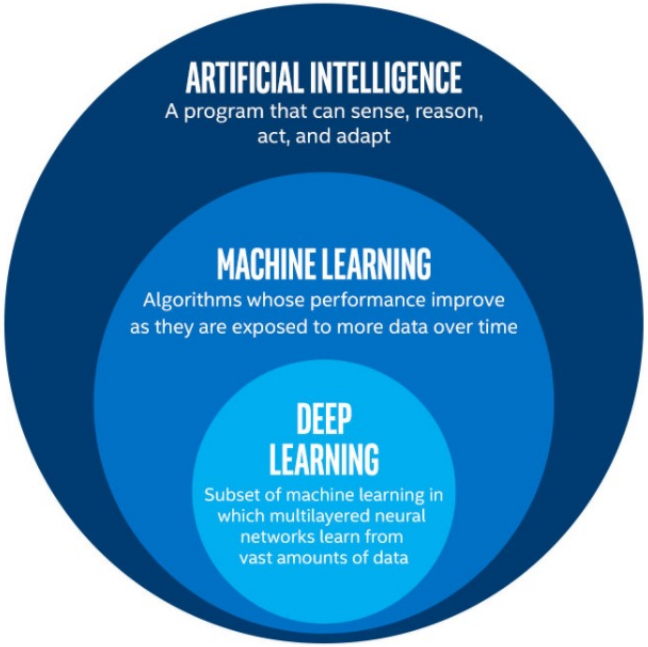
HUGS2021: Data Science (1)



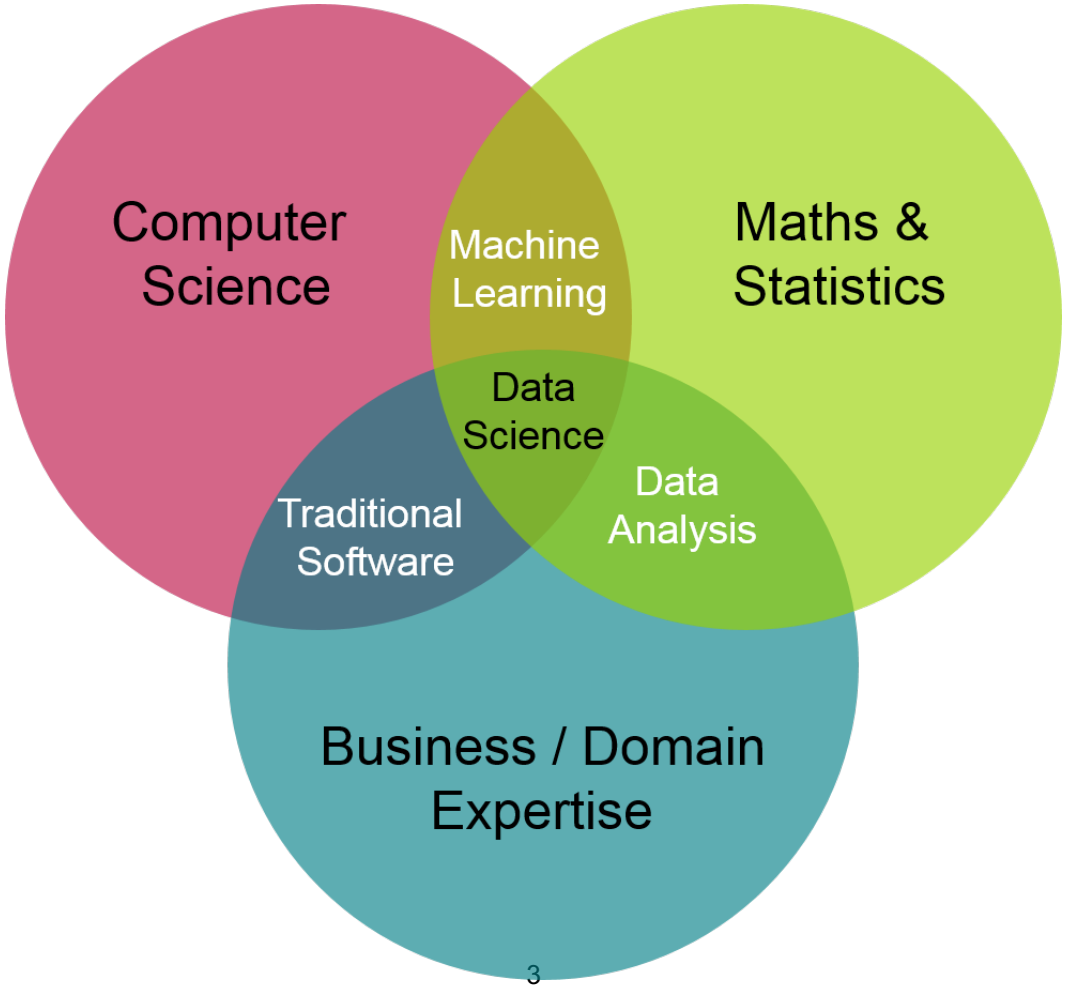
Malachi Schram, Ph.D.
Department of Data Science
Thomas Jefferson National Accelerator Facility

