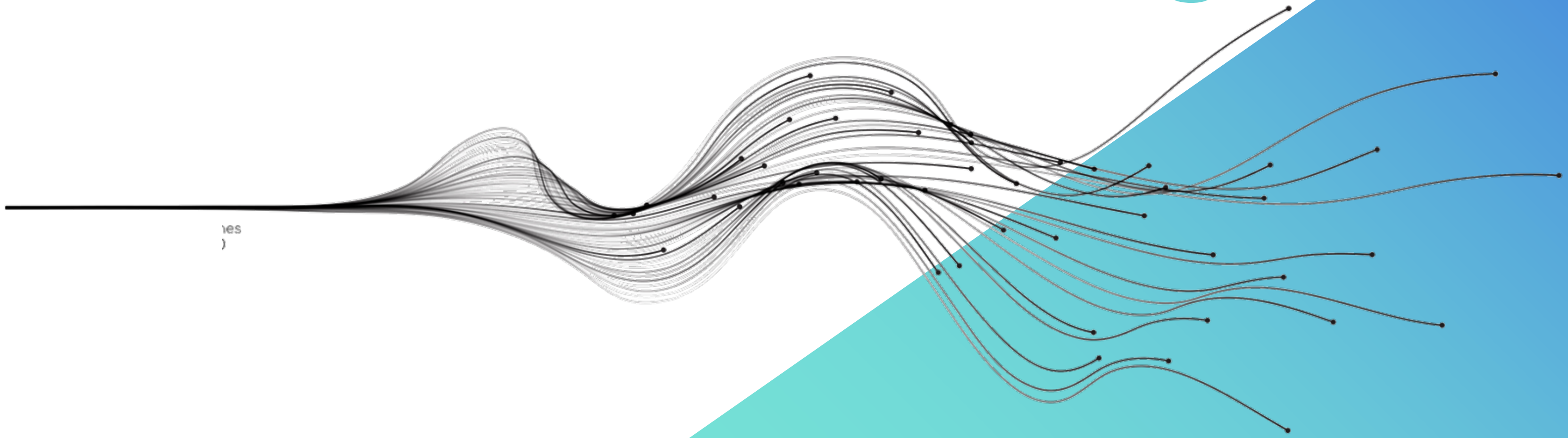
DSP
Concepts

Introduction to Machine Learning



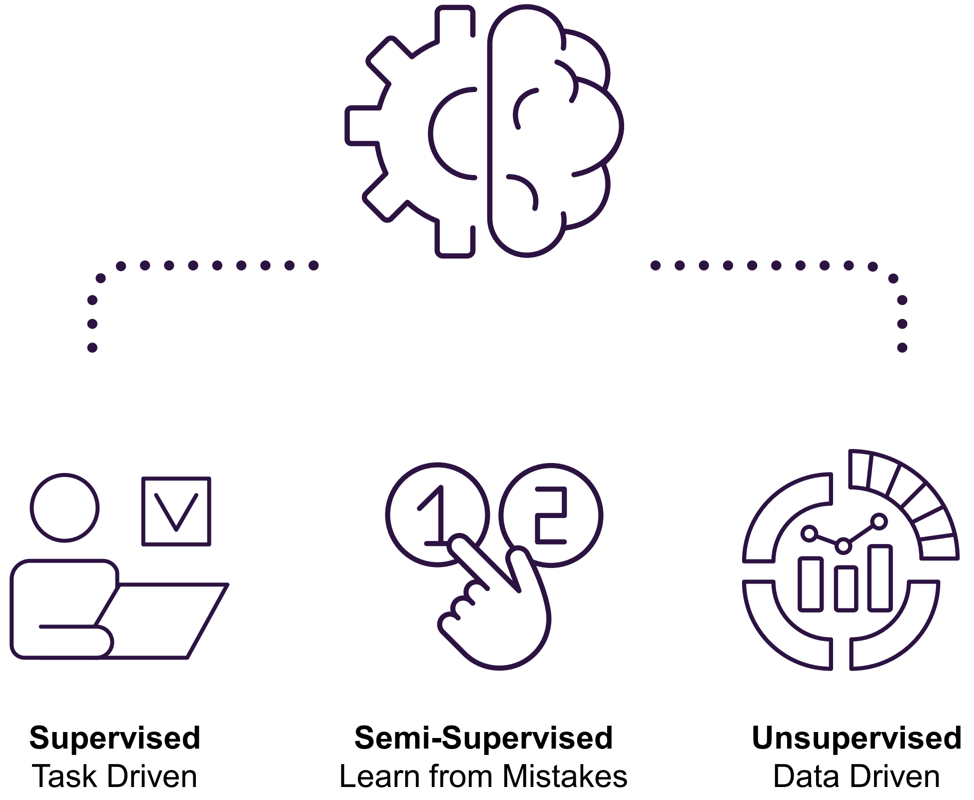
What is Machine Learning?

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.

