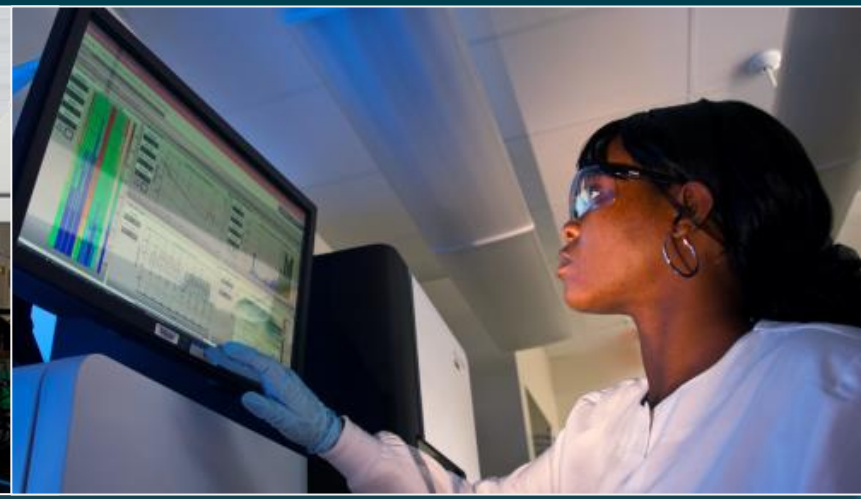
Frederick National Laboratory
for Cancer Research
sponsored by the National Cancer Institute





Machine Learning Jargon
An Introduction to Key Concepts and Terms

Andrew Weisman
High Performance Computing Analyst, Strategic and Data Science Initiatives

May 26, 2021

DEPARTMENT OF HEALTH AND HUMAN SERVICES • National Institutes of Health • National Cancer Institute
Frederick National Laboratory is a Federally Funded Research and Development Center operated by Leidos Biomedical Research, Inc., for the National Cancer Institute

Notes

- **<u>Scope:</u> Basic, commonly used machine learning terminology**
- **These terms will be highlighted in yellow**
- **I will focus on what people typically mean or refer to**
- **These are my own opinions**
- **There are nuances to everything**
- **Feel free to ask questions during the presentation:**
  - During the presentation, use the QA feature of Webex
  - We will allow attendees to unmute themselves for questions after the talk is complete
- **Links to slides and recording will be available at the NCI Data Science Learning Exchange (https://cbiit.github.io/p2p-datasci)**
- **Please fill out brief survey on Webex at the end of the talk**

Introduction

- **Machine learning: The science (and art) of programming computers so they can learn from data (Géron)**
  - Minimal explicit programming
- **Examples:**
  - Image processing: Optical character and facial recognition
  - Natural language processing (NLP): Spam filters
  - Bioinformatics: Tumor classification from gene expression data
- **Main types of machine learning:**
  - Supervised learning
  - Unsupervised learning
  - Semi-supervised learning
  - Reinforcement learning

Simple example of supervised learning: linear regression

- **<u>Experiment:</u> 10 random times throughout the day, look out the window, and count all the birds and foxes you see:**

