Intro to Supervised Learning

Data Boot Camp

Lesson 19.1

Class Objectives

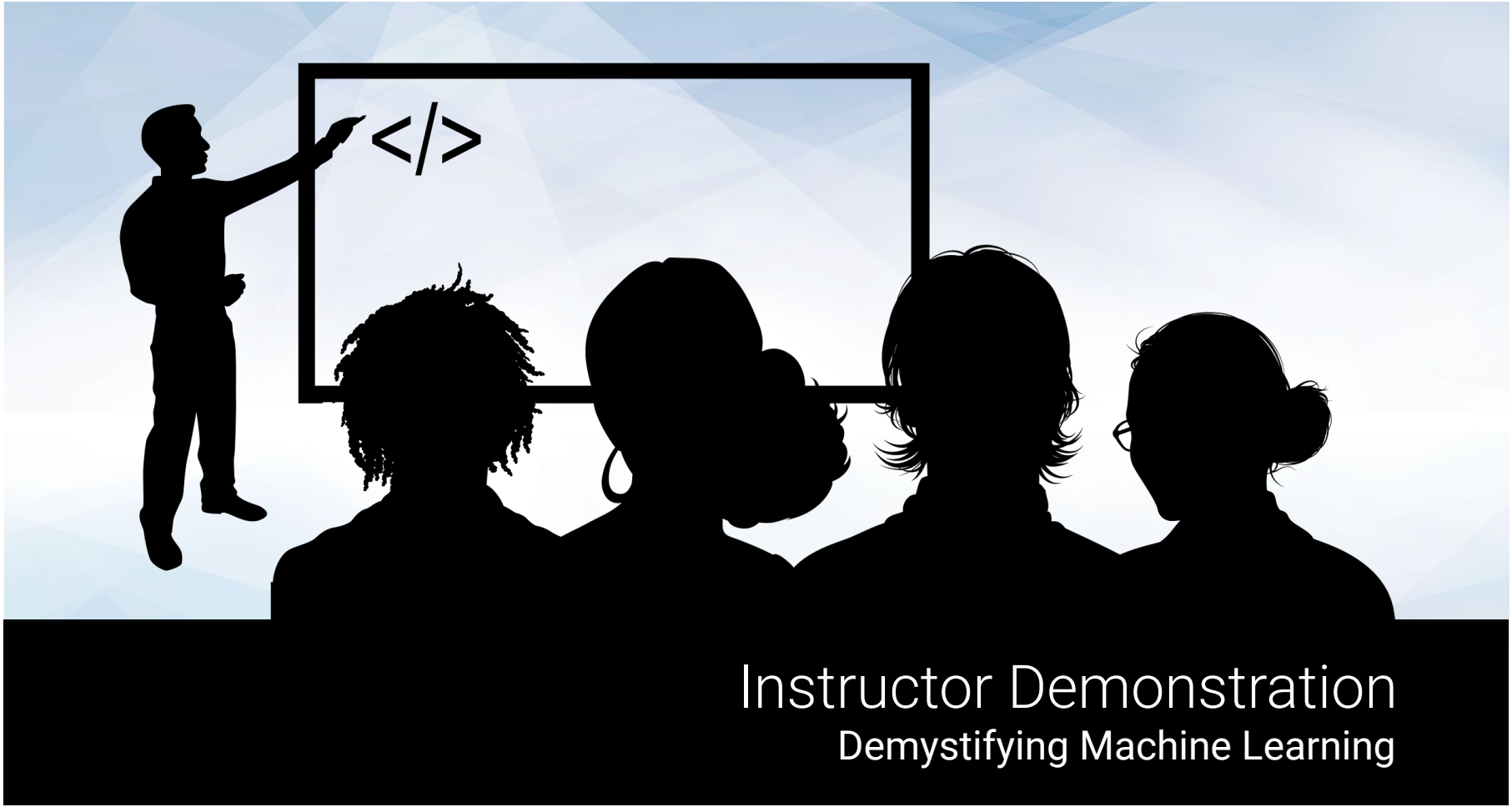
By the end of this lesson, you will be able to:

Explain how machine learning algorithms are used in data analytics.