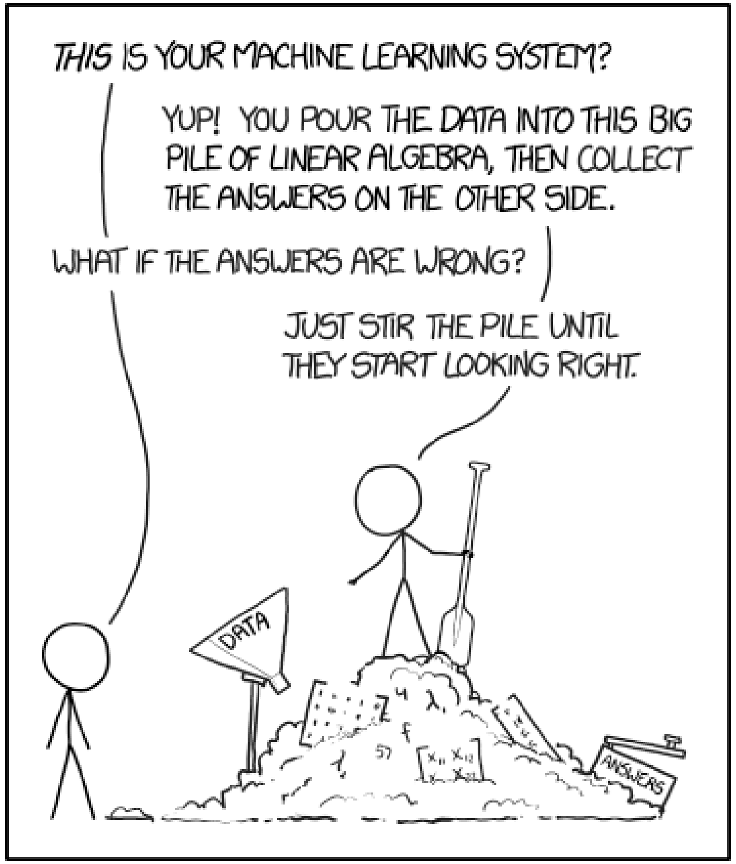
[Wikipedia Article for Machine Learning]

Types of ML

- Unsupervised (e.g. clustering)
- Semi-supervised
- Supervised



What is the process of Supervised Learning?

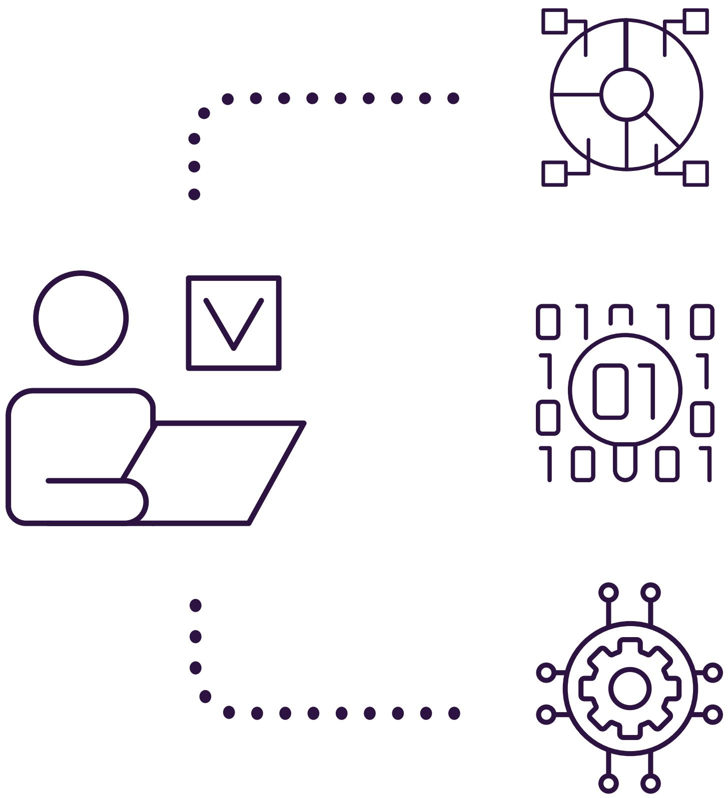


1. Get data
    1. Evaluate
    2. Clean
    3. Label
2. Identify Relevant Metrics
3. Featurize data
4. Select/build Model
5. Train Model
6. Evaluate Model
7. Get data
    1. Testing on device
    2. Deploy
8. Repeat relevant steps as needed

DSP Concepts

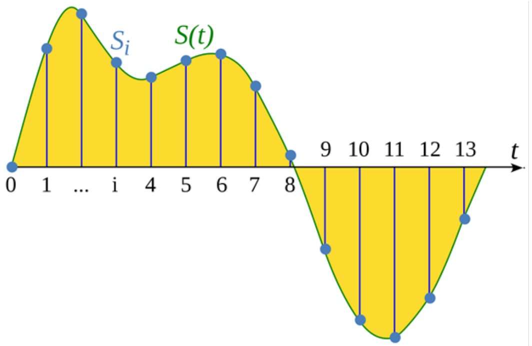
What is the process of Supervised Learning?