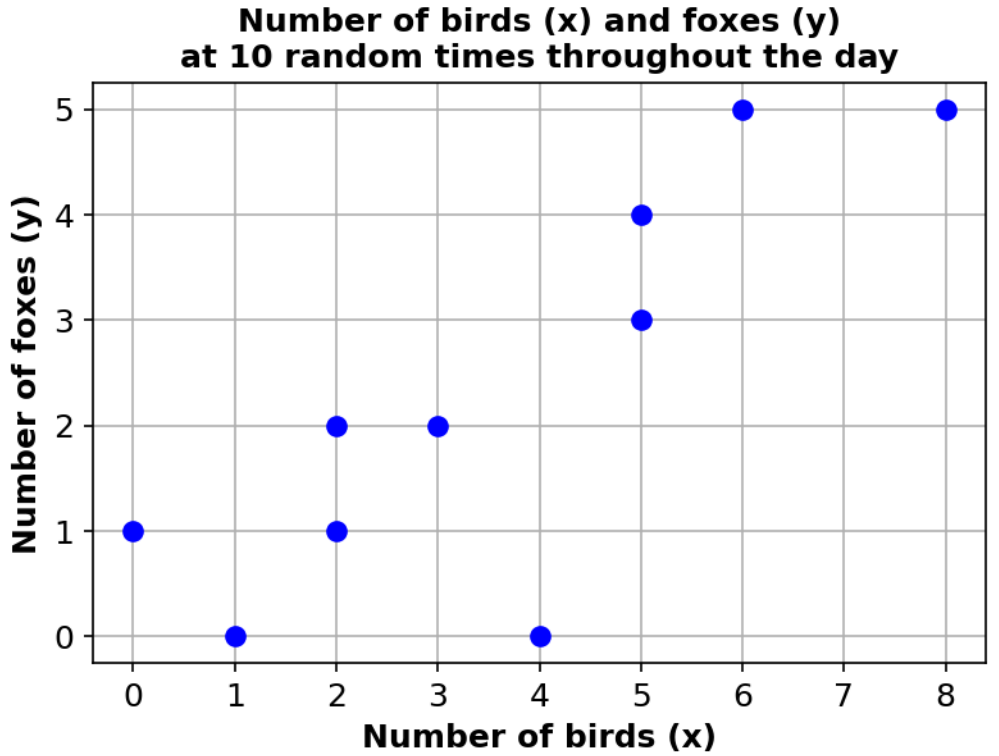
Each observation is called a sample or instance

The dependent variable *y* is called the label or target value

The independent variable *x* is called the feature

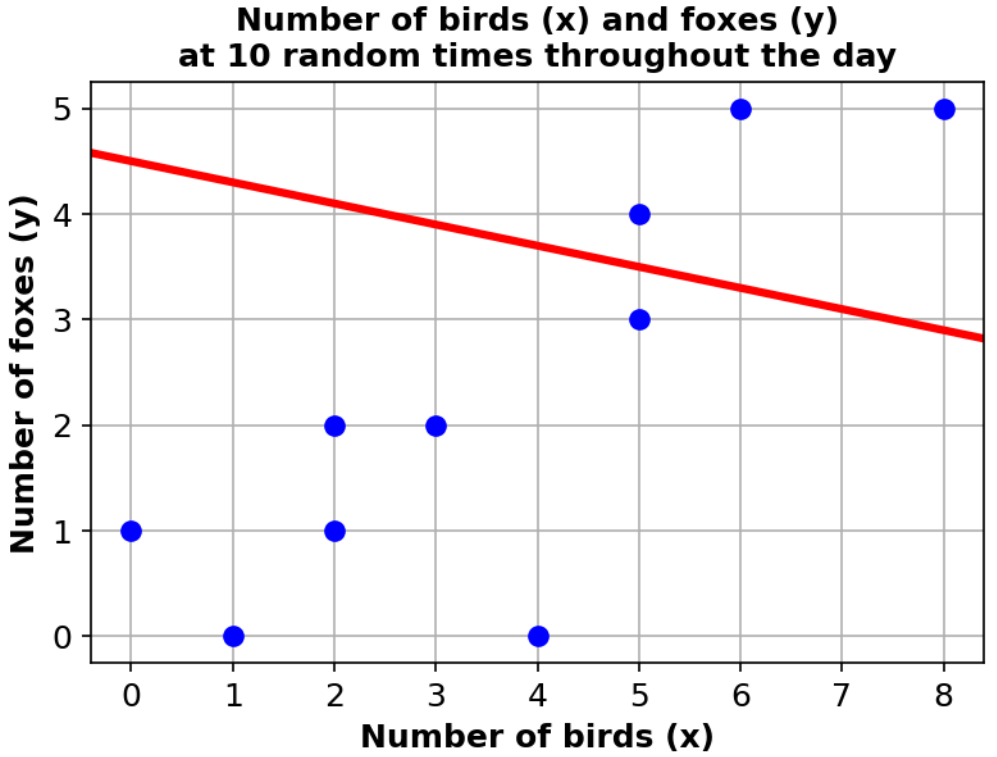
The entire dataset is called the training dataset

Number of birds (*x*)	Number of foxes (*y*)
5	3
2	2
1	0
3	2
6	5
0	1
4	0
2	1
8	5
5	4



Simple example of supervised learning: linear regression, ctd.