Create training and testing sets from a specified data set.

Implement linear and logistic regressions by using scikit-learn.

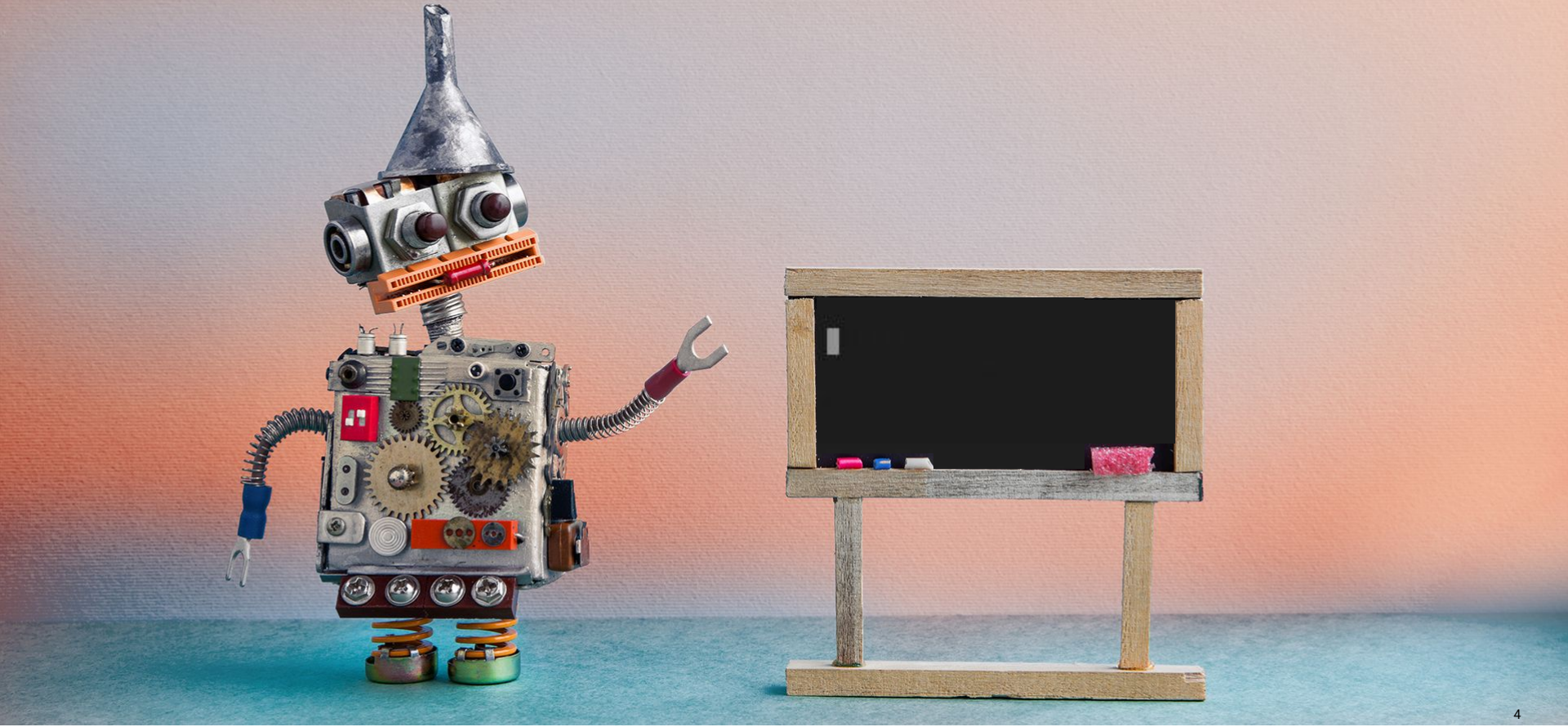
Create confusion matrices for classification outputs.