1. Get data with labels
2. Featurize data
3. Train Model

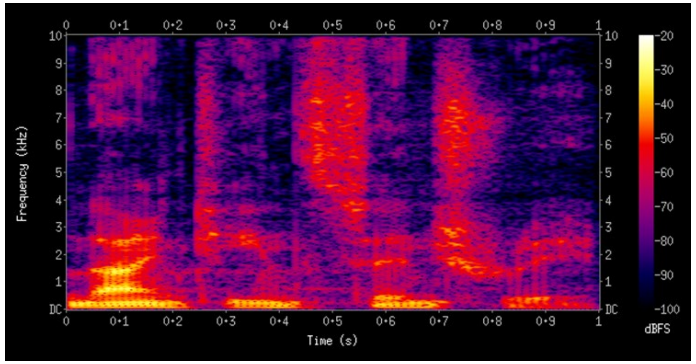


What are Features?

In machine learning and pattern recognition, a feature is an individual measurable property or characteristic of a phenomenon. Choosing informative, discriminating and independent features is a crucial element of effective algorithms in pattern recognition, classification and regression.



https://en.wikipedia.org/wiki/Sampling_(signal_processing)



https://upload.wikimedia.org/wikipedia/commons/c/c5/Spectrogram-19thC.png

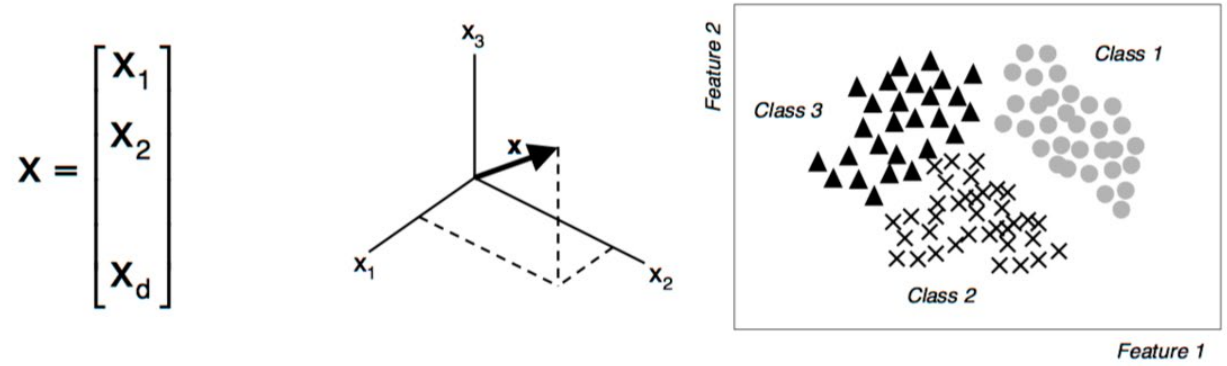
[Wikipedia Article on Feature (Machine Learning)]

DSP Concepts

Leveraging Features

Your dataset describes a feature space

- Once processed into features, your dataset describes a high dimensional problem space
- Good dataset creation ensures that your dataset is representative of the real-world problem you are trying to solve
  - Shortcomings become “errors” in production



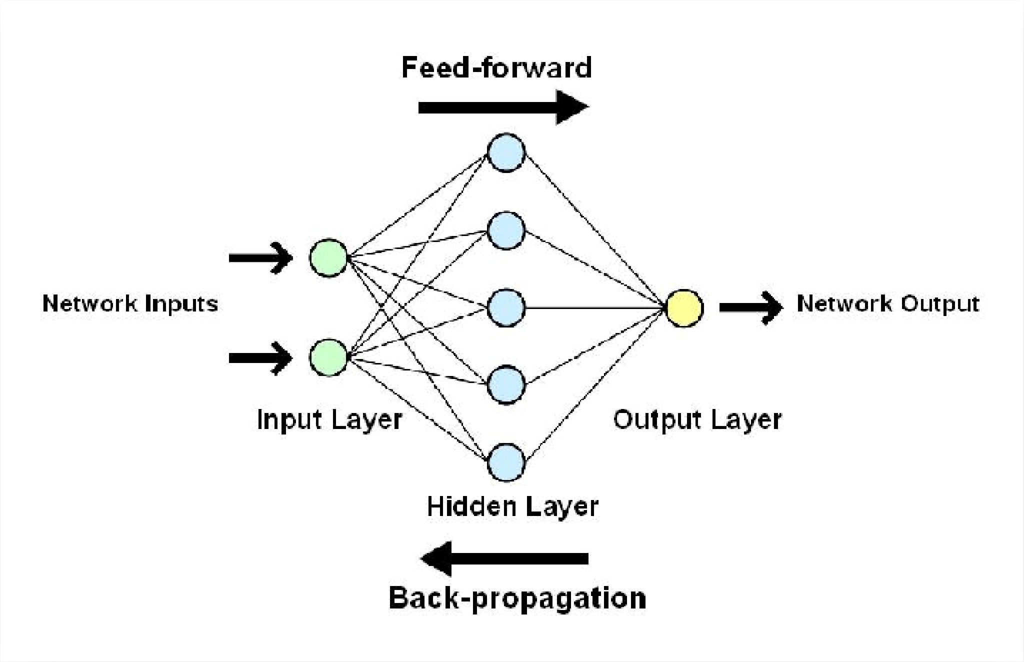
Feature vector | Feature space (3D) | Scatter plot (2D)

https://brilliant.org/wiki/feature-vector/

Training a Model

How does the model learn?