- **Question:** Can the number of birds (*x*) be used to predict the number of foxes (*y*)?
- **Approach:** Choose a **model** that relates *x* to *y*
- **Model selection:** The process of specifying the model architecture
- Here, let's select a **linear regression** model:
  $y_{predict}(m, b) = mx + b$
- m and b are called the **weights** or **parameters** of the model
- The process of determining the model's weights that best describe the data (*x* and *y*) is called **training** or **fitting**

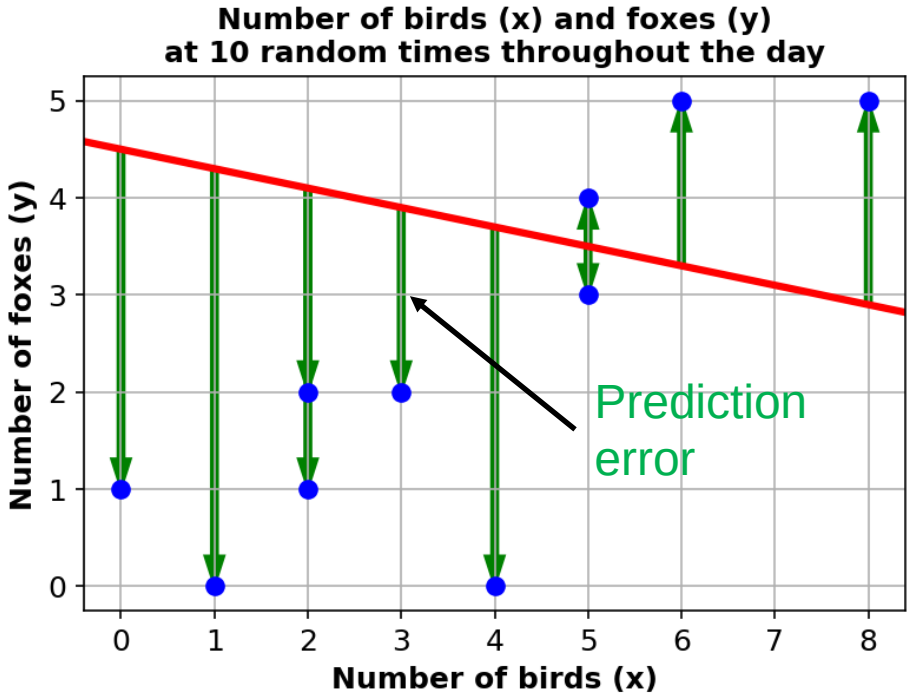


Training procedure

- **Weights initialization: Start with reasonable (or random) values for the model's weights:**
  m = -0.2 b = 4.5
- **Select a loss (or cost) function *L*(m, b) that describes how well the model fits the data with particular values of the weights**
- **Mean squared error:**

$$L(\mathrm{m}, \mathrm{b}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - y_{i,\mathrm{predict}}(\mathrm{m}, \mathrm{b})\right)^2$$

