APPLIED MACHINE LEARNING

BY
MESFIN DIRO

July 21, 2021

OUTLINE

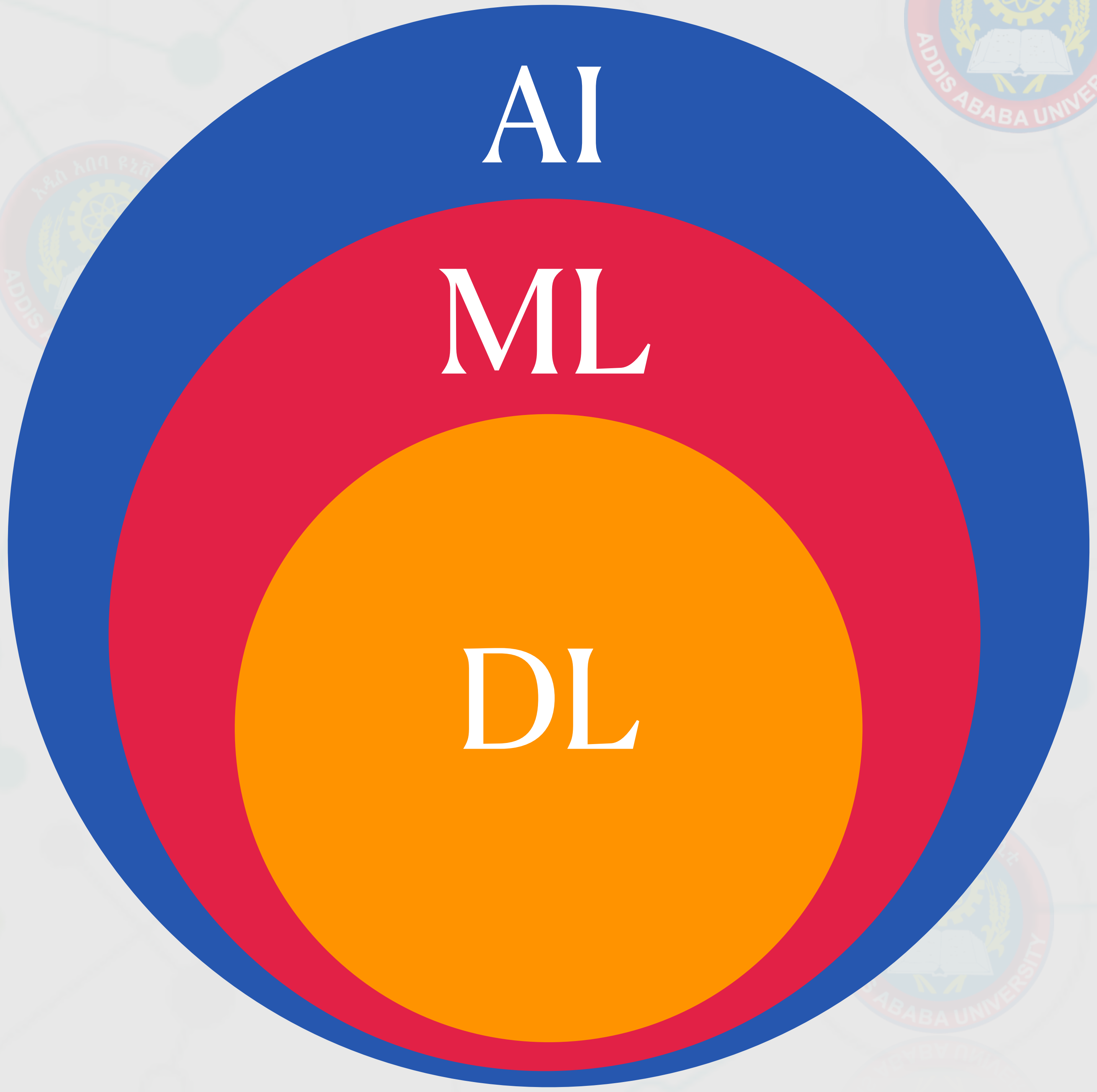
- ML Basic Concepts
  - Supervised ML
    - Classification
    - Regression
  - Unsupervised Learning
    - Clustering
  - Overfitting/under-fitting
  - bias/variance trade-off
  - Learning Curve
- ML Preprocessing
  - Regularization
  - Mean Removal
  - MinMax Scaling
  - Binarization
  - Label Encoder
- ML Algorithms
  - k-Nearest Neighbors
  - Naive Bayes
  - SVM
  - Decision trees
  - AdaBoost and RankBoost
  - Neural Networks