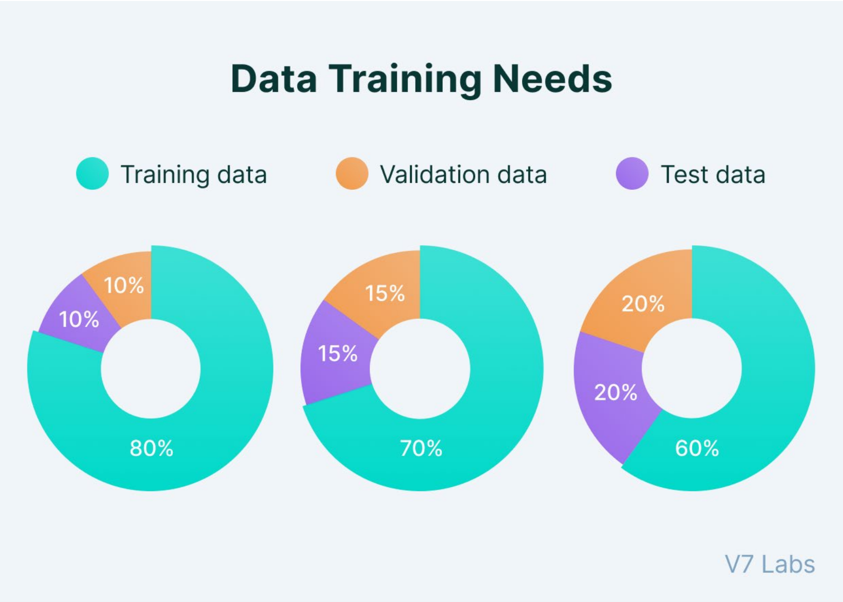
- A **loss function** is a differentiable equation that tells us how much error we have in our prediction compared to the ground truth label.
- Using the chain rule, we can calculate the gradient at each weight in our network and update those values. This is called **back propagation**.
- **Training** is doing this for many iterations of the dataset, called **epochs**, the model learns to map inputs to our desired outputs.



https://www.semanticscholar.org/paper/Application-of-Back-Propagation-Artificial-Neural-A-G%C3%BCll%C3%BC-Yilmaz/db39fd79bb591b04d33207992f6ccde03cabd861

Dataset Best Practices

In order to make sure we don't over-train, we split our data into subsets



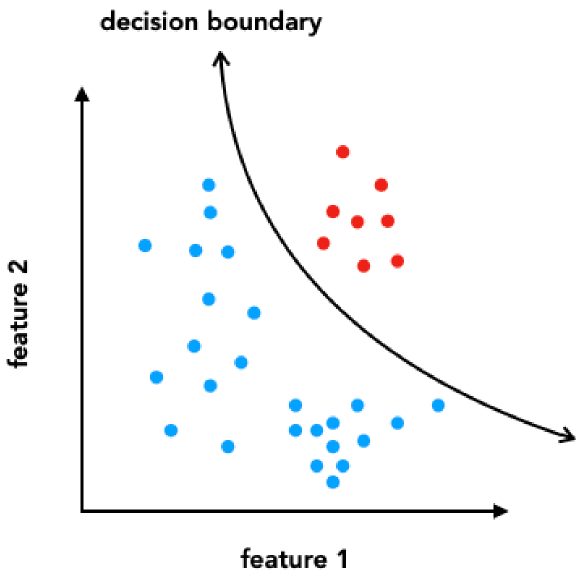
https://www.v7labs.com/blog/train-validation-test-set

- **Train:** Shown to the model during training. The model learns from this data.
- **Validation:** Hold out set during training to ensure model is generalizing well. Technically, we snoop on this data during training if the model didn't learn from it.
- **Test:** Final hold out dataset. Used as a final sanity check before releasing to production. This is the only decision this dataset should be used to make.

Model behavior can be much more difficult to predict than in traditional algorithm development. Good dataset practices are therefore crucial to understanding the performance of your model.

Learning the Feature Space

Your model learns it’s decision boundaries from the feature space



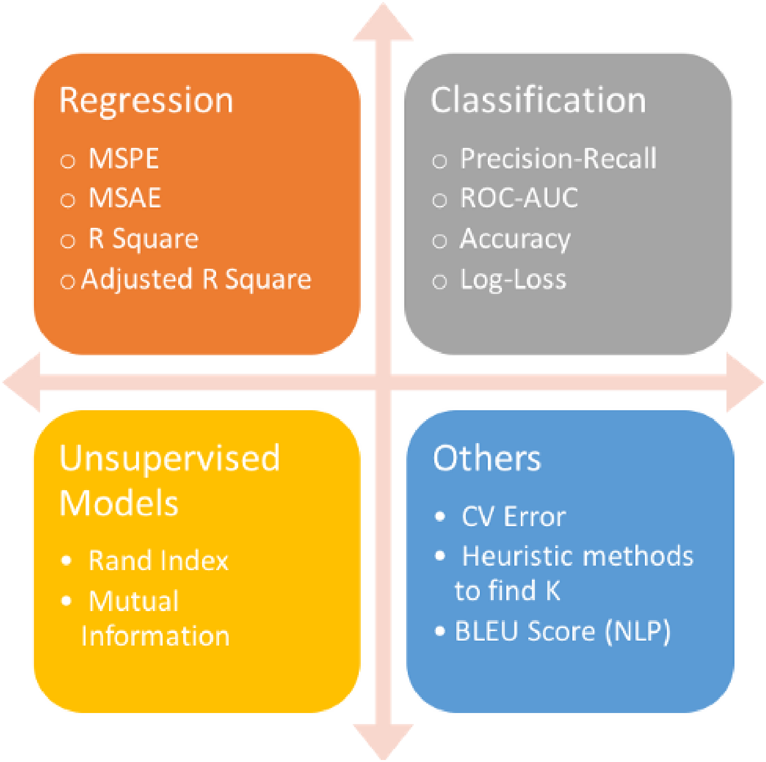
https://towardsdatascience.com/concept-learning-and-feature-spaces-45cee19e49db