- **Example calculation:**

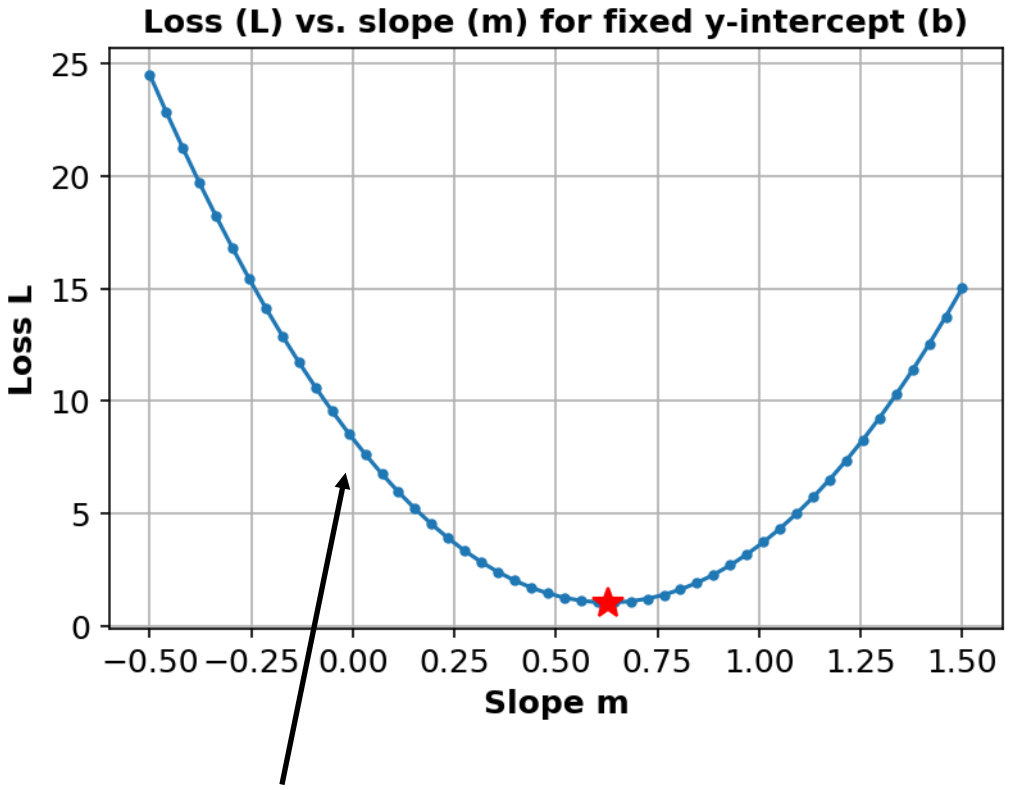


Sample ID (*i*)	# birds (x_i)	# foxes (y_i)	$y_{i,\mathrm{predict}}$(-0.2, 4.5)	(y_i – $y_{i,\mathrm{predict}}$(-0.2, 4.5))^2
1	5	3	3.5	0.25
2	2	2	4.1	4.41
⋮	⋮	⋮	⋮	⋮
N	5	4	3.5	0.25

Take average to obtain the starting loss:
L(-0.2, 4.5) = 6.99

Training procedure, ctd.

- **Point of training:** Minimize the loss *L*(m, b) by choosing better values for the weights m and b
- Weights are updated iteratively to reduce the loss
- A common algorithm for updating the weights is called gradient descent
- The model is said to be converged when the loss is minimized with respect to the weights
- This signifies the end of the training procedure