Goal of this talk

- Provide a high level overview of some key concepts
  - Data science and machine learning are massive research areas and there is no way to cover a fraction of these topics in a few lectures
- Provide some resources to get you started
  - Python centric ... sorry
- Cover some terminology
- Hopefully get you excited ☺

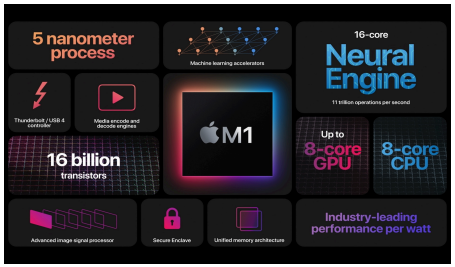
What is Data Science?

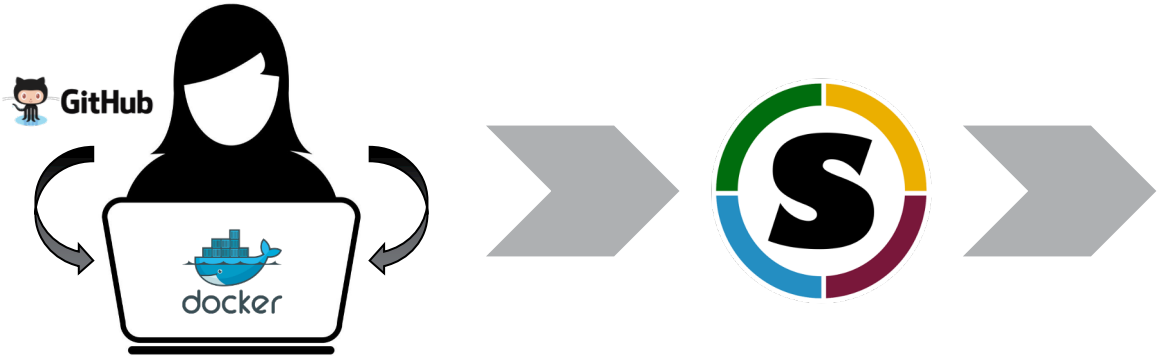
- Interdisciplinary field that leverages computer science, mathematics, and domain expertise to extract knowledge and insights from data
- Collaborative effort built on teams of experts



Computing Resources

- A lot of data science can be done on a modern laptop
  - Data preprocessing and some visualization
  - A large number of machine learning algorithms can efficiently run on your laptop
  - Deep learning will usually require bigger computing machines
- Developing a workflow that is portable:
  - Containers (https://www.docker.com/)
  - Singularity (https://sylabs.io)