- Models do not think. They are powerful correlation machines that learn to map inputs to outputs based on the data you showed them during training.
- It is possible to over-train your model and memorize your dataset.

Metrics

Understanding model performance

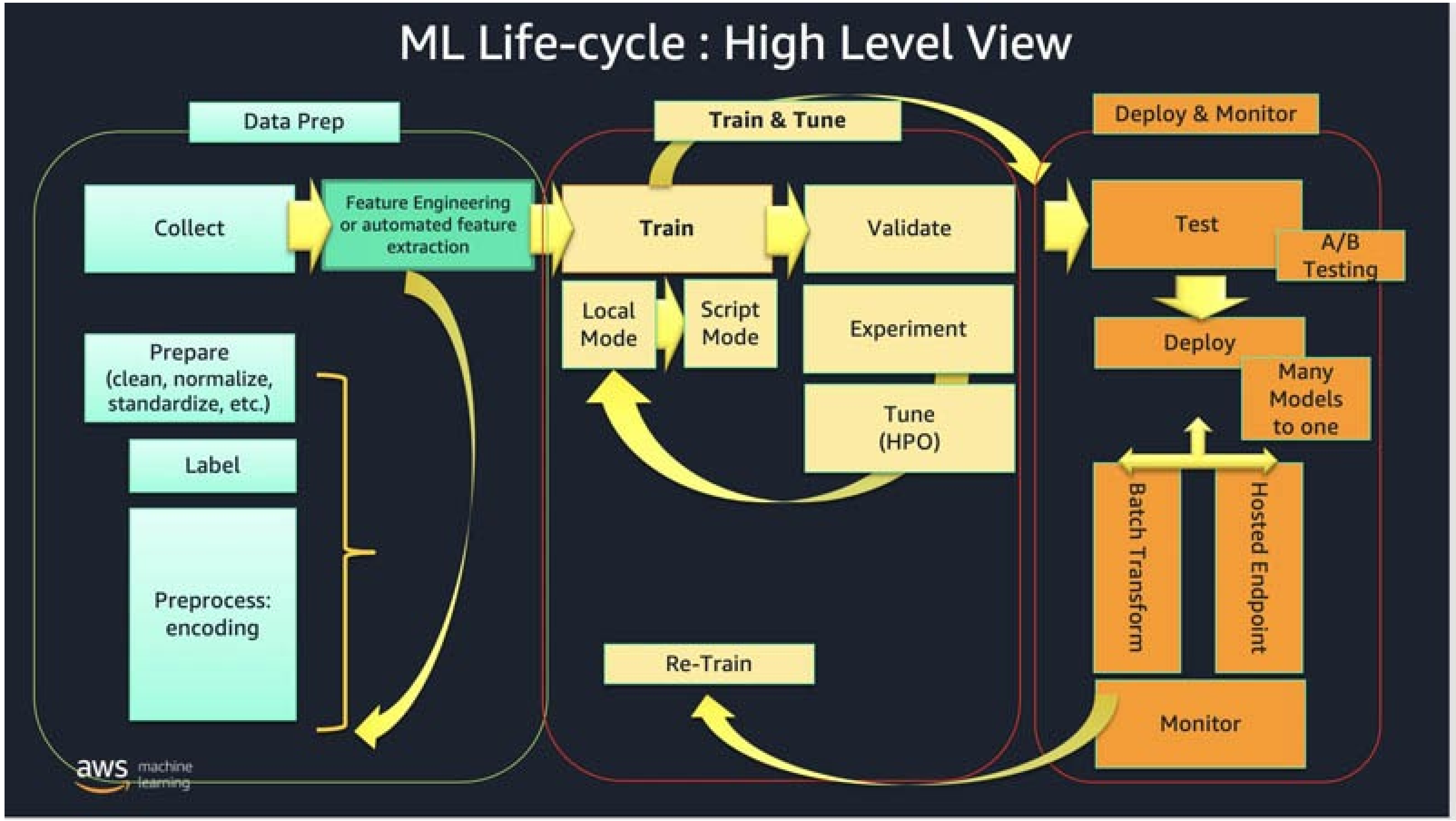
- Loss functions are not always the easiest to interpret or understand
- Often, a ML practitioner will have other ways of evaluating the model that are not directly used in training but can help with model selection
  - MOS Scoring, PESQ
  - F1, Precision, Recall



https://www.kdnuggets.com/2018/06/right-metric-evaluating-machine-learning-models-2.html