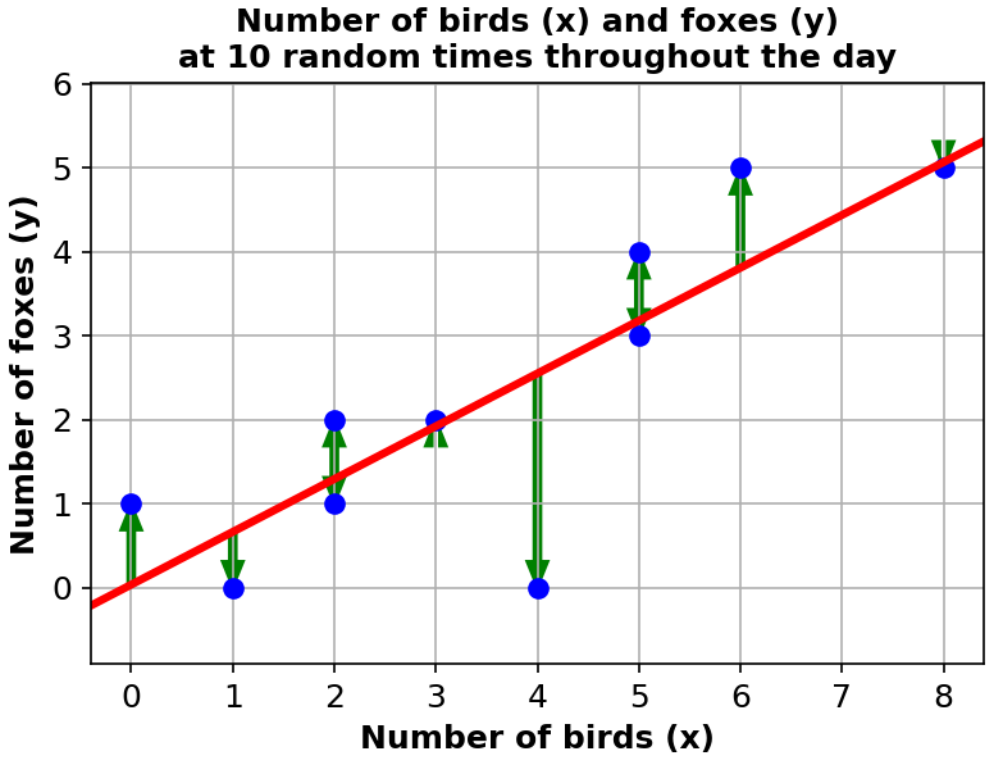


The degree the weights are allowed to change in an update step is called the learning rate

Training procedure: Key takeaways

- **After the training procedure is complete, you are left with weights (e.g., m and b) that allow you to most accurately predict the labels <u>using the selected model</u> (e.g., linear regression)**
- **In our example, we find**
  m = 0.63 b = 0.04
  are the weights that minimize the mean squared error, i.e., the loss
- **In other words, we are able to best predict the labels y_i in our training dataset using the formula:**

$$y_{i,\text{predict}}(0.63, 0.04) = 0.63x_i + 0.04$$



Testing procedure

- **Question: How does our model perform in the real world?**
  - We know our model performs as well as possible on the training dataset because that's what we used to optimize its weights
- **Answer: Provide some real-world data, called the testing dataset, that was not used for the training procedure**
  - Using the already-trained model (called inference), calculate the loss on the testing dataset, e.g.,

$$L_{\text{test}}(\mathbf{0.63}, \mathbf{0.04}) = \frac{\mathbf{1}}{N_{\text{test}}} \sum_{j=1}^{N_{\text{test}}} \left(\boldsymbol{y}_j - \boldsymbol{y}_{j,\mathbf{predict}}(\mathbf{0.63}, \mathbf{0.04}) \right)^{\mathbf{2}}$$

- **Using a holdout dataset such as a testing dataset helps to avoid overfitting the model to the training dataset**
- **This procedure gives you an idea of how well the model will generalize, informing you of the generalization error**

Data splitting

- **Data splitting is the process of splitting the available dataset into a training dataset and a testing dataset**
- **It is often as easy as taking a 70-30 or 80-20 split of the full dataset**
- **However, quality control measures must be taken to ensure both the training and testing datasets are representative of the real-world population**
- **Data splitting can include splitting into another dataset that is not used for training (other than the testing dataset) called the validation dataset**
- **The validation dataset is generally used for optimizing a model's hyperparameters**

	Number of birds (*x*)	Number of foxes (*y*)
Training dataset	5	3
	2	2
	1	0
	3	2
	6	5
	0	1
	4	0
Testing dataset	2	1
	8	5
	5	4