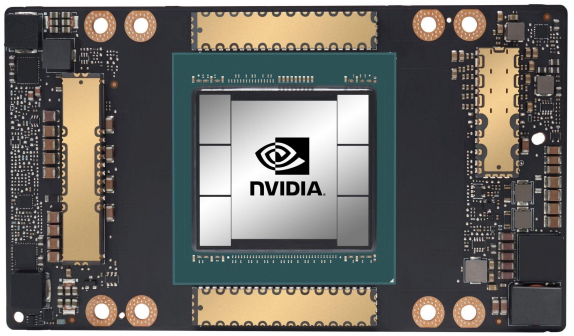




Computing Resources

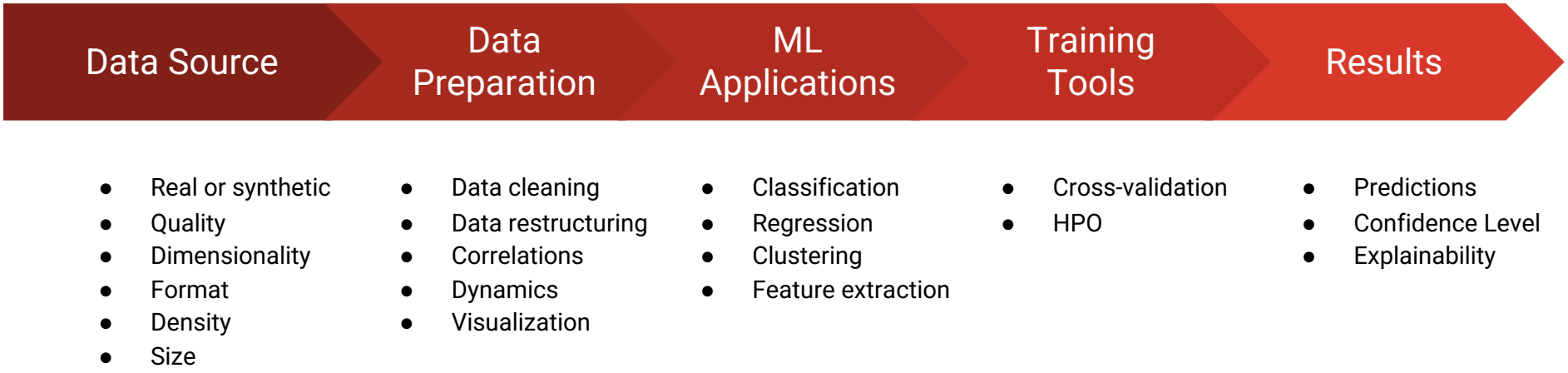
- There are some great tools to quickly test ideas and develop prototypes, such as:
  - Jupyter notebooks (https://jupyter.org/)
  - Google collaboration: (https://colab.research.google.com)

- For some studies, the data will be very large and the model with require some machines with GPU.
  - Regional HPC centers
  - Cloud resources
  - DOE LCF INCITE Proposal (https://www.doeleadershipcomputing.org/)
  - DOE LFC ASCR Leadership Computing Challenge
  - DOE LFC Director's Discretion



An example of a data science pipeline

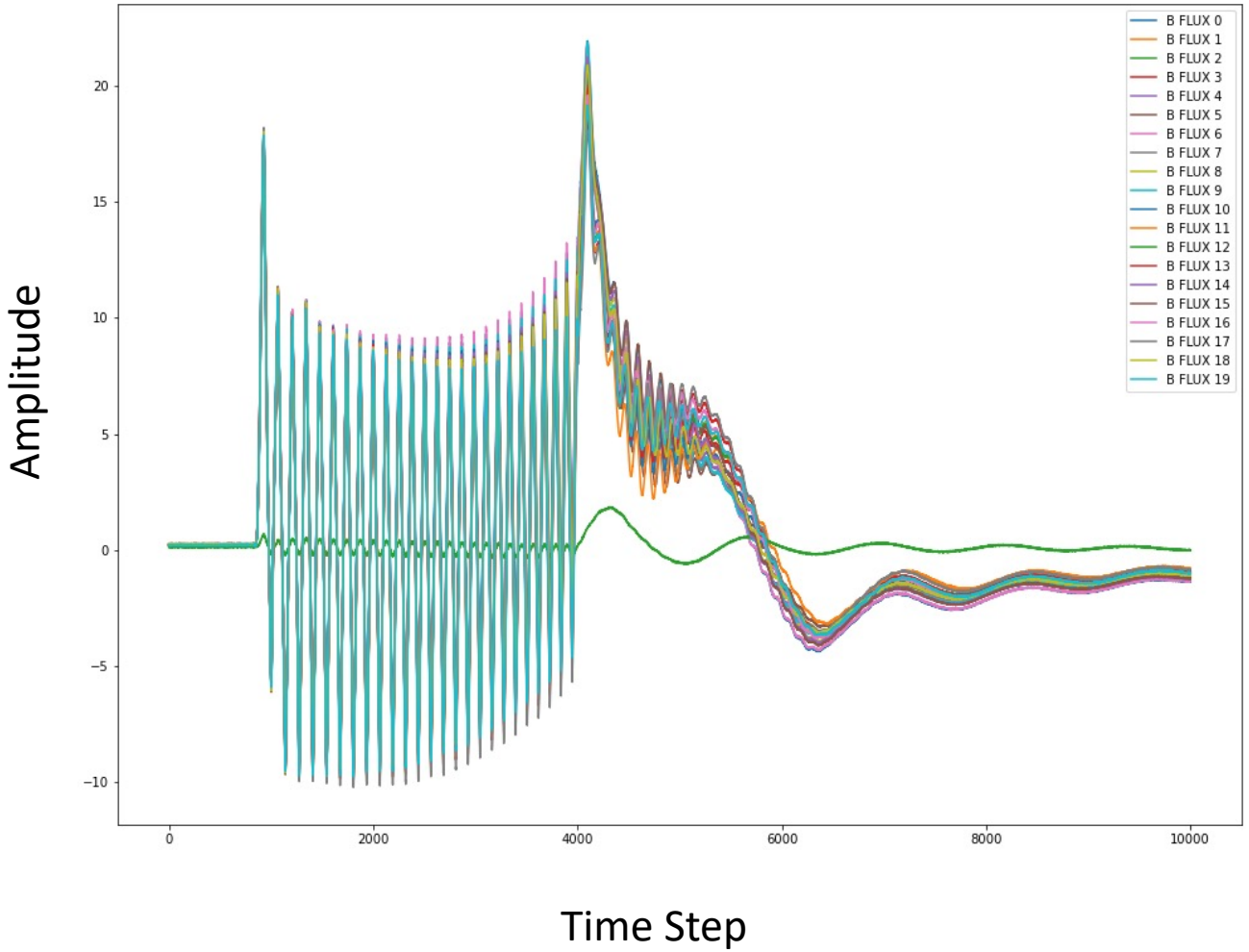
- What questions are we trying to answer with the data?
- Do we have the right data?
- What do we know about the data?
- Can we learn something from the data before using machine learning (ML) techniques?



