Machine Learning

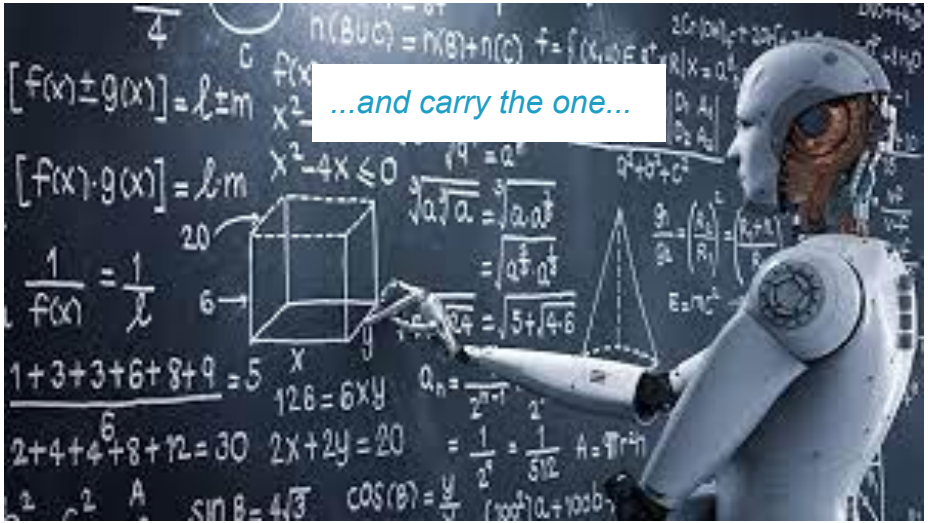


Image Source

Welcome to Week 5 Lecture 1!

Data Science in Python & Machine Learning

Announcements:

1. The new stack is open. Make sure to go to the Learn Platform and start working in the new stack.

2. The Machine Learning Calendar has been posted and is linked here.

3. Make sure to sign up for Group Code Reviews again if you would like to continue or start to have a group code review.

4. Check your feedback document on a regular basis.

Assignments Due This Week by Sunday night

1. Abalone Preprocessing Exercise (Core)
2. Pipelines Activity (Core)
3. Project 1 - Part 5 (Core)

Warm up

What is Machine Learning?

Learning Goals

After this class you will be able to:

1. Explain what machine learning is and what it does
2. Describe the different major categories of machine learning
3. Define ‘model validation’ and explain how we know if a model is valid and will work in the real world
4. Explain data leakage and how to avoid it
5. Use Standard Scaler to scale numeric values using without leaking data.

Questions Machine Learning Answers:

- Does the patient have cancer?

Questions Machine Learning Answers:

- Does the patient have cancer?
- How much will my house sell for?

Questions Machine Learning Answers:

- Does the patient have cancer?
- How much will my house sell for?
- How old is this animal?

Questions Machine Learning Answers:

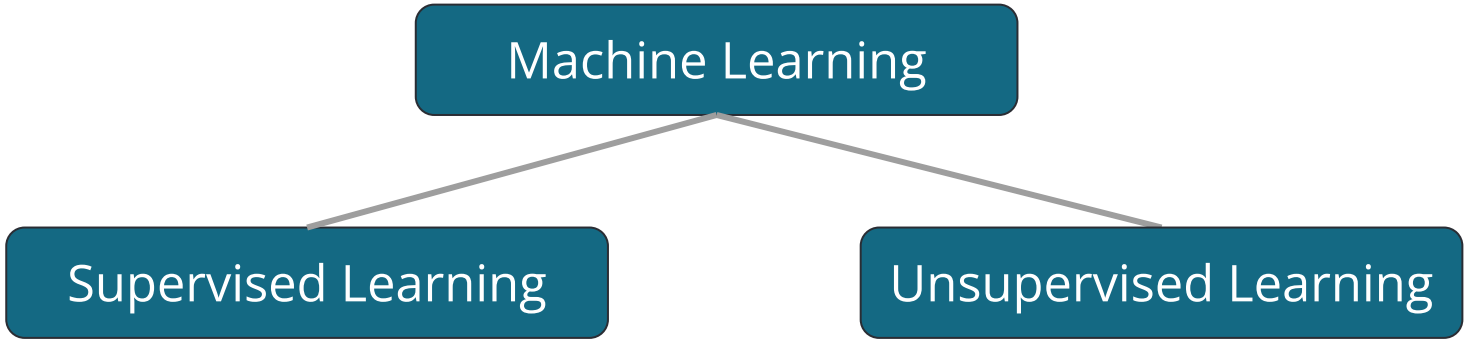
- Does the patient have cancer?
- How much will my house sell for?
- How old is this animal?
- Will the client repay this loan?