TOOLKITS LAB 1

- Toolkit Lab
  - Anaconda / miniconda
  - Jupiter lab / Jupiter notebook
  - Markdown
  - ipython
  - scikit-learn
  - pytroch
  - google colab
  - binder

INTRODUCTION

INTRODUCTION



- Artificial AI is the automation of intellectual tasks normally performed by humans

INTRODUCTION

AI

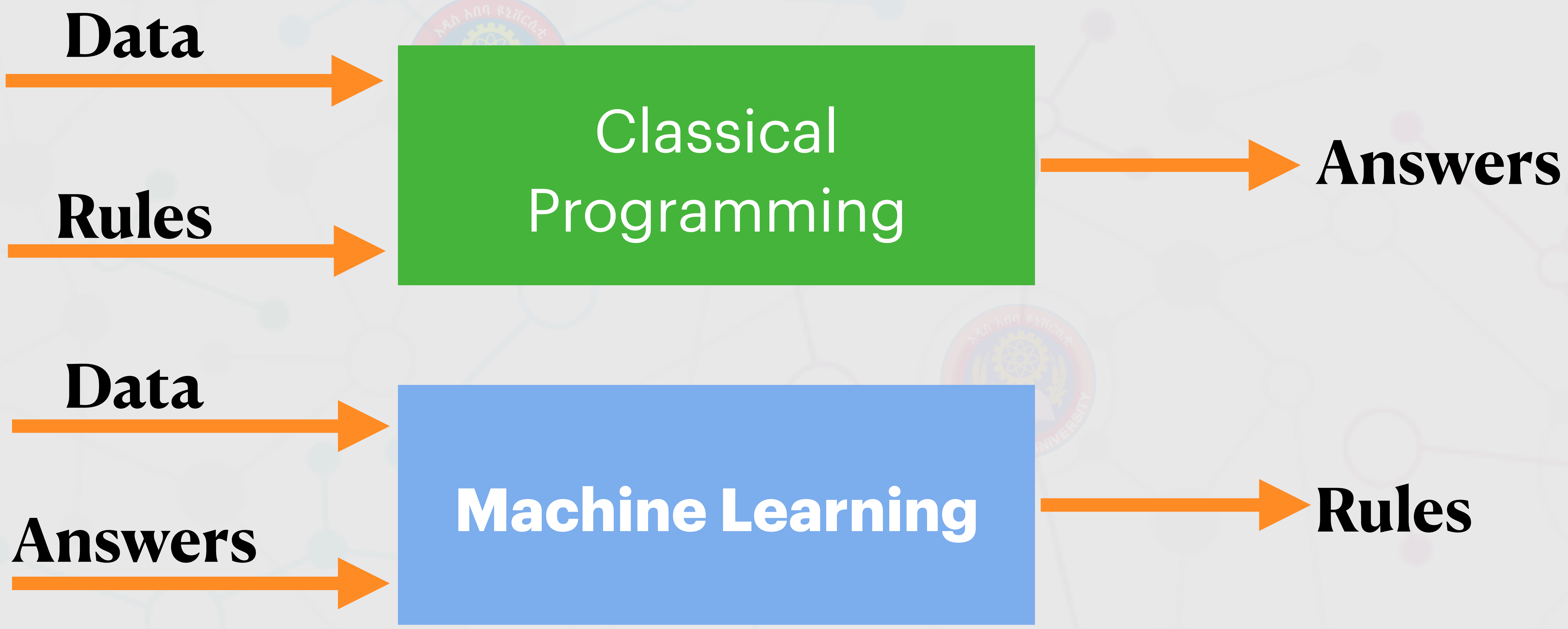
- Artificial Intelligence(AI) is a way to make machines think and behave intelligently.
- There are different branches of AI:
  - Machine Learning(ML)
  - Deep Learning(DL)
  - Natural Language Processing(NLP)
  - Computer Vision
  - Speech

INTRODUCTION

- Arthor Samuel(1959): Machine Learning is a field of study that gives computers the ability to learn without being explicitly programmed
- Tom Mitchell(1998): A computer program is said to learn from Experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improve with experience E.