Hyperparameters

- A hyperparameter is a parameter that defines the model architecture itself
- It is not optimized in the training process
- For example, let's replace our linear regression model with a more-general polynomial regression model that we define as:

  $$y_{\text{predict}}(\{\theta\}) = \sum_{k=0}^{D} \theta_k x^k$$

  where *D* is the Degree of the polynomial
  - E.g., *D*=0: $y_{\text{predict}}(\theta_0) = \theta_0$
  - E.g., *D*=1: $y_{\text{predict}}(\theta_0, \theta_1) = \theta_0 + \theta_1 x$
  - E.g., *D*=2: $y_{\text{predict}}(\theta_0, \theta_1, \theta_2) = \theta_0 + \theta_1 x + \theta_2 x^2$
- We can say *D* is a hyperparameter of the polynomial regression model
- *D* defines the model architecture, including the weights $\{\theta\}$ that are present in the model
- Once *D* is specified and fixed, the weights are optimized as usual to minimize the loss

Hyperparameter optimization

- **Hyperparameter optimization is the process of optimizing a model's hyperparameters**
- **Just as you should evaluate a model's real-world performance on data that was not used in the training process (i.e., on a testing dataset)…**
  - …you should evaluate the effect of the hyperparameter values on data that was not used in the training process, on a third dataset called the validation dataset
- **Just as earlier we calculated L_{test} on the testing dataset using the already-trained model…**
  - …now we calculate L_{valid} on the validation dataset using the already-trained model
- **Note there are alternatives to "holding out" data just for validation, e.g., in cross-validation**

	Number of birds (*x*)	Number of foxes (*y*)
Training dataset	5	3
	2	2
	1	0
	3	2
	6	5
Validation dataset	0	1
	4	0
Testing dataset	2	1
	8	5
	5	4

Sample workflow for the polynomial regression model

- **For the *D*=0 model, optimize the weight θ_0 on the training dataset**
  - Evaluate the resulting model on the validation dataset, i.e., calculate $L_{\text{valid}}(D = 0)$
- **For the *D*=1 model, optimize the weights $\{\theta_0, \theta_1\}$ on the training dataset**
  - Evaluate the resulting model on the validation dataset, i.e., calculate $L_{\text{valid}}(D = 1)$
- **For the *D*=2 model, optimize the weights $\{\theta_0, \theta_1, \theta_2\}$ on the training dataset**
  - Evaluate the resulting model on the validation dataset, i.e., calculate $L_{\text{valid}}(D = 2)$
- **Etc. for $D \geq 3$**
- **Choose the value of *D* for which $L_{\text{valid}}(D)$ is a minimum**
  - Congratulations, you have just optimized the hyperparameter *D*!
- **Train the corresponding model on the training+validation datasets**
  - This yields your best model
- **Evaluate this model on the test dataset, i.e., calculate L_{test}**
  - This is your estimate of the generalization error

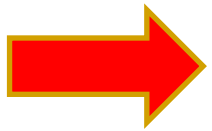
	Number of birds (*x*)	Number of foxes (*y*)
Training dataset	5	3
	2	2
	1	0
	3	2
	6	5
Validation dataset	0	1
	4	0
Testing dataset	2	1
	8	5
	5	4

Increasing the number of features

Number of birds (x)	Number of foxes (y)
5	3
2	2
1	0
3	2
6	5
0	1
4	0
2	1
8	5
5	4

x y

vector



Number of birds (x_1)	Number of cicadas (x_2)	Number of foxes (y)
5	90	3
2	30	2
1	50	0
3	50	2
6	40	5
0	20	1
4	70	0
2	40	1
8	150	5
5	30	4

X y

matrix

Data for classification tasks

Image classification

Image ID	Pixel 1	Pixel 2	. . .	Pixel *M*	Class
1	155	42	⋯	210	Hot dog (1)
2	24	255	⋯	192	Hot dog (1)
⋮	0	137	⋯	5	Not hot dog (0)
N	1	2	⋯	35	Hot dog (1)
		X			***y***

Cancer type classification using RNA-Seq counts data

Sample ID	Gene 1	Gene 2	. . .	Gene *M*	Cancer type
1	12005	2	⋯	500	Liver (0)
2	20000	500	⋯	1005	Pancreatic (1)
⋮	8888	0	⋯	459	Liver (0)
N	9000	5	⋯	1208	Breast (2)
		X			***y***