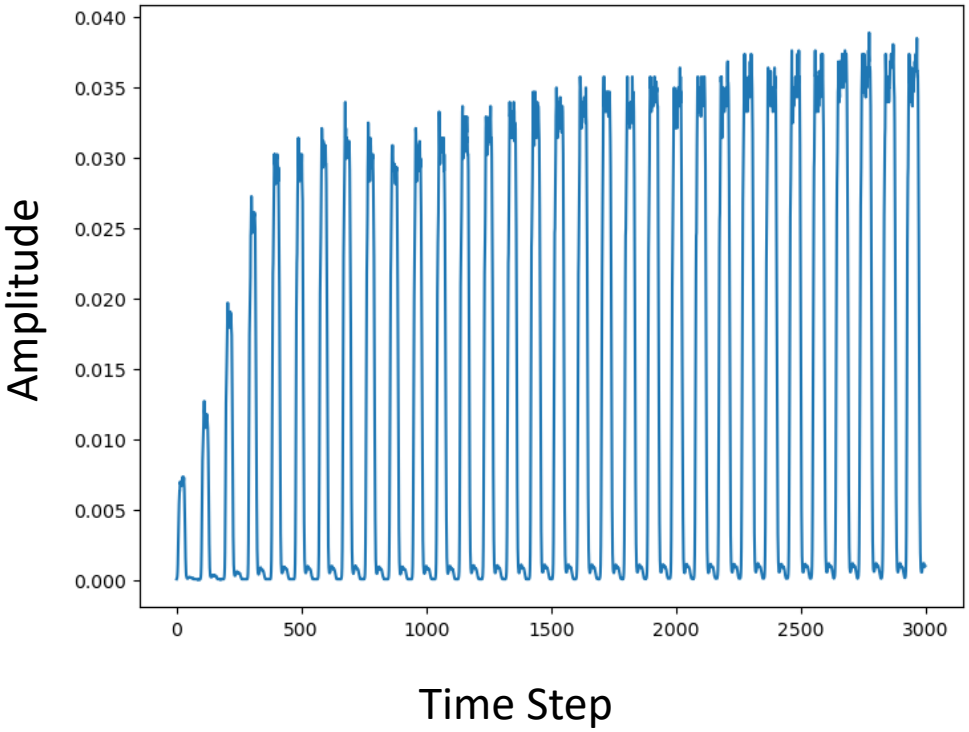
Data Source

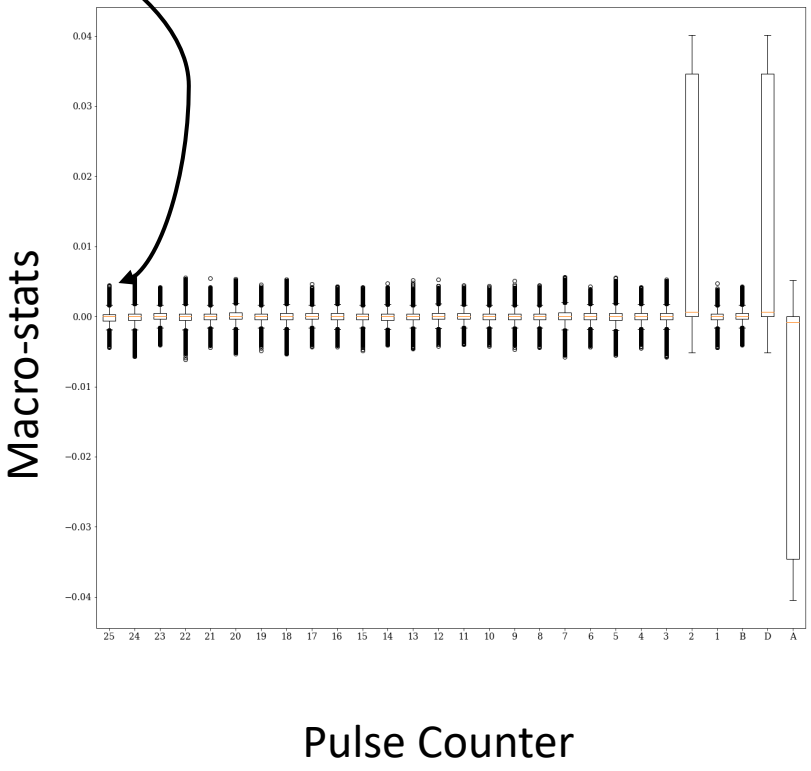
- How was the data collected and labeled?
  - Real world data is messy!
  - It will have missing/noisy data that you will need to account for.
- How was it curated?
  - Data curation is the organization and integration of data collected from various sources.
  - Do the various sources of data need to be temporally aligned?
- What are the data formats for your study?
  - Images (cats and dogs), temporal (time series weather data), categorical (ex: labels A-Z), ordinal (ex: ranking between 1-5)