ML

MACHINE LEARNING

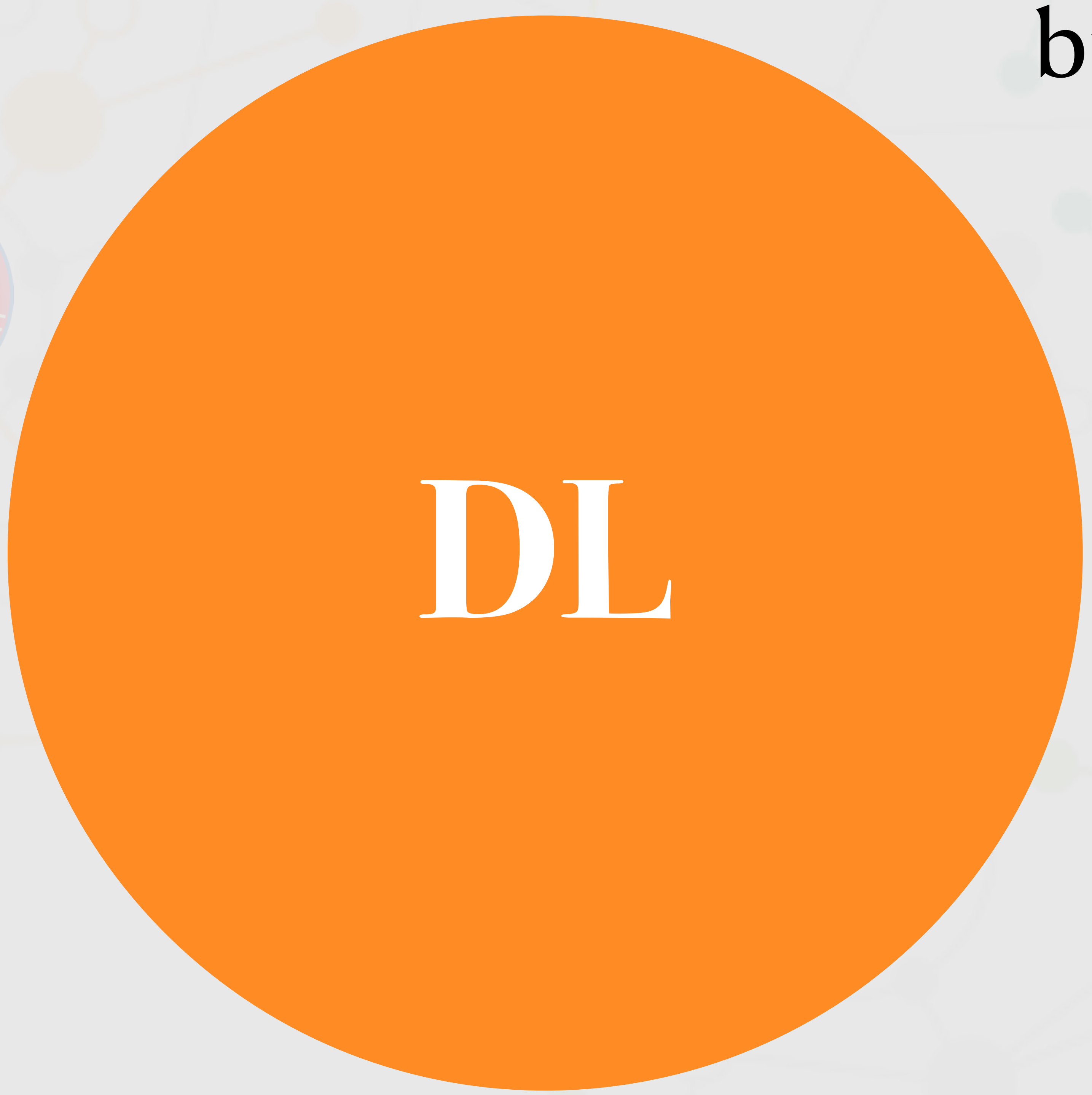


- At a high level, ML systems look a tons of data and come up with rules to predict outcomes for unseen data.
- Fundamentally, machine learning involves building mathematical models to help understand data.