Architecture

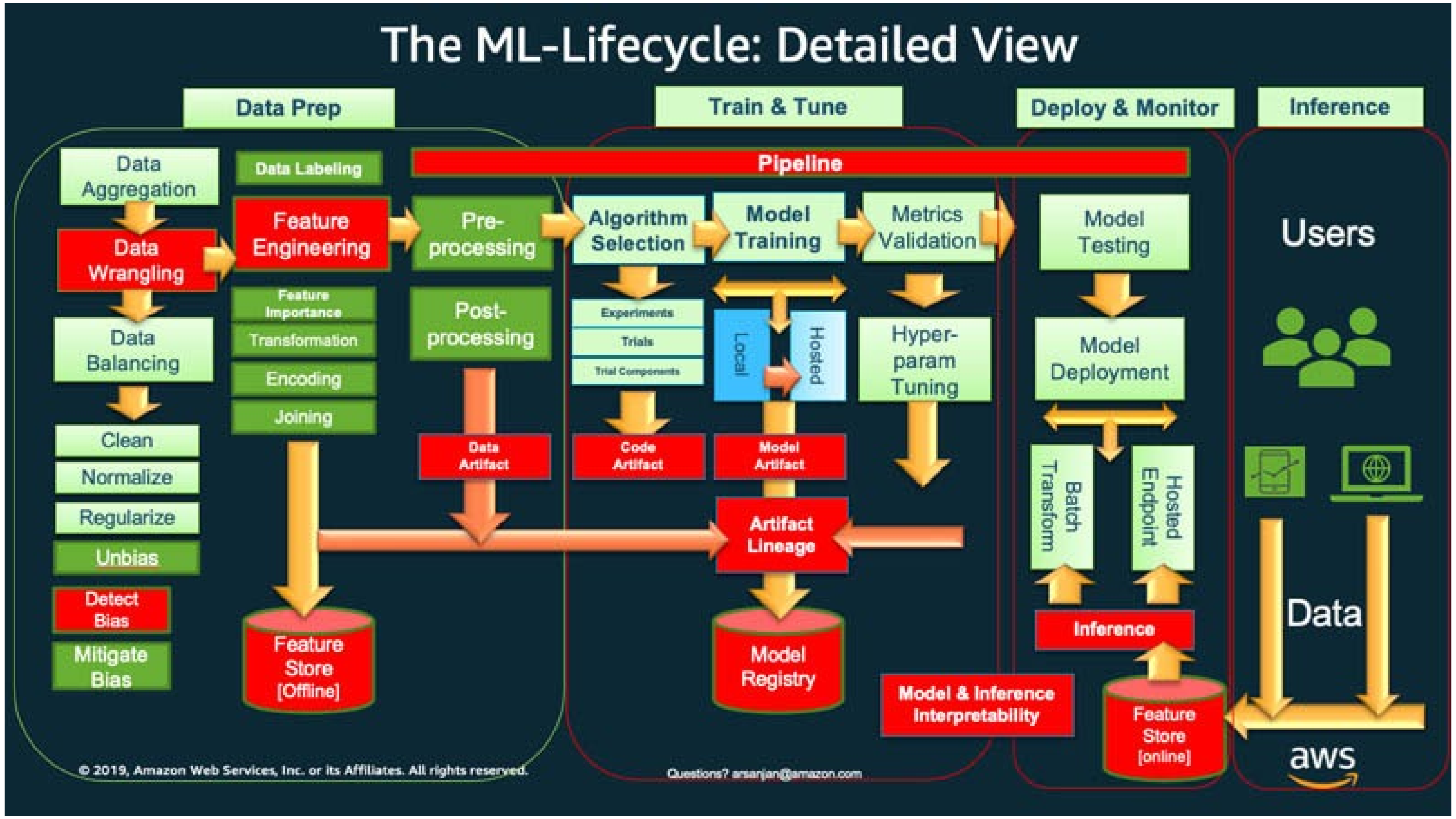
What Does ML Look Like?