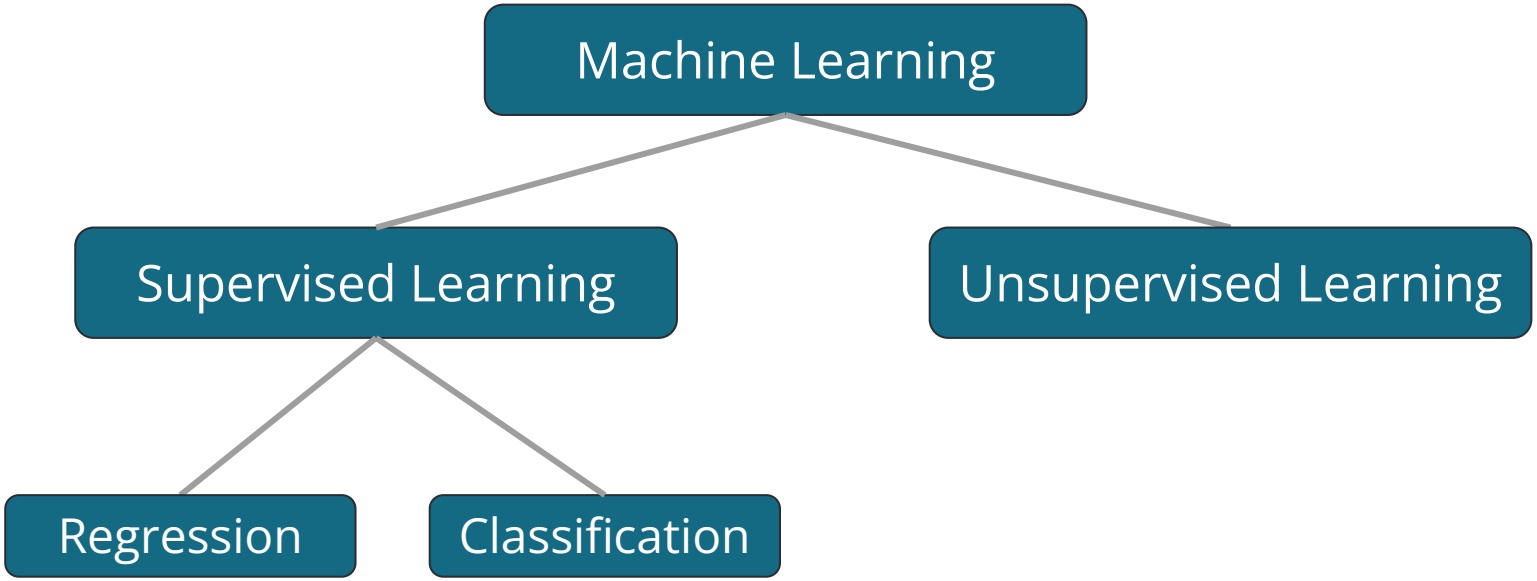
Questions Machine Learning Answers:

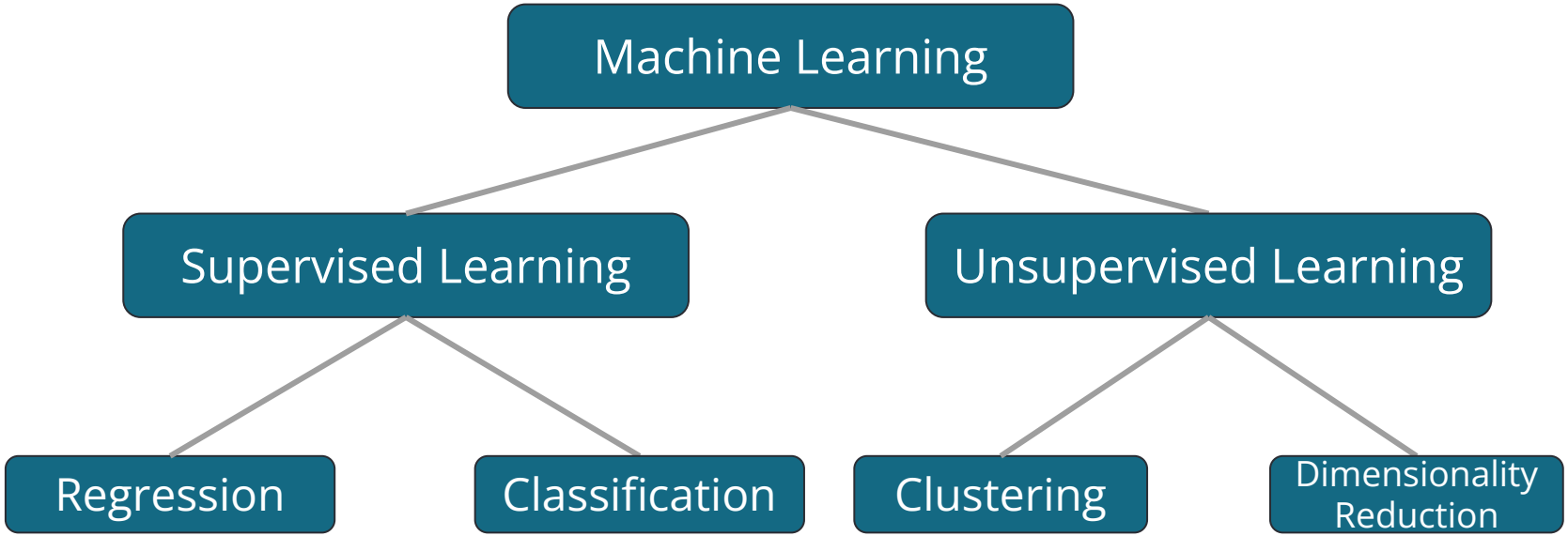
- Does the patient have cancer?
- How much will my house sell for?
- How old is this animal?
- Will the client repay this loan?
- Which customers are similar?