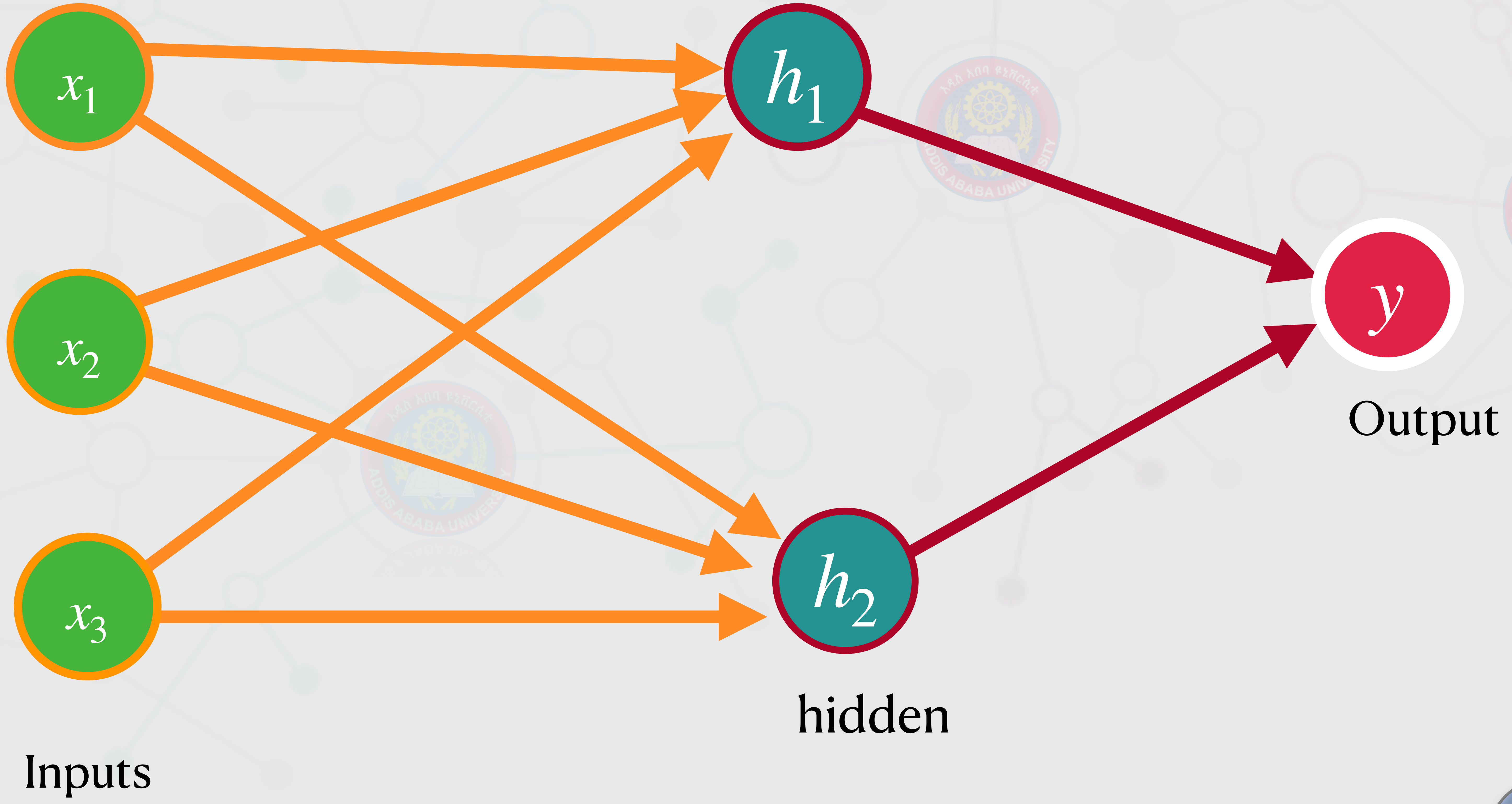
- ML systems learn how to combine input to produce useful predictions on never-before-seen data.
- A label is the thing we're predicting—the *y* variable in simple linear regression. The label could be the future price of wheat, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.
- A feature is an input variable—the x variable in simple linear regression. A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features, specified as:

$$x_1, x_2, \cdots, x_N$$

ML



- **Deep Learning** is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called **artificial neural networks**.



- A linear layer applied by a linear transformation as:

$$y = wx + b \qquad (1.1)$$

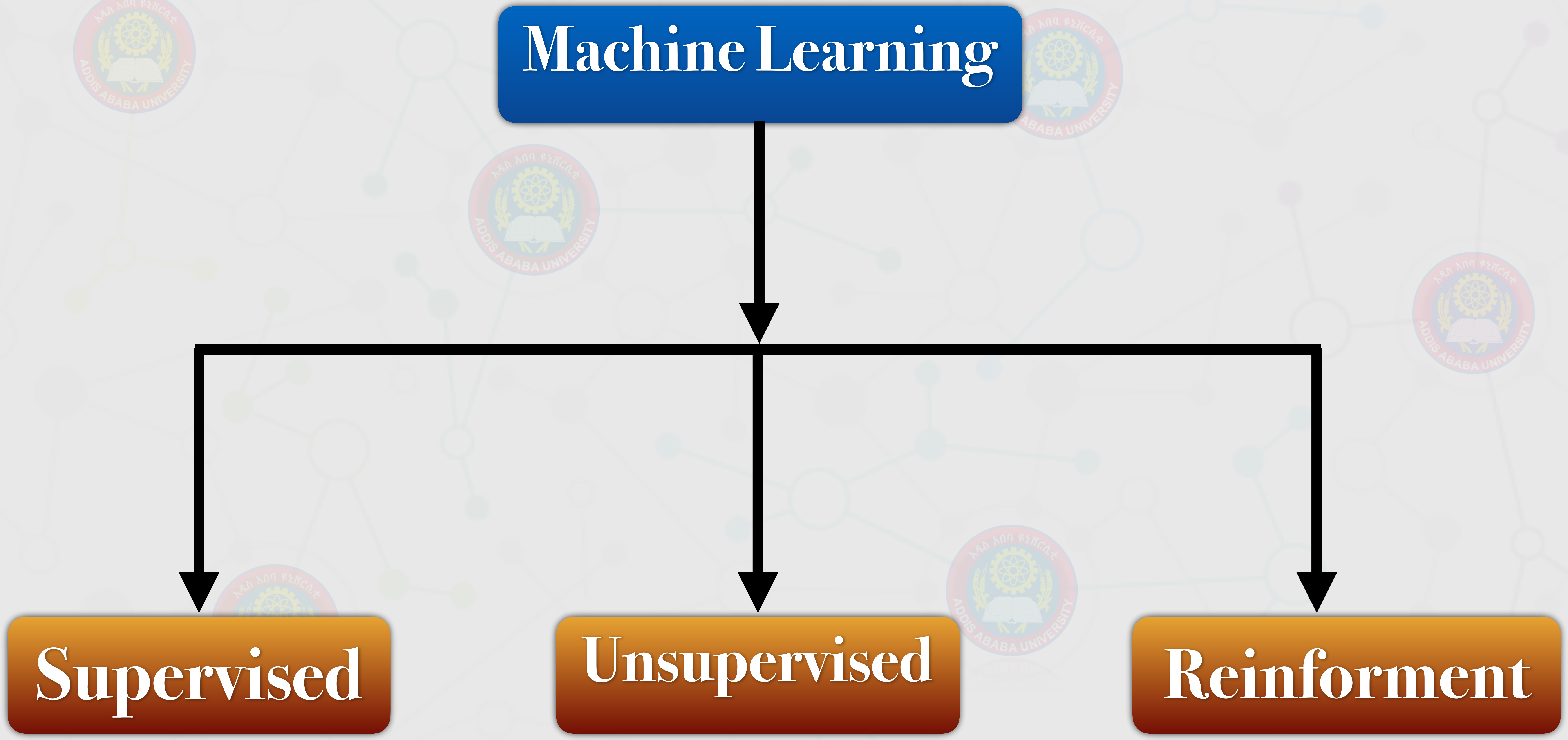
where $w$and b are weights and bias respectively which are learnable parameters in the neural network

DL

MACHINE LEARNING MODEL

- **A model** defines the relationship between features and label. For example, a spam detection model might associate certain features strongly with "spam". Let's highlight two phases of a model's life:
  - **Training** means creating or **learning** the model. That is, you show the model labeled examples and enable the model to gradually learn the relationships between features and label.
  - **Inference** means applying the trained model to unlabeled examples. That is, you use the trained model to make useful predictions (y'). For example, during inference, you can predict *medianHouseValue* for new unlabeled examples.

MACHINE LEARNING MODEL



The three domains of ML

TYPE OF MACHINE LEARNING

- The ML modeling process insists a modeling phase of four stages:
  1. Feature engineering and model selection
  2. Training the model
  3. Model validation and selection
  4. Applying the trained model to unseen data

SUPERVISED LEARNING

- Most of the successful use cases of machine learning and deep learning space fall under **supervised learning**.
- Some of the common examples of supervised learning are:
  - Classification Problem
  - Regression Problem

SUPERVISED LEARNING

- A regression model predicts continuous values. For example, regression models make predictions of that answer questions like the following:
  - What is the value of a house in Addis Ababa?
  - What is the probability of having rain tomorrow in Addis Ababa?
- A classification model predicts a discrete(categorical) values. For example, classification models make predictions of that answer the questions like the following:
  - is this image of a cat or dog
  - is a given whether cloudy, rainy, or sunny

