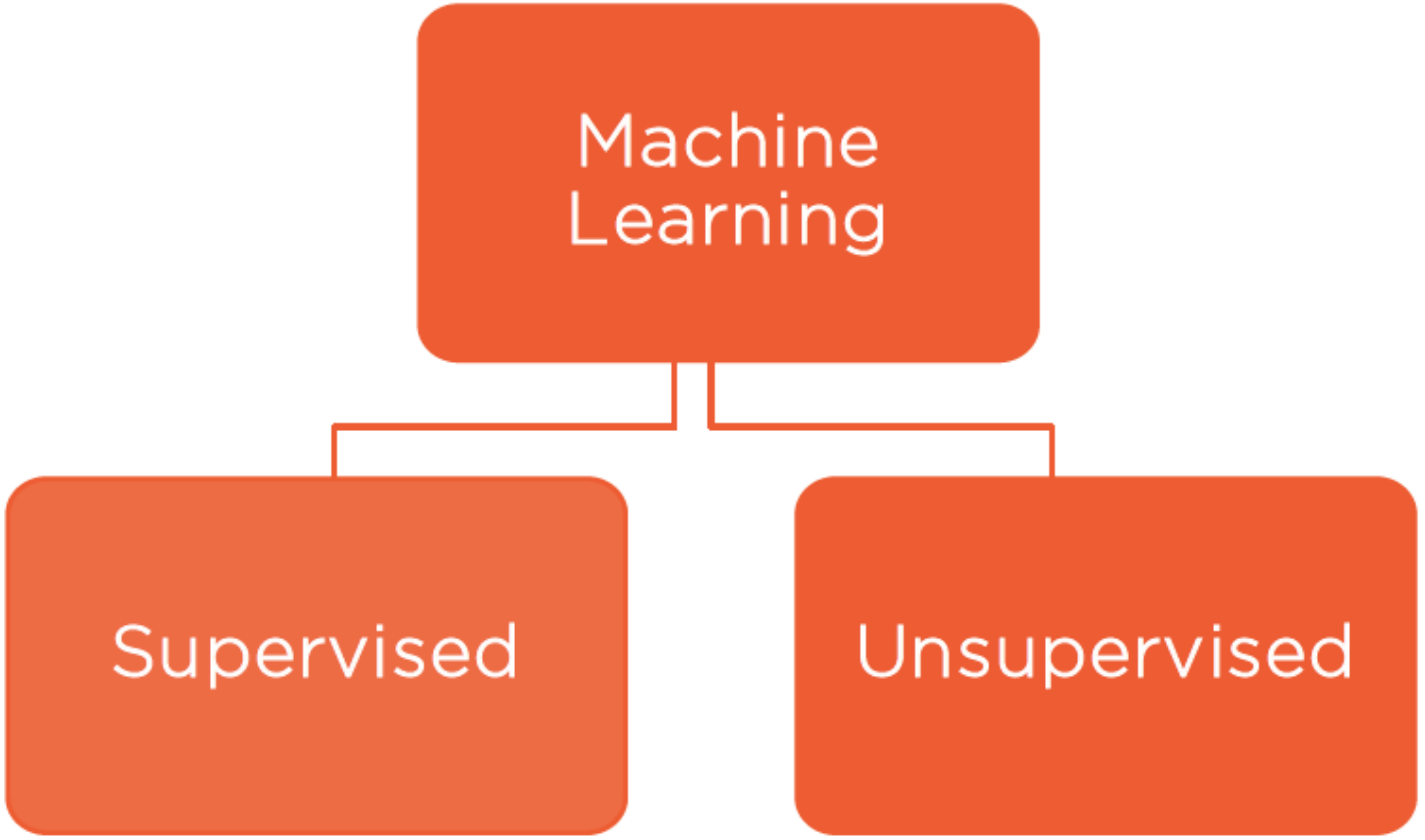


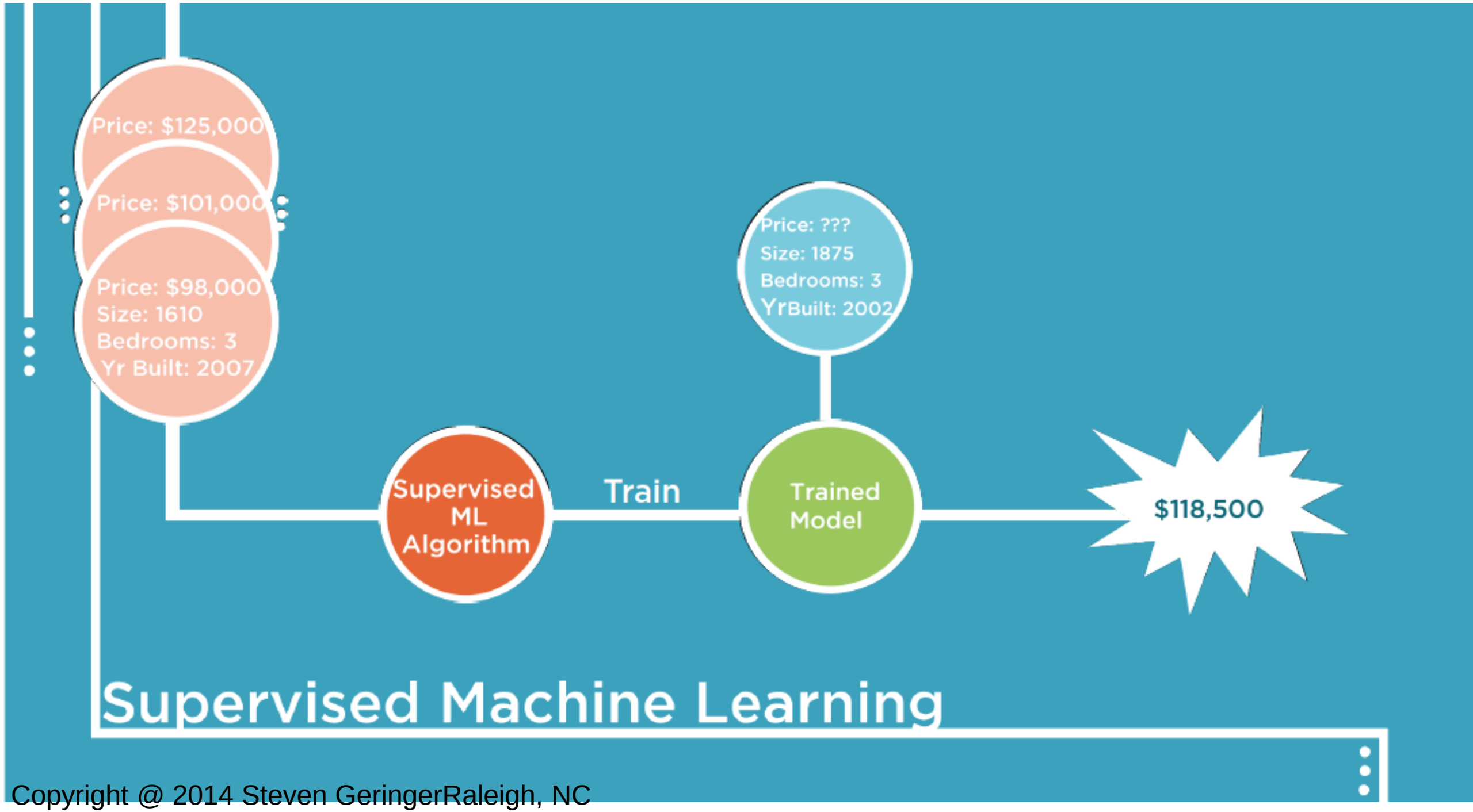
Machine Learning