DSP Concepts

Artifacts

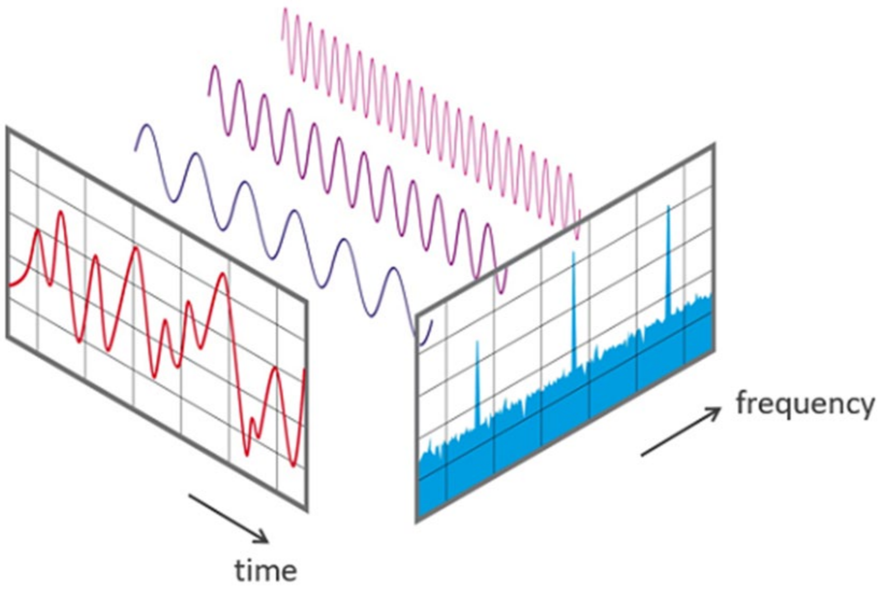
Keeping Track of all the Things



DSP Concepts

ML Features

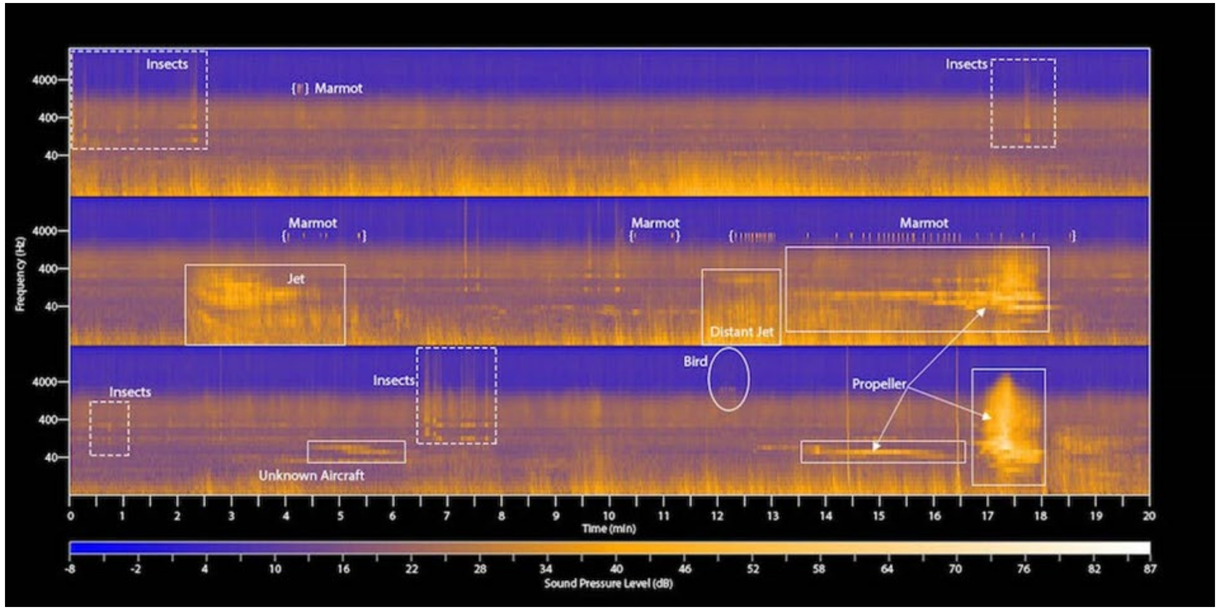
Fast Fourier Transform



https://commons.Wikimedia.org/wiki/File:FFT-Time-Frequency-View.png

- Shows frequency over time
- Linearly spaced frequency bins
- Can apply processing in the frequency domain and then use an inverse fast Fourier transform (IFFT) to get time domain audio

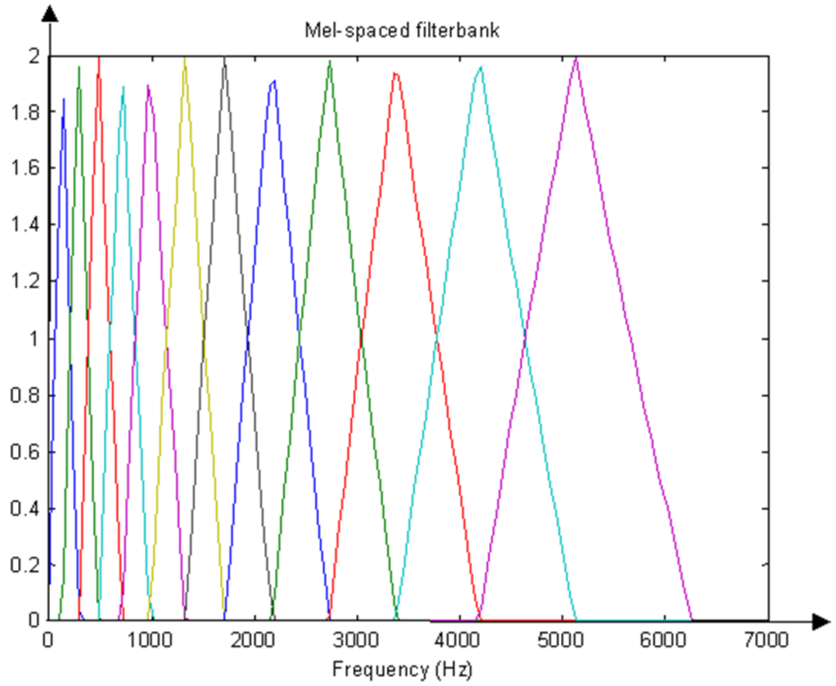
Power Spectrogram



https://upload.wikimedia.org/wikipedia/commons/9/99/Mount_Rainier_soundscape.jpg

- Built from short time Fourier transform
- Repeat Fourier transform with set window and hop size
  - short-time Fourier transform (STFT)
- Take magnitude squared of frequency bins
- Common for classification tasks