UNSUPERVISED LEARNING

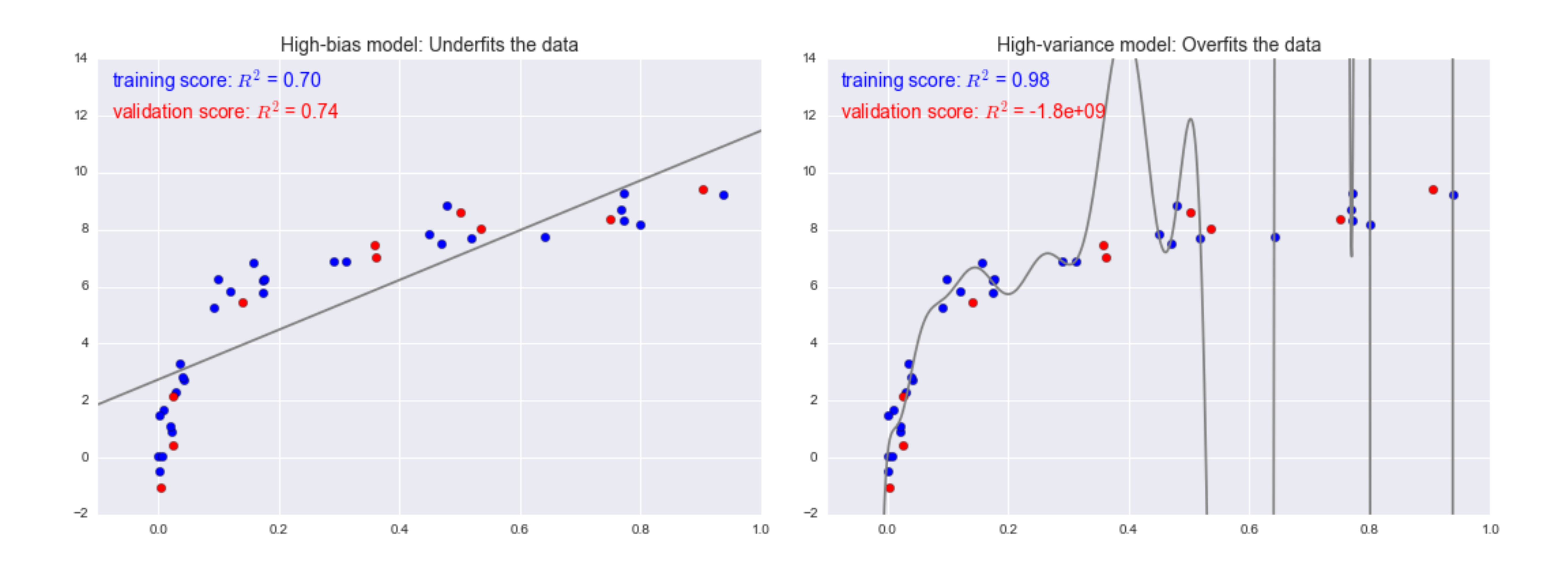
- When there is no label data, unsupervised learning techniques help in understanding the data by visualizing and compressing.
- The two most commonly-used techniques in unsupervised learning are:
  - Clustering : helps in grouping all similar data-points together
  - Dimensionality Reduction: helps in reducing the number of dimensions(features) so that we can visualize the high dimensional data to find hidden patterns.

EVALUATING ML MODELS

- To avoid the problem of an information leak and improve generalization, it is often a common practice to split the datasets into three different parts: training dataset, validation dataset and test dataset
  1. Trying the algorithm on the training dataset
  2. Perform hyper-parameter tuning based on the validation dataset
  3. Perform the first two steps interactively until the expected performance is achieved
  4. After freezing the algorithm and the hyper-parameters, evaluate it on the test dataset

THE BIAS -VARIAS TRADE-OFF

- Fundamentally, the question of “ the best model” is about finding the sweet spot in the trade-off between bias and variance.
- For high bias models, the performance of the model on the validation set is similar to the performance in the training set.
- For high variance models, the performance of the model on the validation set is far worse than the performance of the training set.