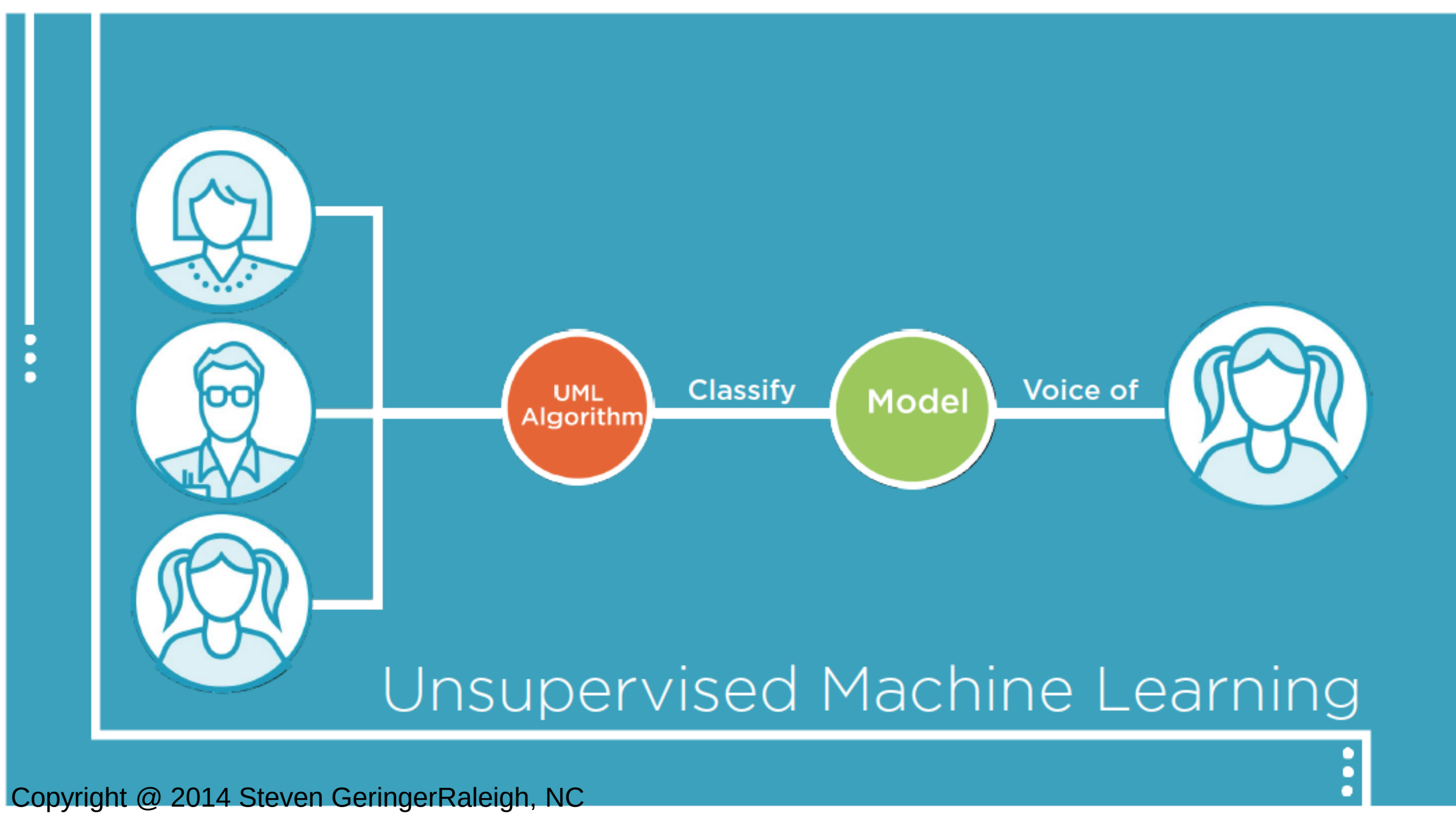
What is Machine Learning

Building a model from example inputs to make data-driven predictions vs. following strictly static program instructions