- What is the dimensionality of the data sources?
  - High dimensional (ex: images)
  - Low dimensional (ex: single variable sensor)
- How many samples do you have?
  - Large number of samples (>10k): Google images or large time series data
  - Limited: A few experimental measurements and/or simulation samples
- Does the data capture the dynamics (physics) of interest or are they distinct samples?
- What are the input and output features of interest?

Finding problems with labeled data (1)

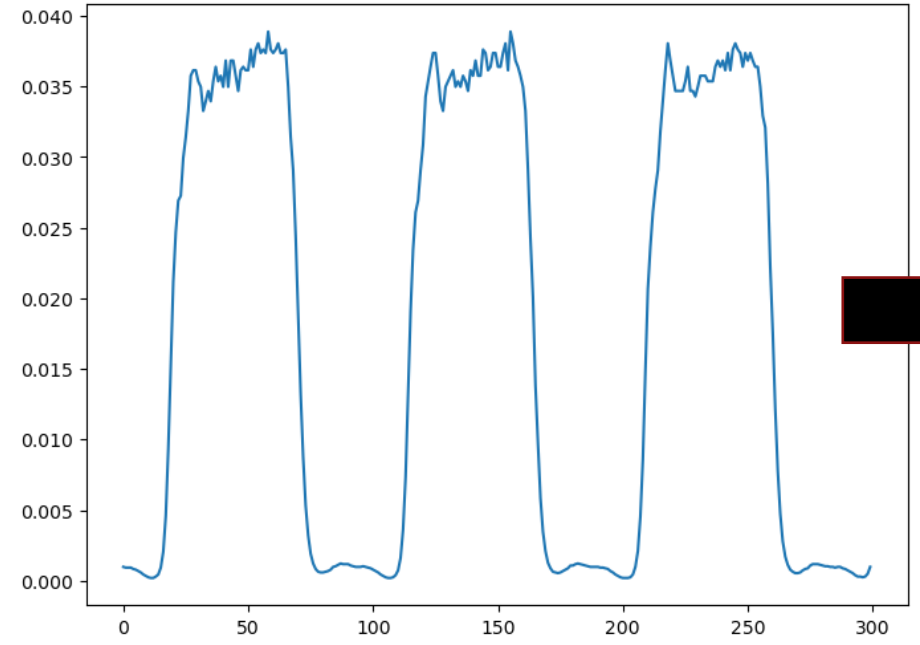


Finding problems with labeled data (2)