Instructor Demonstration

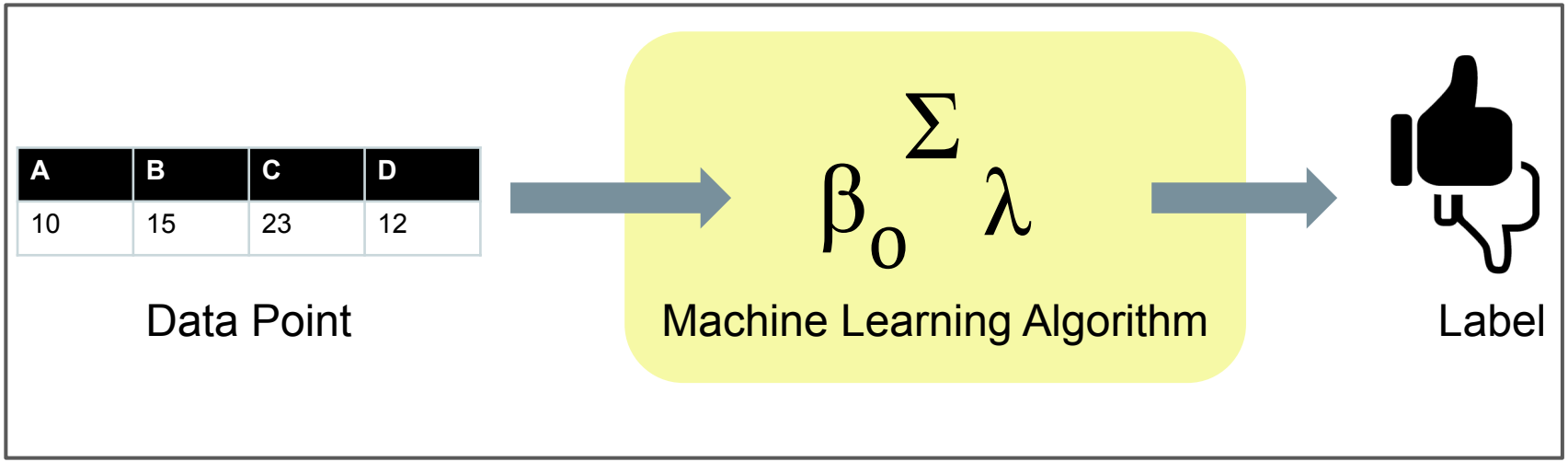
Demystifying Machine Learning

Instructor Do: Demystifying Machine Learning



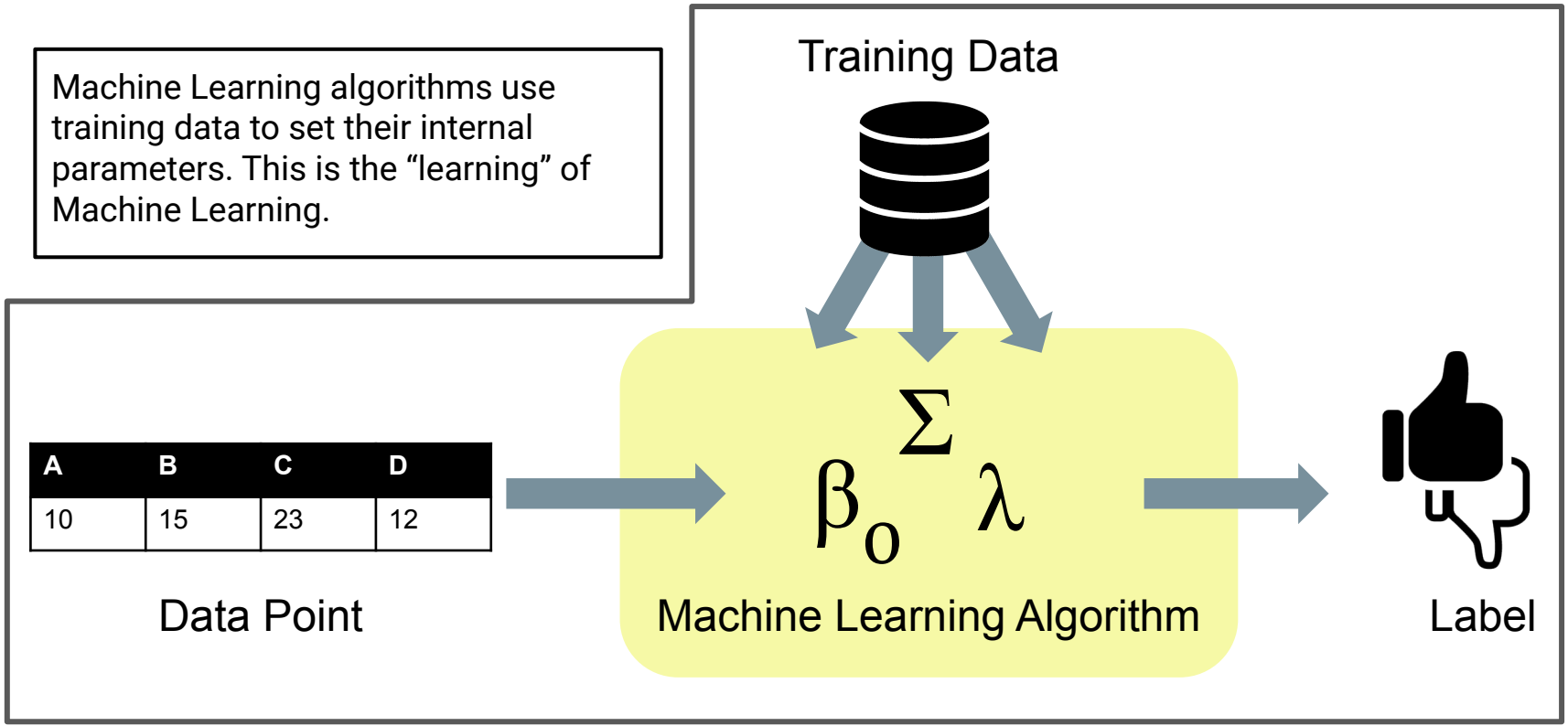
Instructor Do: Demystifying Machine Learning