Types of Machine Learning







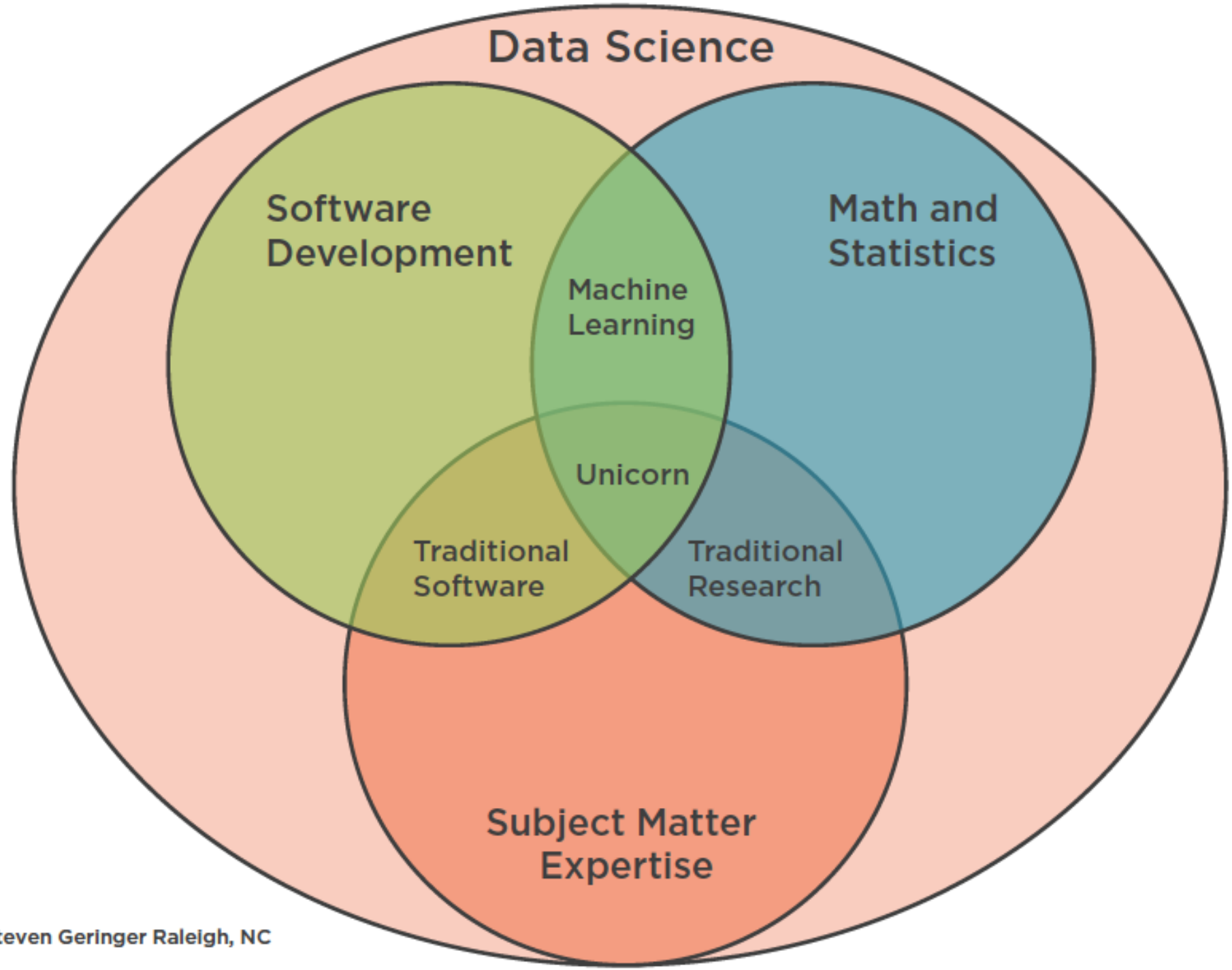
Machine Learning Technique Comparison

Supervised:

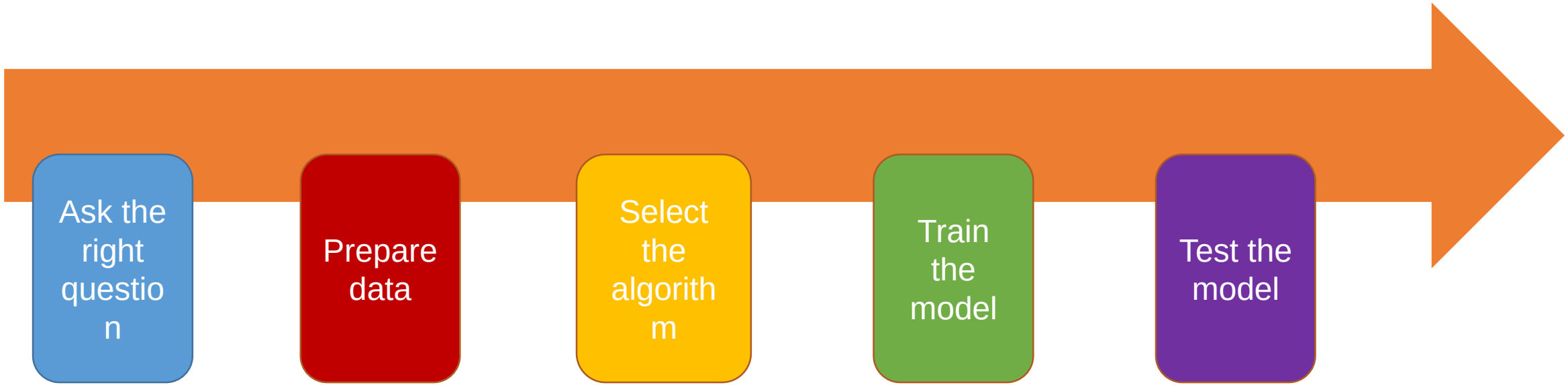
- Value prediction
- Needs training data containing value being predicted
- Trained model predicts value in new data

Unsupervised:

- Identify clusters of like data
- Data does not contain cluster membership
- Model provides access to data by cluster



Understand Project Workflow



Ask the Right Question

“Predict if a person will develop diabetes”

- **We need statement to direct and validate work**
- **Define end goal, starting point, and how to achieve goal**
  - what is the scope (including data sources)?
  - what is the target performance?
  - what the context for usage?
  - how solution will be created?

Ask the Right Question

“Use the Machine Learning Workflow to process and transform Pima Indian data to create a prediction model.

This model must predict which people are likely to develop diabetes with 70% or greater accuracy”

Prepare Data

1. **Find the data we need**
2. **Inspect and clean the data**
3. **Explore the data**
4. **Mold the data to Tidy data**

Prepare Data - Tidy Data

Tidy datasets are easy to manipulate, model and visualize, and have a specific structure:

- each variable is a column
- each observation is a row

Prepare Data

70% - 80% of a Model Build project is spent getting, cleaning, and organizing data