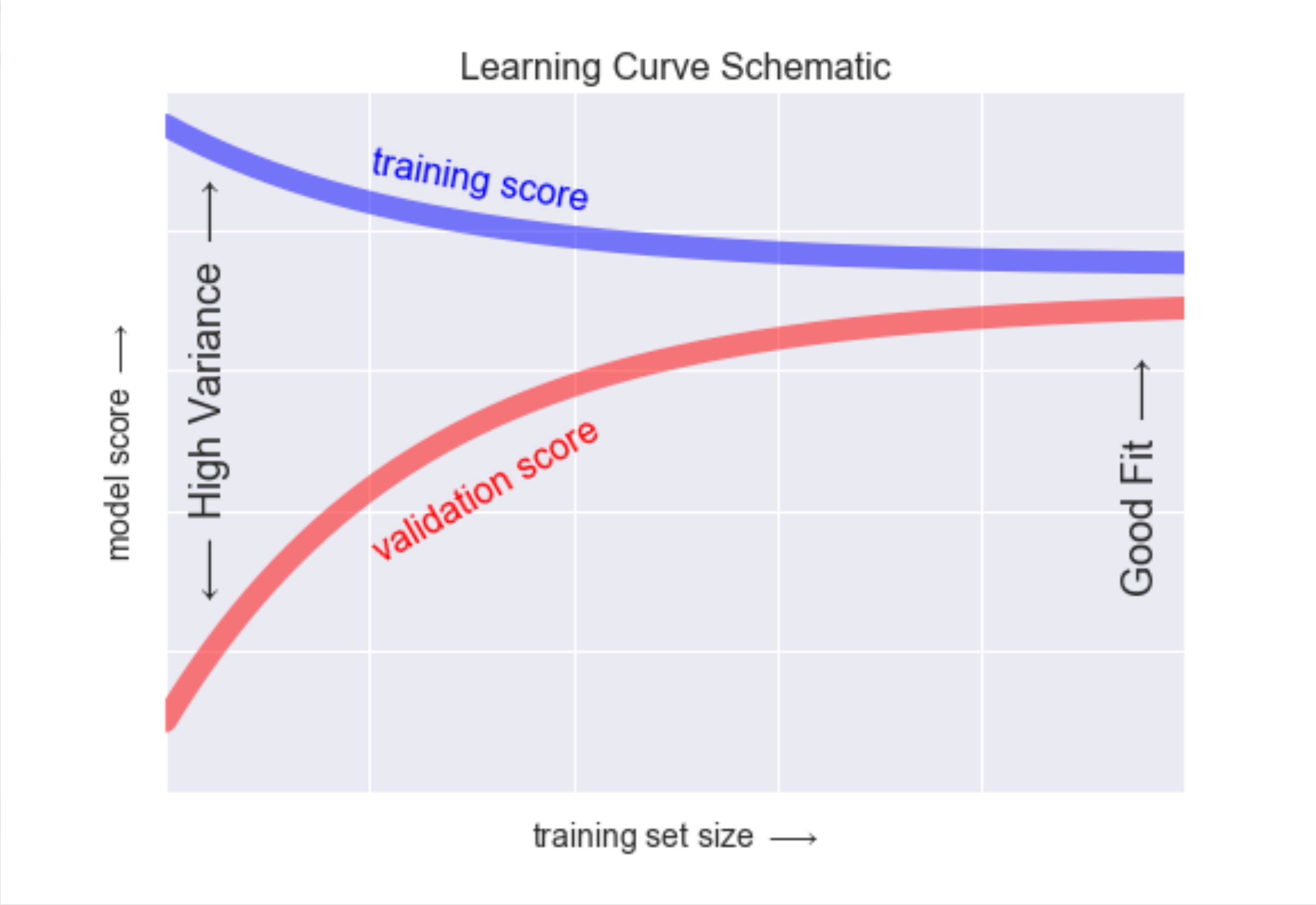


OVER-FITTING AND UNDER-FITTING

- Overfitting, or not generalizing, is a common problem in machine learning and deep learning.
- A particular algorithm overfits when it performs well on the training dataset but fails to perform on unseen or validation and test datasets.
- There are different techniques that can be used to avoid the algorithm overfitting. Some of the techniques are:
  - Getting more data
  - Applying regularizer

LEARNING CURVE

- One important aspect of model complexity is that the optimal model will generally depend on the size of the training data.
- A plot of the training/validation score with respect to the size of the training set is known as a learning curve
- A model will never except by chance give a better score to the validation set that the training set.



DATA PREPROCESSING

- Data preprocessing is a process in which we make the data more suitable for the ML algorithms to train on. The following are some of the commonly-used data preprocessing steps:
  - Binarization: convert our numerical values into boolean values
  - Mean removal:
  - MinMax Scaling
  - Normalization

BINARIZATION

- This is a process when we want to convert our numerical values into boolean values. for instance:

```
import numpy as np
from sklearn.preprocessing import Binarizer

data = np.array([[5.2, -3, 3.5],
                 [-2.0,7.0,-6.2],
                 [-7.4,-9.9,-5.4]])


#Binarize data

binarized = Binarizer(threshold=2.0,).transform(data)
print("\n Binarized data: \n", binarized)
```