Mel Spectrogram



http://www.ifp.illinois.edu/~minhdo/teaching/speaker_recognition/speaker_recognition.html

- Triangular frequency windows
- Filter banks that attempt to approximate human hearing
  - Humans struggle to hear frequencies that are close together. This anomaly is known as masking.
- Get MFCC by applying DCT

Model Serving

How Do We Get Trained Models Into Production?

Compile the Model

- Examples
  - CMSIS-NN
  - Glow
  - µTVM

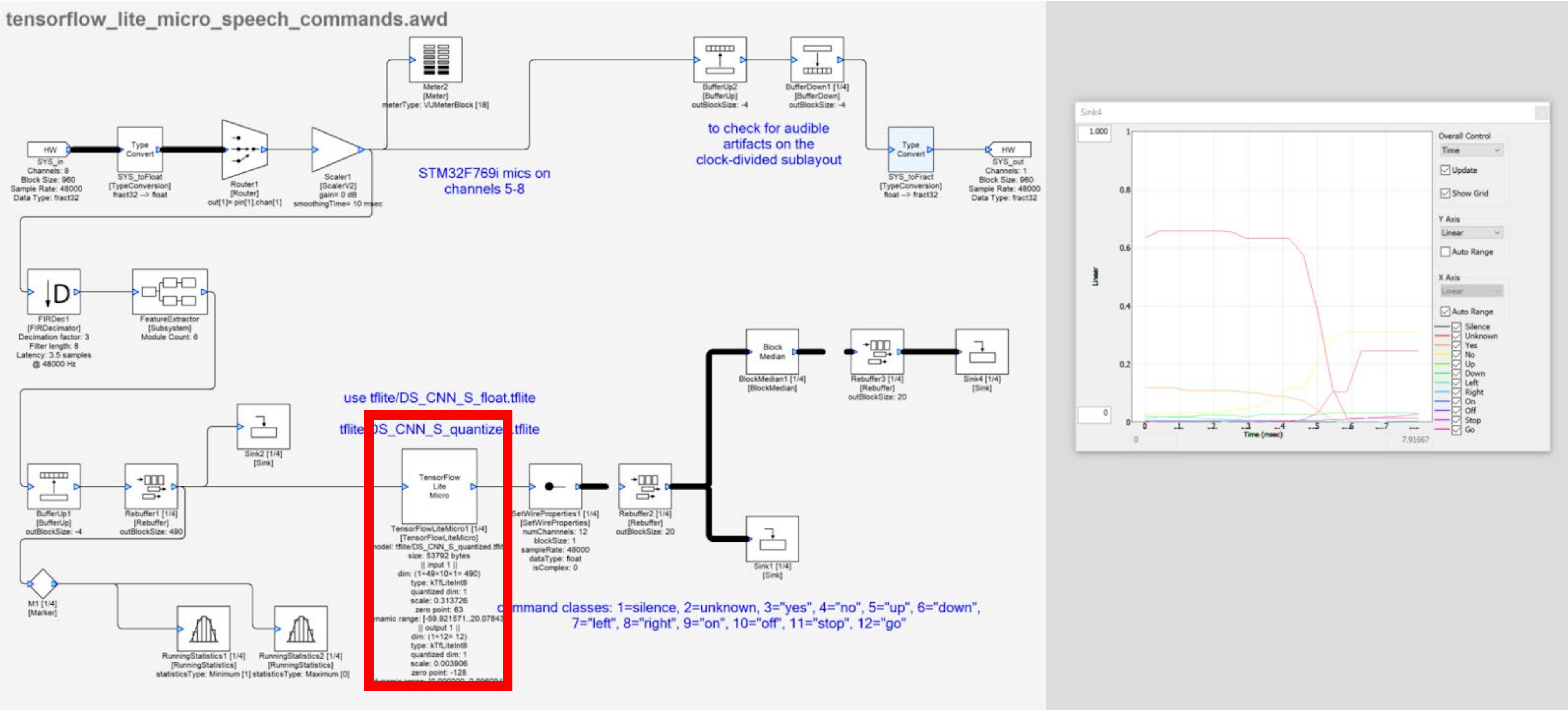
- Pros
  - More efficient
  - Supports more layers
- Cons
  - More complicated
  - Requires a build system/compilation process
  - Requires building a custom AW module for each model created

Runtime Interpreter

- Examples
  - Tensorflow Lite
  - Tensorflow Lite Micro
  - ONNX Runtime
  - Qualcomm eAI

- Pros
  - Easy to use
    - Interpreter can be wrapped as an AW module
  - No model compilation
    - Convert model using CLI tools
- Cons
  - Less efficient
  - Slower to support new layers/architectures

Example AWD Using TF Lite Micro Interpreter



Thank you!