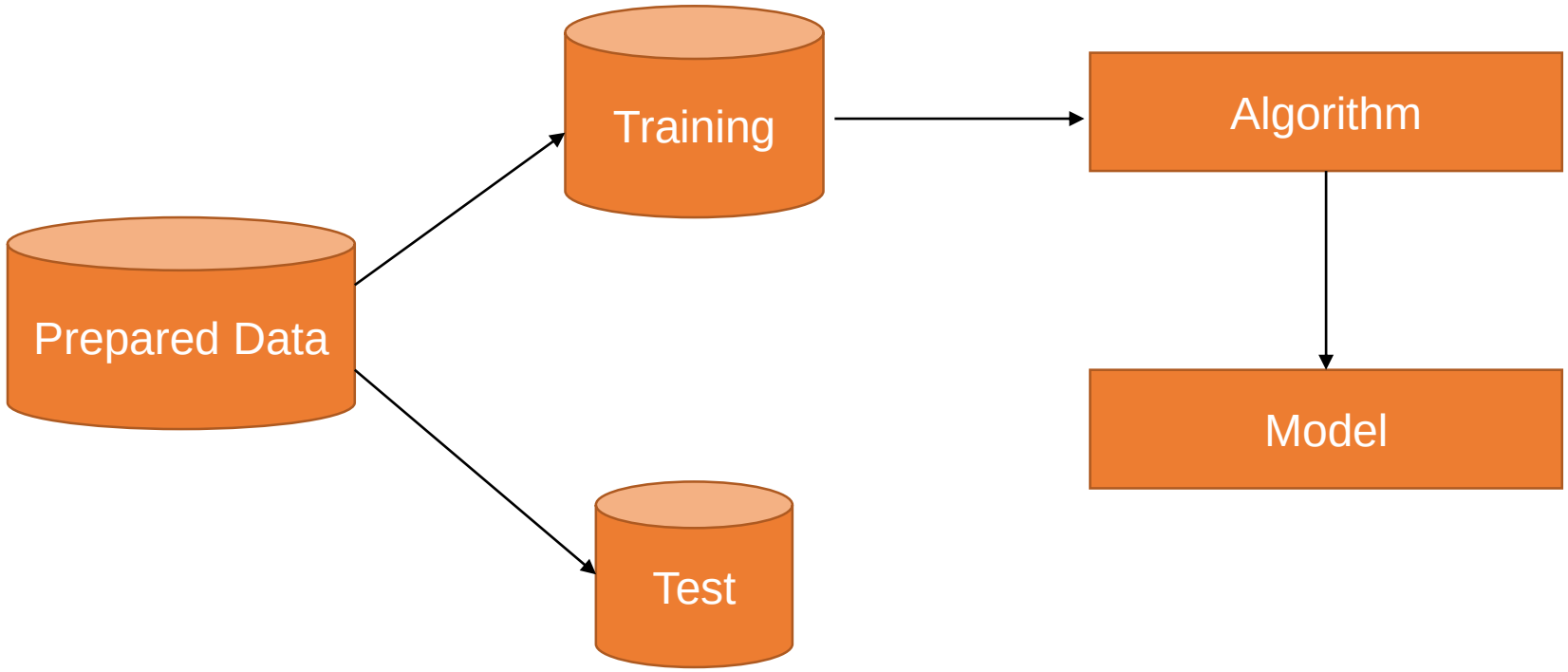
Select the algorithm

Perform algorithm selection:

- Use solution statement to filter algorithms
- Discuss best algorithms
- Select one initial algorithms

Train the Model

Let specific data teach a Machine Learning algorithm to create a specific prediction model



Test Model's Accuracy

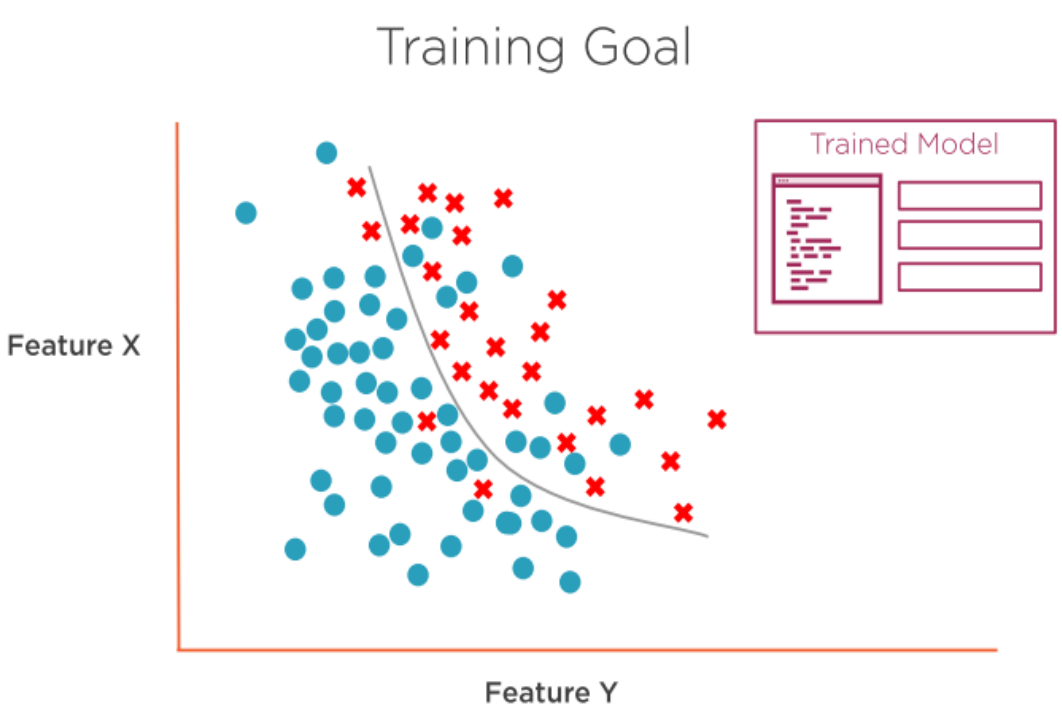
- Evaluate the model against test data
- Interpret results
- Improve results