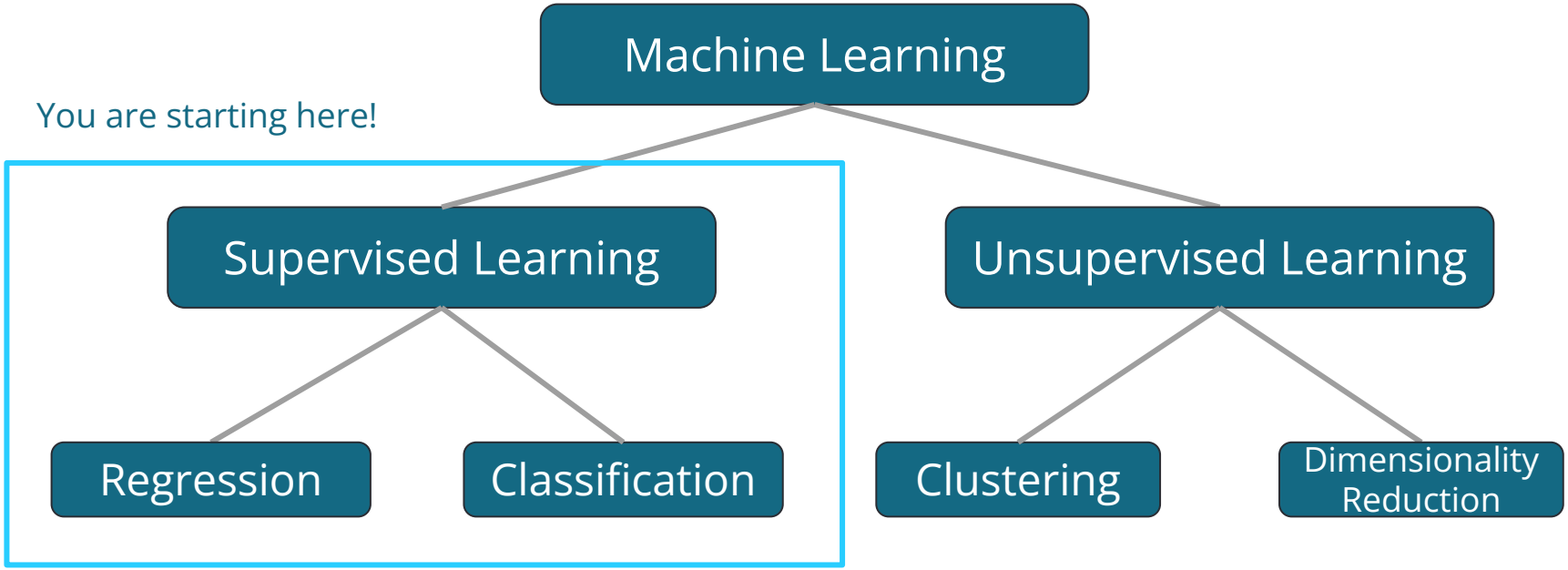
Machine Learning

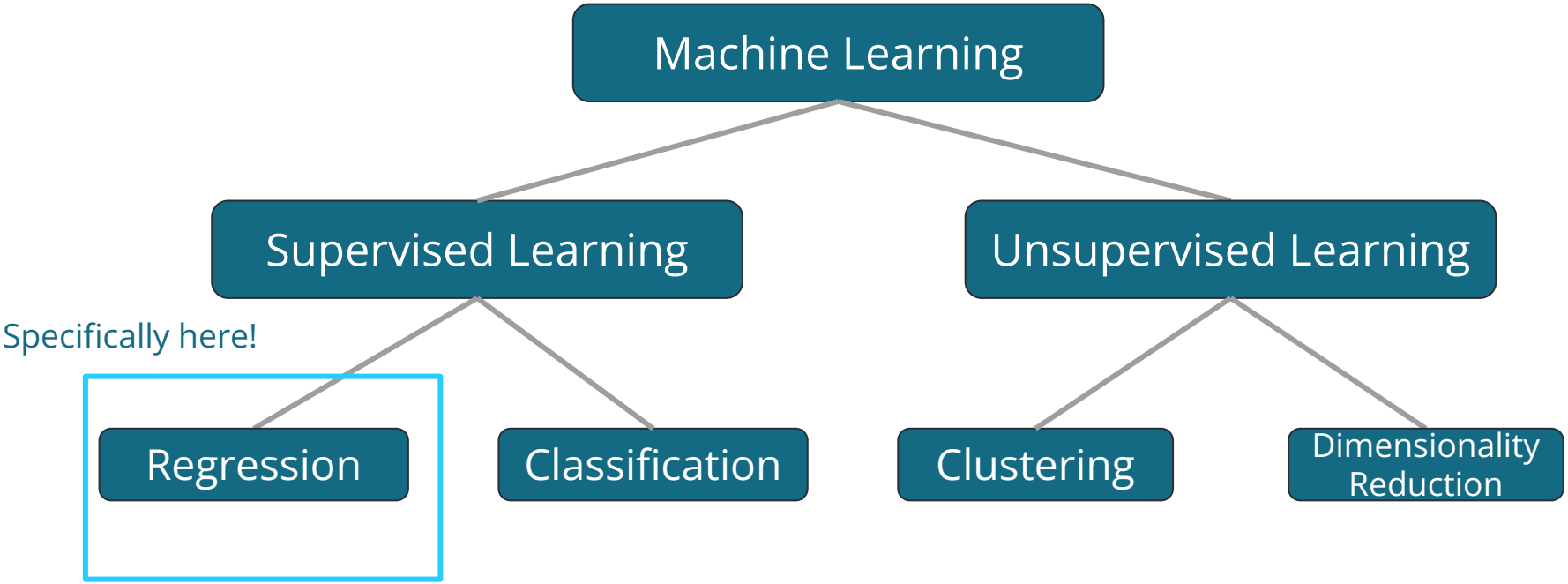
Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