Machine Learning algorithms are functions with internal parameters that apply labels to data points.



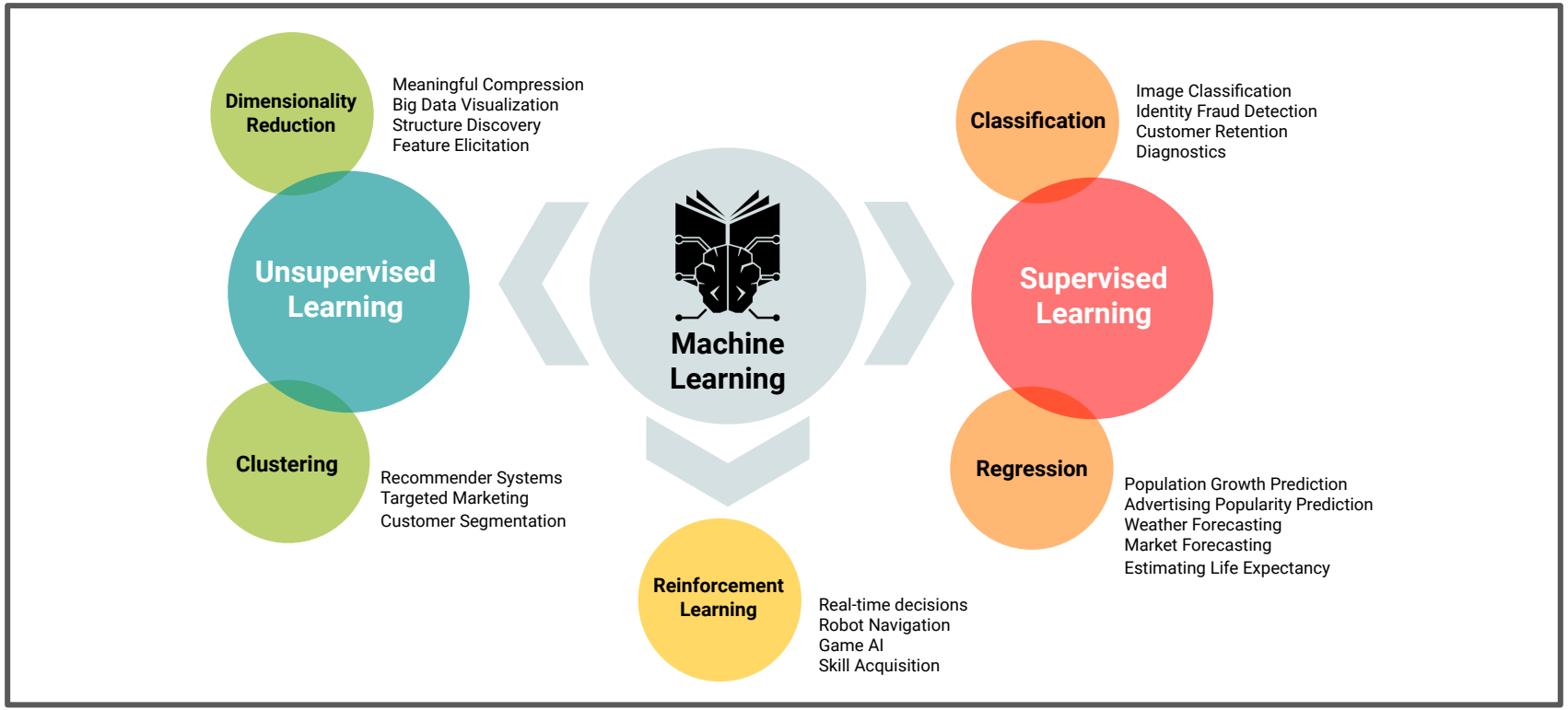
Instructor Do: Demystifying Machine Learning