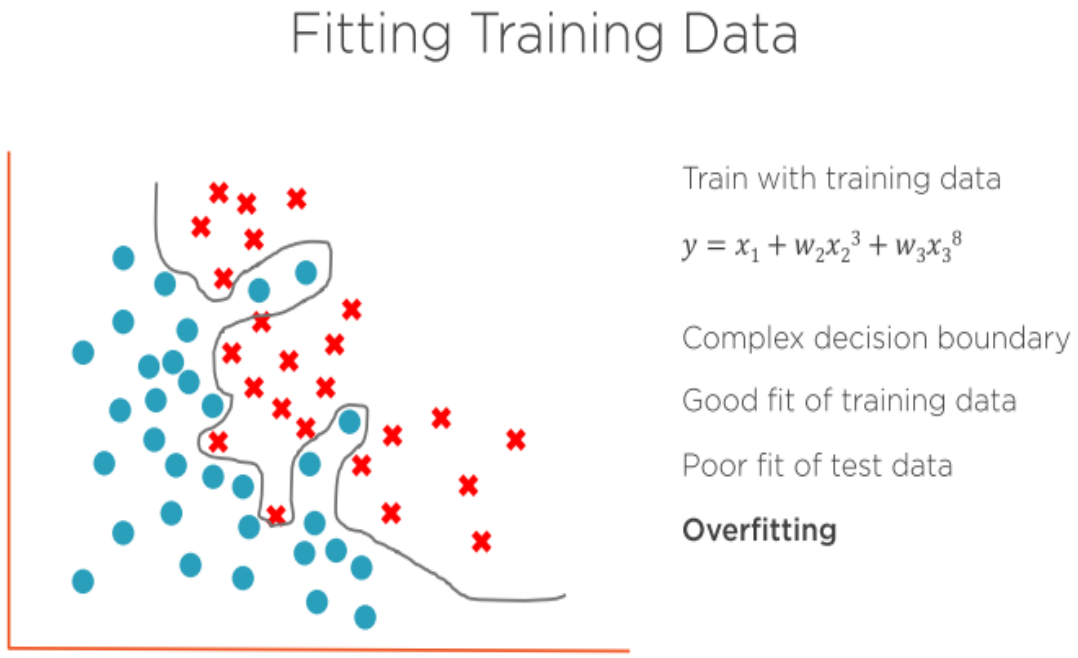
Performance Improvement Options:

- Adjust current algorithm
- Get more data or improve data
- Improve training
- Switch algorithms

Why Retain Test Data?

- New data => Better predictions
- Verify training performance with new data





How to deal with overfitting?

- The simplest strategy for correcting the overfitting is to holdout a portion of the development for assessment
- Large differences between the performance on the training and test sets usually indicate overfitting