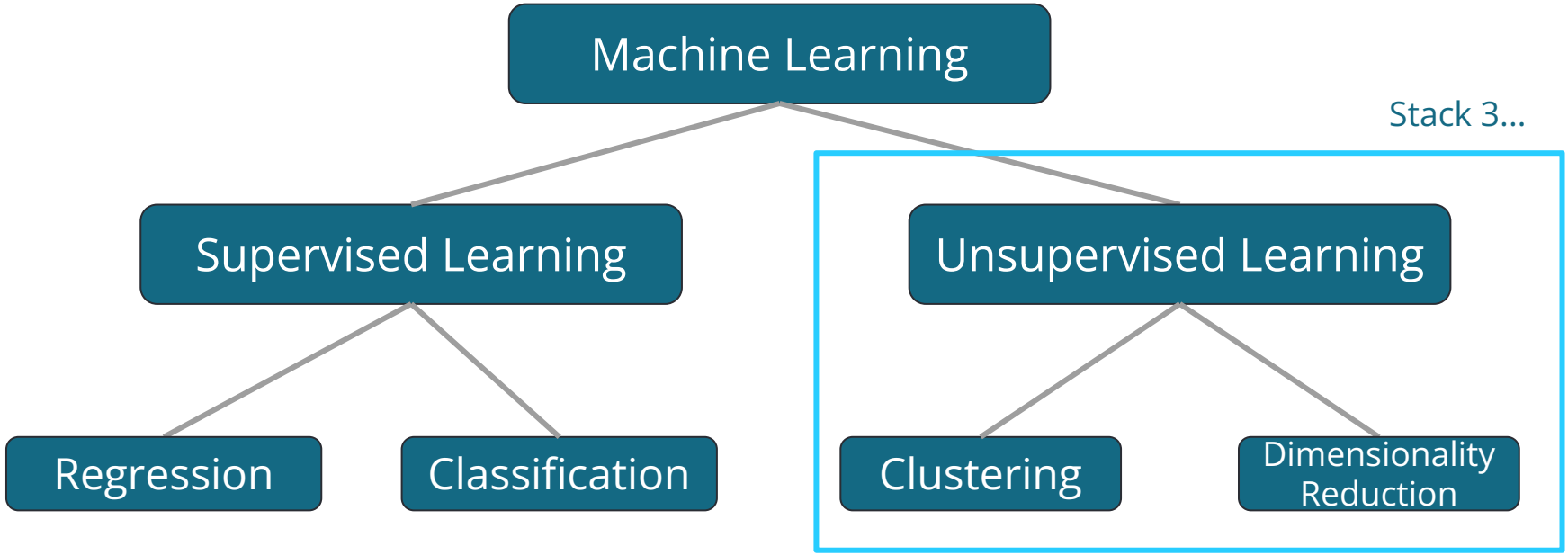

Machine Learning
Supervised Learning
Unsupervised Learning


Machine Learning
Supervised Learning
Unsupervised Learning
Regression
Classification


Machine Learning
Supervised Learning
Unsupervised Learning
Regression
Classification
Clustering
Dimensionality Reduction


Machine Learning
You are starting here!
Supervised Learning
Unsupervised Learning
Regression
Classification
Clustering
Dimensionality Reduction


Machine Learning
Supervised Learning
Unsupervised Learning
Specifically here!
Regression
Classification
Clustering
Dimensionality Reduction


Machine Learning
Stack 3...
Supervised Learning
Unsupervised Learning
Regression
Classification
Clustering
Dimensionality Reduction

Supervised Learning

Supervised Learning

Under supervision! Has a target variable to learn patterns - fitting a “if this - then this” pattern.

Supervised Learning

↳ Regression

Supervised Learning

↳ Regression

When your goal is to predict a numeric variable.