Supervised learning summary

- **In supervised learning, you generally start with a feature matrix *X* and a labels vector *y***
- **Select a model, which will contain adjustable weights $\{\theta\}$**
- **Train the model on *X* and *y* by minimizing a loss with respect to the weights $\{\theta\}$**
- **Then, obtain a single new sample, e.g.:**
  - Number of birds
  - Number of birds and number of cicadas
  - An image
  - RNA-Seq counts data from a single biological sample
- **Input this sample into the trained model (i.e., in inference mode) in order to make a prediction, e.g.:**
  - Number of foxes
  - Hot dog or not hot dog
  - Tumor type

Supervised learning summary, ctd.

- **In addition to making a prediction, some models are able to explain why that prediction was made**
  - This generally means informing the user which features were most crucial to making predictions
  - This is called feature importance
  - For example:
    - Were the number of birds more important than the number of cicadas in determining the number of foxes?
    - Which pixels in the image were most important in determining whether the image was a hot dog?
    - Which genes are most important in determining the type of tumor?
- **Often, the more complex the model, the harder it is to explain its predictions**
- **Sample supervised learning models: linear/polynomial regression, logistic regression, K-nearest neighbors, perceptron, support vector machines, random forests, neural networks**

Aside: Deep learning

- **When a reasonably large neural network model is used for a machine learning task, deep learning is being performed**
- **Deep learning is just a subset of machine learning**
- **It is special because the model architecture and the algorithms used to update its weights fit perfectly on graphics processing units (GPUs)**
  - I.e., we already have hardware specialized for deep learning!
- **As neural networks are often very large, they are also very powerful**
  - Major application in intensive tasks such as facial recognition and word/phrase prediction

Unsupervised learning

- **In unsupervised learning, the labels *y* are not provided (nor is there a need for them)**
- **Instead of prediction, the goal is to learn about the feature matrix *X* itself**
- **Sample questions that are asked:**
  - Which samples are similar to each other (clustering), or which ones are outliers?
  - Can we reduce the number of features to a smaller set of better-discriminating features (dimensionality reduction)?
- **Just as in supervised learning, these methods generally optimize weights to minimize a cost function**
- **Sample unsupervised learning models: K-Means, Gaussian mixture models, principal component analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), autoencoders**