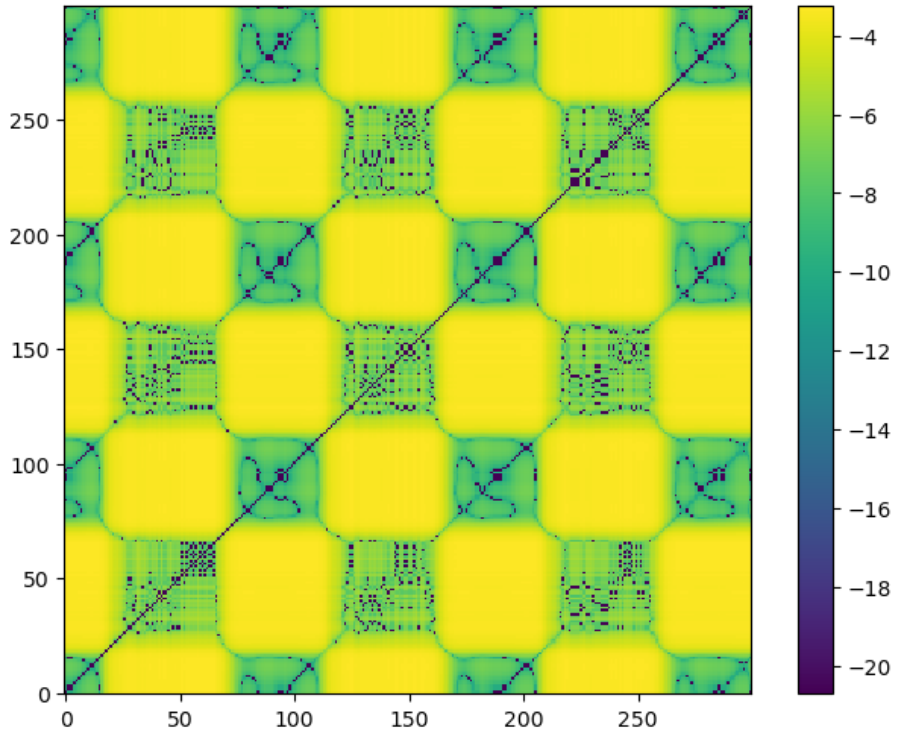




Look at your data in a different way



Time Step



Time Step

Some useful Python data packages

- Numpy (https://numpy.org/):
  - A library that supports large multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- Pandas (https://pandas.pydata.org/):
  - A fast, powerful, flexible, and easy to use open source data analysis and manipulation tool.
- Dask (https://dask.org/):
  - Provides advanced parallelism for analytics, enabling performance at scale for the tools you love (numpy and pandas)
- Scikit-image (https://scikit-image.org/):
  - A collection of algorithms for image processing

Some useful visualization packages

- Matplotlib (https://matplotlib.org/):
  - A comprehensive library for creating static, animated, and interactive visualizations in Python.
- Seaborn (https://seaborn.pydata.org/):
  - Seaborn is another data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- Bokeh (https://bokeh.org/):
  - Bokeh is used for creating interactive visualizations for modern web browsers.

Data preparation for ML

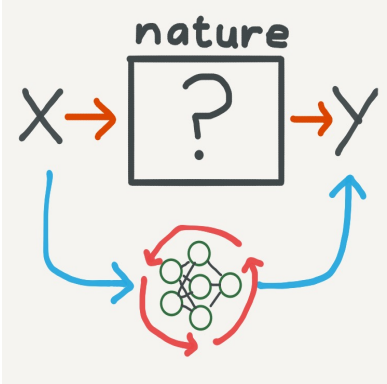
- Normalizing your data:
  - The purpose of normalizing your data is to provide the data with a common scale. This is particularly important when dealing with multiple input variables as it will effect the relative contribution from each variable when calculating the cost function.