Supervised Learning

↳Regression

When your goal is to predict a numeric variable.

ex. Predicting home prices, sales, etc.

Supervised Learning

↳Classification

Supervised Learning

↳ Classification

When your goal is to predict a category or group.

Supervised Learning
↳Classification

When your goal is to predict a category.

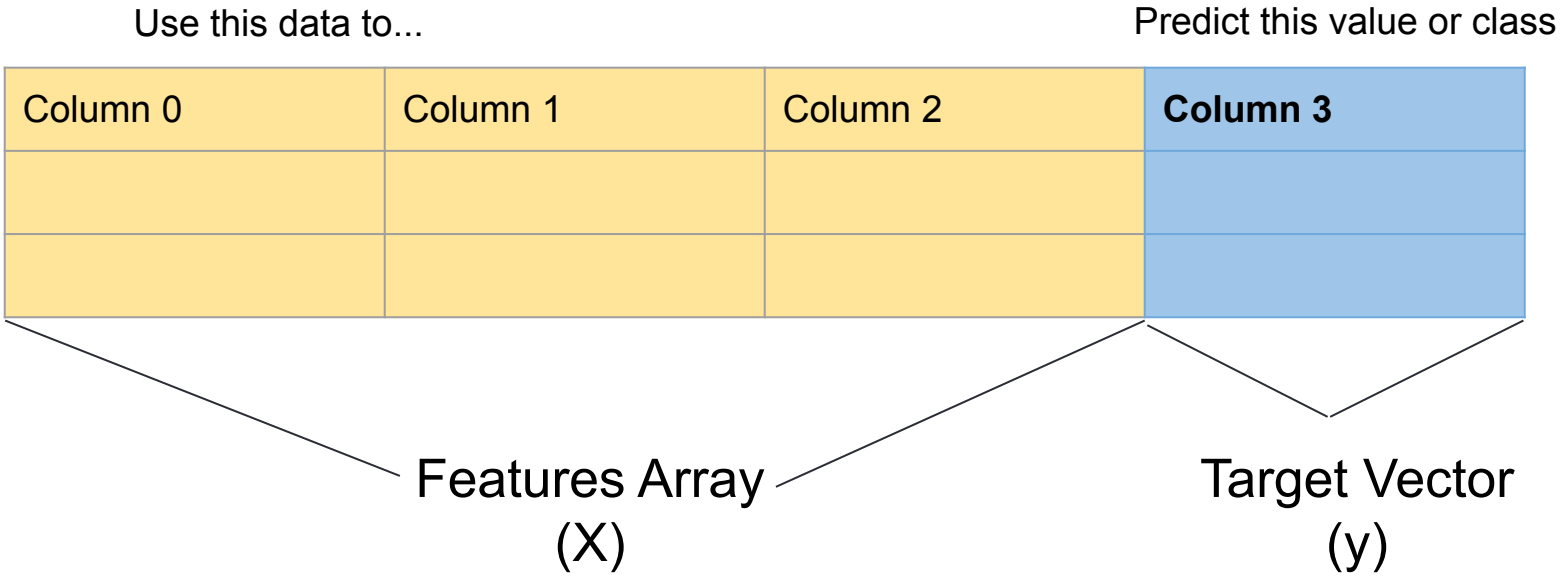
ex. Predicting yes/no, low/medium/high, etc.

Unsupervised Learning

Unsupervised Learning

Without supervision! Has to learn patterns without a guide/target.