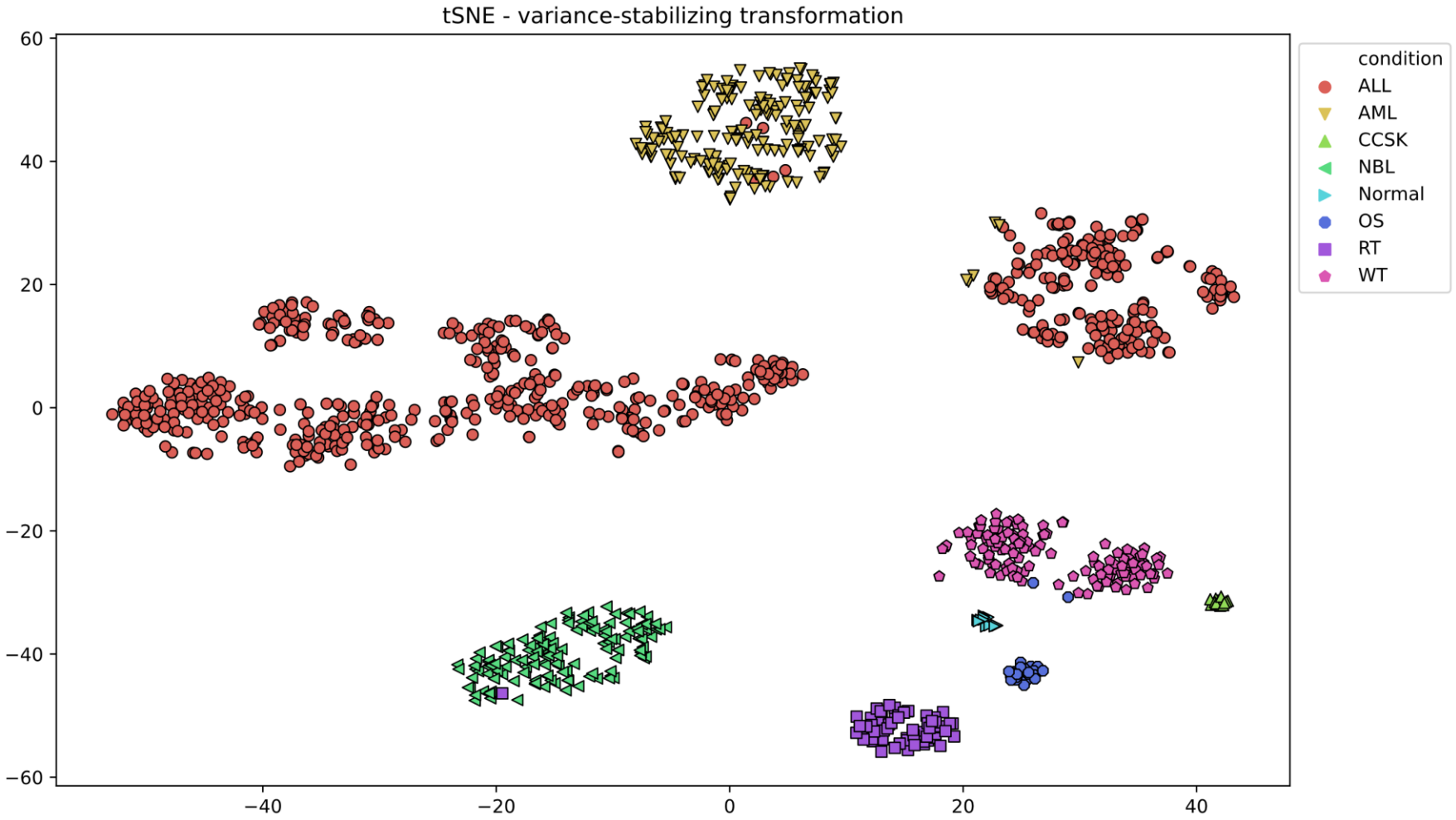
Example of dimensionality reduction on RNA-Seq counts data

- **Recall format of the data:**

Sample ID	Gene 1	Gene 2	. . .	Gene *M*	Cancer type
1	12005	2	⋯	500	Liver (0)
2	20000	500	⋯	1005	Pancreatic (1)
⋮	8888	0	⋯	459	Liver (0)
N	9000	5	⋯	1208	Breast (2)
			X		***y***

- **Reduce the dimensionality of the feature matrix *X* using t-SNE**
- **Create a scatterplot of the two most highly varying, transformed features**
- **Color the datapoints by their labels *y***

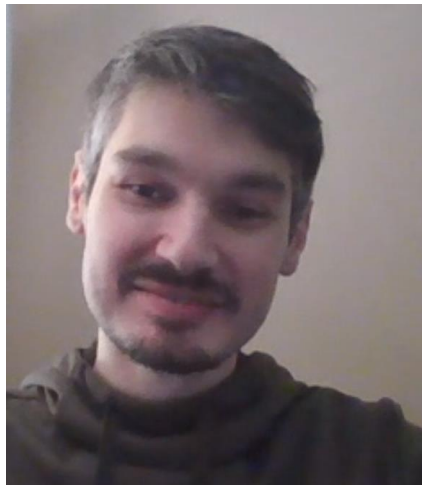
Example of dimensionality reduction on RNA-Seq counts data, ctd.



Credits

Frederick National Laboratory for Cancer Research
sponsored by the National Cancer Institute

SURVEY!!



Mike Rinaldi
Audio Visual Support

Petrina Hollingsworth
Community Engagement

Lynn Borkon
AI/HPC Collaborations Development

Questions?