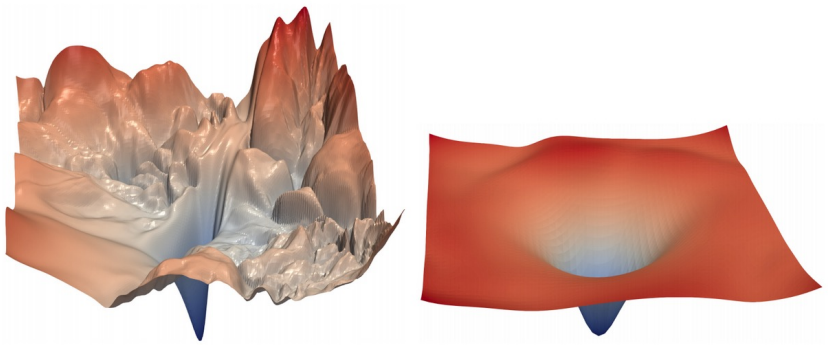
- Reformatting your data:
  - You will need to frequently reformat your data in order to satisfy the input requirements of your model architecture.
  - As an example, a time series model will typically require the full trace to be restructured into smaller sequential traces with a defined look-back and look-forward setup.

- Data consideration for multi-modal models:
  - In some situations you will want to combine different input data types (video, sense traces, etc.) into a hybrid predictive model. You will need to resample the data to ensure that the data sources are temporally aligned and normalized to simplify merging the combined cost functions.

Machine Learning Applications: Building a ML model

- What do we want from our models:
  - Provides a transformation between input and output data
  - It should be generalizable
    - Does the model apply to an orthogonal dataset?
    - Does the model capture the fundamental transformation?
  - Explainable
  - Stability and guarantees

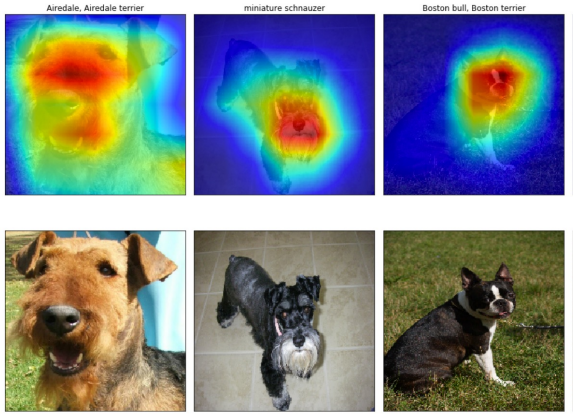




(a) without skip connections (b) with skip connections

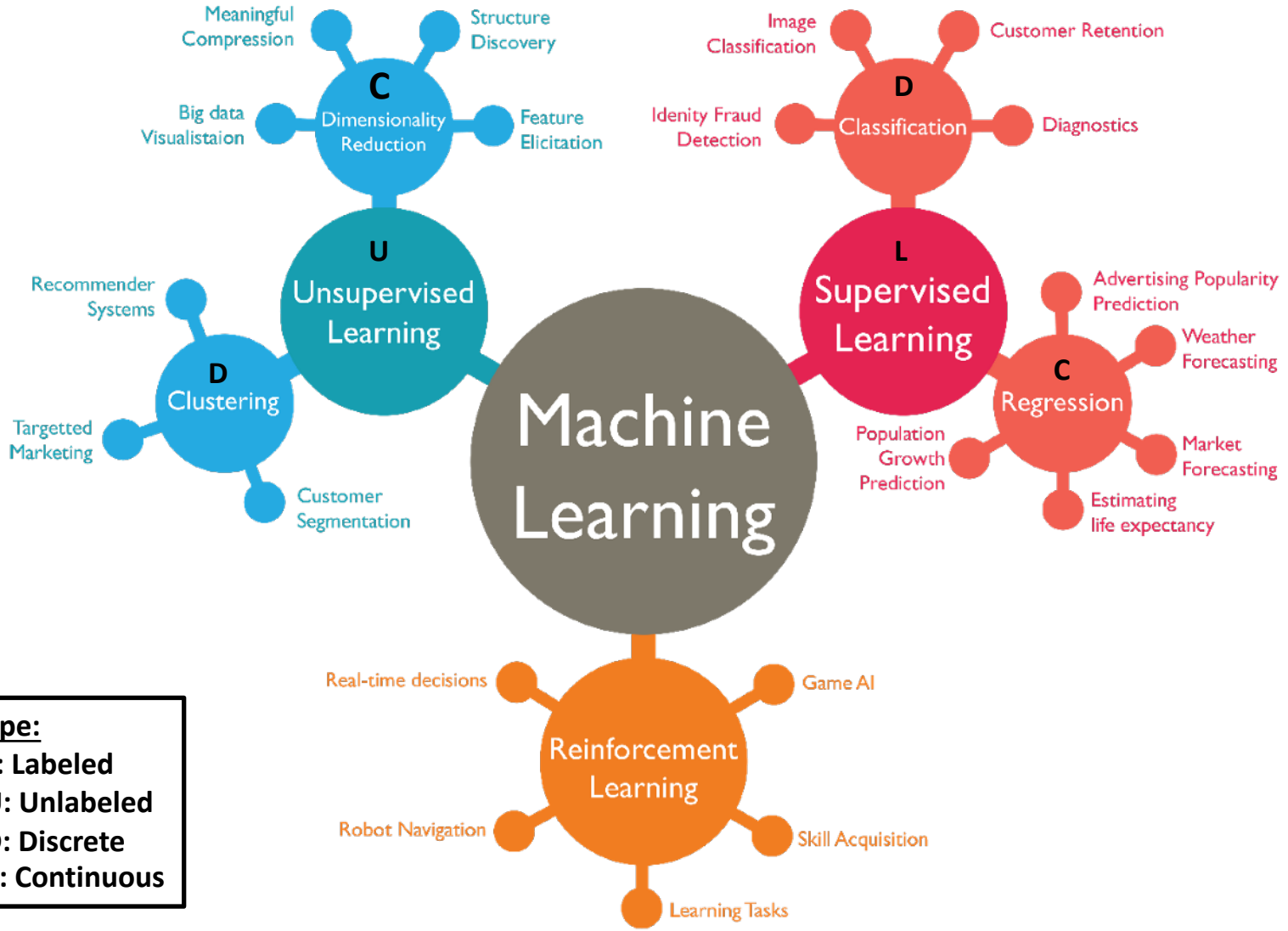
Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