Supervised Learning has **Features** and a **Target**

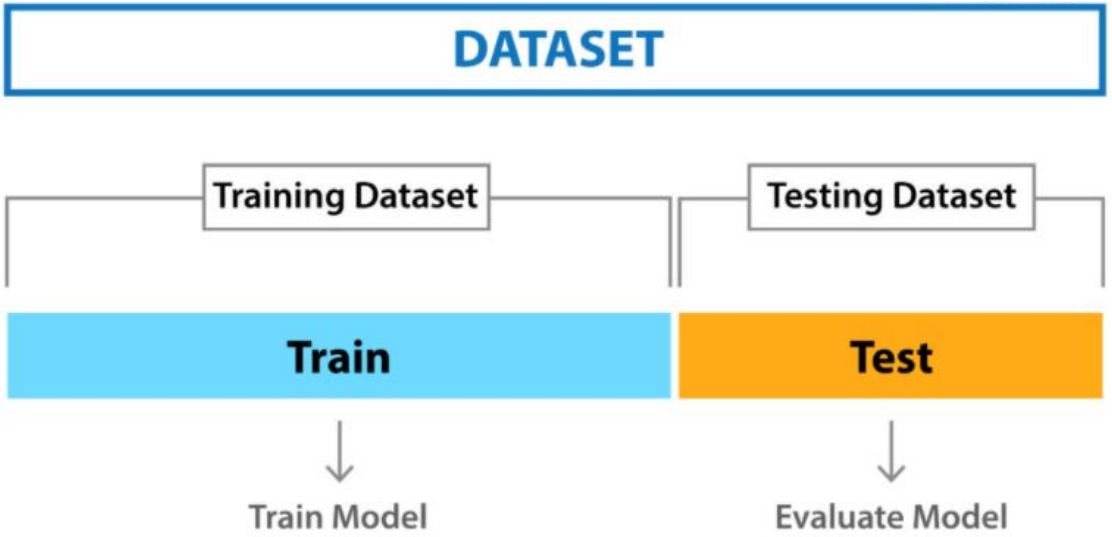


Model Validation

We trained our model by giving it data with the targets, but…

How do we know our model will be successful on **new data** with no targets?

Split data into a training set and a testing set



- Computer “sees” all the features AND the correct answers
- It “learns” and develops the model

- Computer only “sees” features (no answers)
- It makes a prediction
- We then compare prediction to our known “answer”
- This *simulates* making predictions when the answer is truly not known yet

Machine Learning in Python: Scikit-Learn
AKA
sklearn

- One stop shopping for machine learning tools
- Works great with Pandas and Numpy!
- LARGE library
- Only import what we want.
- from sklearn import <what you want>

```
[1]    import pandas as pd
       from sklearn.datasets import load_iris
```

Prepare the Data: The Iris Dataset

```
iris = load_iris()

df = pd.DataFrame(iris.data, columns = iris.feature_names)
df['target'] = iris.target
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Prepare the Data:
X = features, y = target

```
X = df.drop(columns='target')
y = df['target']
```

Model Validation in Python:

```
from sklearn.model_selection import train_test_split
# Validation Split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
```

In chat, answer: Why do we split the data into train and test sets?

Standard Scaler - Why We Scale

- We scale for certain machine learning models.
- To scale means to change the range of values but the distribution stays the same.
- Standardize - to scale the values so that the distribution has a standard deviation of 1 with a mean of 0.
- Scaled values lose their original units.
- To avoid data leakage, the scaler should only be fit on the training set.

Transforming Data in Python:
Standard Scaler (scales data)

```
from sklearn.preprocessing import StandardScaler

#Instantiate scaler object
scaler = StandardScaler()

#Fit scaler object
scaler.fit(X_train)

#Transform Data
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Fit on Train

Does NOT change in place.

Transform both Train and Test

Transforming Data in Python:
StandardScaler (scales data)

In sklearn:

- **.fit()** learns from data (means, medians, standard deviations, etc.)
- **.transform()** returns transformed data
- Testing data simulates UNSEEN data
- **Nothing should ever learn from testing data**
  - But it's okay to transform testing data after learning from training data.

Transforming Data in Python:
Standard Scaler (scales data)

In chat, answer:

- Why not just fit on all of X before splitting into X_train and X_test?

Model Validation in Python:

Can Prevent Data Leakage!

Thursday's Topics

1. Data Preparation
2. One Hot Encoding
3. Simple Imputer
4. Pipelines
5. Pipelines and Column Transformers

CodeAlong Notebook

Challenge

1. Import the data
2. Split columns into X features and y target
3. Split rows into training and testing sets
4. Use Standard Scaler to scale the data WITHOUT leaking data.

Challenge Notebook