BINARIZATION

- If you run the code, you will get the following output

```
Binarized data:
 [[1., 0.0, 1.],
  [0.,1.0,0.0],
  [0.,0.,0.0]])

#Binarize data
```

- All the values above 2.0 becomes 1 and the remains values become 0.0

MEAN REMOVAL

- Removing the mean is a common preprocessing technique used in machine learning.
- It helps to center each feature mean on zero in order to remove bias from the featured in feature vectors

```
import numpy as np
from sklearn.preprocessing import scale

data = np.array([[5.2, -3, 3.5],
                [-2.0,7.0,-6.2],
                [-7.4,-9.9,-5.4]])
#print mean and standard deviation
print("\n Before:")
print("mean = ", data.mean(axis=0))
print("Standard deviation=", data.std(axis=0))
#Remove mean
scaled = scale(data)
print("\n After:")
print("Mean =", scaled.mean(axis=0))
print("Standard deviation =", scaled.std(axis=0))
```

MEAN REMOVAL

- If you run the code, we will get the following printed on our terminal where the mean is very close to 0 and the standard deviation is 1.

```
Before:
mean =  [-1.4        -1.96666667 -2.7       ]
Standard deviation= [5.16139516 6.93797921 4.39621049]

 After:
Mean = [7.40148683e-17 0.00000000e+00 1.11022302e-16]
Standard deviation = [1. 1. 1.]
```

MINMAX SCALING

- When the value of each feature varies between many random values, it becomes important to scale those features so that it is a level playing field for the ML algorithm to train on.

```
import numpy as np
from sklearn.preprocessing import MinMaxScaler
data = np.array([[5.2, -3, 3.5],
                [-2.0,7.0,-6.2],
                [-7.4,-9.9,-5.4]])
#Min max Scaling
minmax_scaler = MinMaxScaler(feature_range=(0,1))
minmax_scaled = minmax_scaler.fit_transform(data)
print("\n Min max scaled data= \n", minmax_scaled
```

MINMAX SCALING

- The minimal Scaler in our previous code will generate the following print on our terminal.

```
Min max scaled data=
[[1.         0.40828402 1.        ]
 [0.42857143 1.         0.        ]
 [0.         0.         0.08247423]]
```

NORMALIZATION

- Normalization modify the values in the feature vectors so that we can measure them on a common scale.
- The most common forms of normalization aim to modify the values so that they sum up to one(1).
- **L1 normalization** which refers to **Least Absolute Deviations** works by making sure that the sum of absolute values is 1 in each row.
- L2 normalization which refers to least squares works by making sure that the sum of squares is 1.
- In general L1 normalization technique is considered more robust than L2 normalization.

NORMALIZATION


```
import numpy as np
from sklearn.preprocessing import normalize

data = np.array([[5.2, -3, 3.5],
                 [-2.0,7.0,-6.2],
                 [-7.4,-9.9,-5.4]])

#normalize data

l1_norm = normalize(data, norm='l1')
l2_norm = normalize(data, norm='l2')

print("\n L1 normalized data: \n", l1_norm)
print("\n L2 normalized data: \n", l2_norm)
```

NORMALIZATION

- The output of the previous normalization code print the following result on our terminal:

```
L1 normalized data:
[[ 0.44444444 -0.25641026  0.2991453 ]
 [-0.13157895  0.46052632 -0.40789474]
 [-0.32599119 -0.43612335 -0.23788546]]

L2 normalized data:
[[ 0.74829827 -0.43171054  0.5036623 ]
 [-0.20915194  0.73203177 -0.648371  ]
 [-0.54863001 -0.73397799 -0.40035163]]
```

LABEL ENCODEING

- When we perform classification, we usually deal with a lot of categorical labels.
- These labels can be in the form of words, numbers or something else.
- However, the ML algorithms expect them to be numbers

LABEL ENCODEING

```
 Label mapping:
cloudy --> 0
rainy --> 1
sunny --> 2

Encoded values = [2, 0, 2]
Decoded labels = ['sunny', 'cloudy', 
'sunny']
```

EVALUATION METRICS(NEXT)

- Performance metrics are a part of every machine learning pipeline.
- It is important to keep measure the performance of trained model how well it generalizes on the unseen data.
- The machine learning fall within different categories and we have different metrics for these categories.
- Metrics are different from loss functions:
  - Loss functions show a measure of model performance using some king of optimization like gradient descent
  - Metrics are used to monitor and measure the performance of a model which don't need to be differentiable
- These metrics can be broken down to either Classification or Regression

CLASSIFICATION METRICS

- CONFUSION MATRIX