arXiv:1712.09913



arXiv:1512.04150

Machine Learning Applications: Categories



Data Type:

- **L: Labeled**
- **U: Unlabeled**
- **D: Discrete**
- **C: Continuous**

Machine Learning Applications: Packages

The following are a few popular packages used to develop ML models:

- Sklearn (https://scikit-learn.org):
  - Simple and efficient tools for predictive data analysis
  - Includes a large variation on machine learning models and pre-processing techniques
- Keras/Tensorflow (https://www.tensorflow.org/):
  - A popular open source machine learning platform developed by Google
- Pytorch (https://pytorch.org/):
  - Another popular open source machine learning platform
  - Provides a number of extensions

Machine Learning Applications: Example of Algorithms

- Supervised:
  - Gaussian processes
  - Support Vector Machine
  - Neural Networks
  - Random Forest

- Unsupervised:
  - K-mean clustering
  - Principal component analysis (dimension reduction)
  - Auto-encoders (dimension reduction)

Training Tools: Cross-validation

- There are a few common techniques used to prepare the available datasets for training and validating ML models.
- The easiest method is to split your data as follows:
  - Training data (80%) of which 20% of it is used for validation
    - Validation samples are used to evaluate the performance and any potential overfitting
  - The remaining 20% of the data is used to test the model
- Creating these orthogonal data samples is an integral component of validating your model.
- A more advanced technique is K-fold cross-validation.
  - Randomly shuffle k-groups from the data to create training and testing samples
- Be careful to understand if you are interpolating or extrapolating.
  - Some systems are non-stationary and the historical data used for training the model might not match the current or future system condition
    - Building a physics based model generally provides better prediction and guarantees that the prediction is accurate

Training Tools: Model and hyper-parameter optimization

- When developing ML models there are inevitably several parameters that are configurable and can be used to optimize the model.

- For example, you can change the:
  - learning rate
  - number of layers
  - dropout rate
  - kernel
  - loss function
  - etc.

- There are several packages available to explore the large parameter space. For example:
  - MLFlow (https://mlflow.org/):
    - Platform to manage the ML lifecycle which includes HPO
  - Keras Tuner (https://www.tensorflow.org/tutorials/keras/keras_tuner):
    - Library that helps optimize hyperparameters for TensorFlow models

Malachi Schram, Ph.D.
Department of Data Science
Thomas Jefferson National Accelerator Facility
schram@jlab.org

- Upcoming Computing Trends in Nuclear Physics Talks:
  - Data Science 2-3 (Malachi Schram)
    - Examples of machine learning models:
      - Forecasting, few-shot learning, hybrid models
    - Reinforcement learning session

- New data science position at JLab:
  https://careers.peopleclick.com/careerscp/client_jeffersonlab/external/jobDetails.do?functionName=getJobDetail&jobPostId=1910&localeCode=en-us