Machine Learning algorithms use training data to set their internal parameters. This is the “learning” of Machine Learning.



Machine Learning Categories

Instructor Do: Demystifying Machine Learning



Instructor Do: Demystifying Machine Learning

➔ Succinctly, **supervised learning** is algorithms for which the potential outcomes are knowable in advance (i.e., category or numeric range) and can be used to correct the model's predictions.

01 Example

Using data such as credit score, credit history, income, etc., we are trying to predict whether an individual is a credit risk or not.

Known Category:
"Credit Risk" vs. "Not Credit Risk"

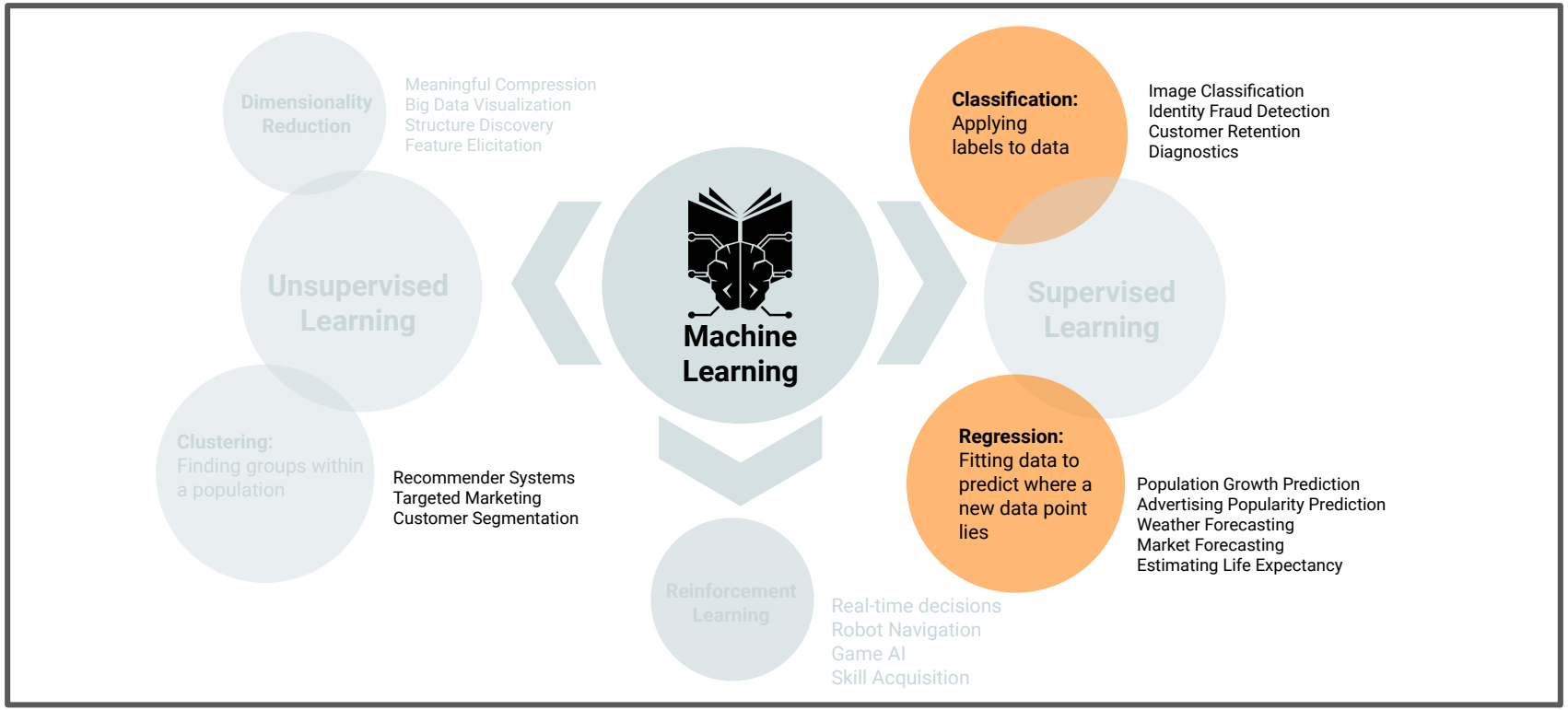
02 Example

Using features such as number of bedrooms, square feet, etc, we are trying to predict the market value of a house.

Numeric Range:
50,000 - 500,000

Supervised Learning Subcategories

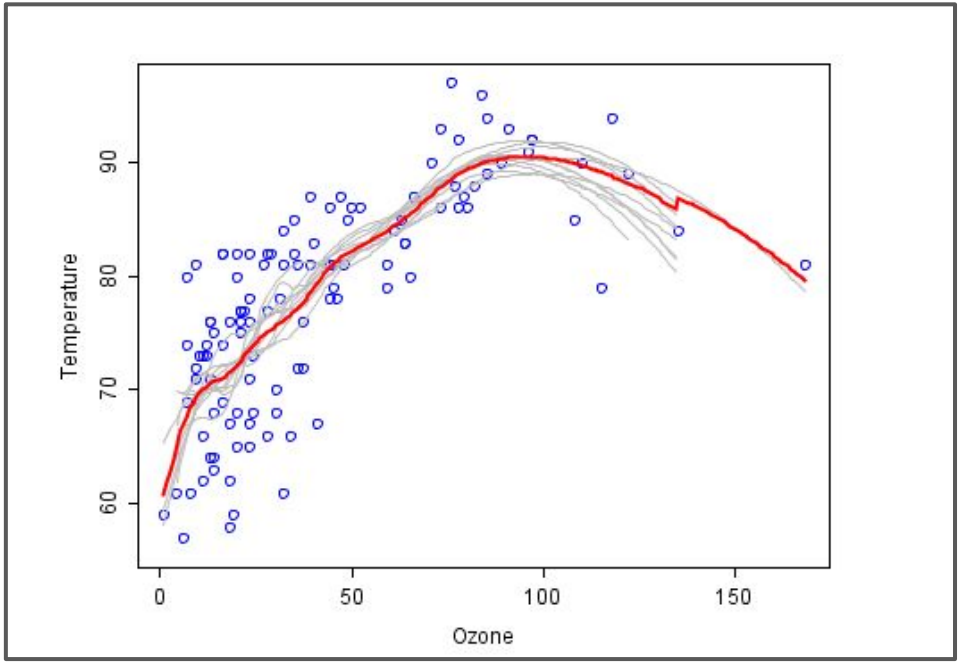
Instructor Do: Demystifying Machine Learning



Instructor Do: Demystifying Machine Learning

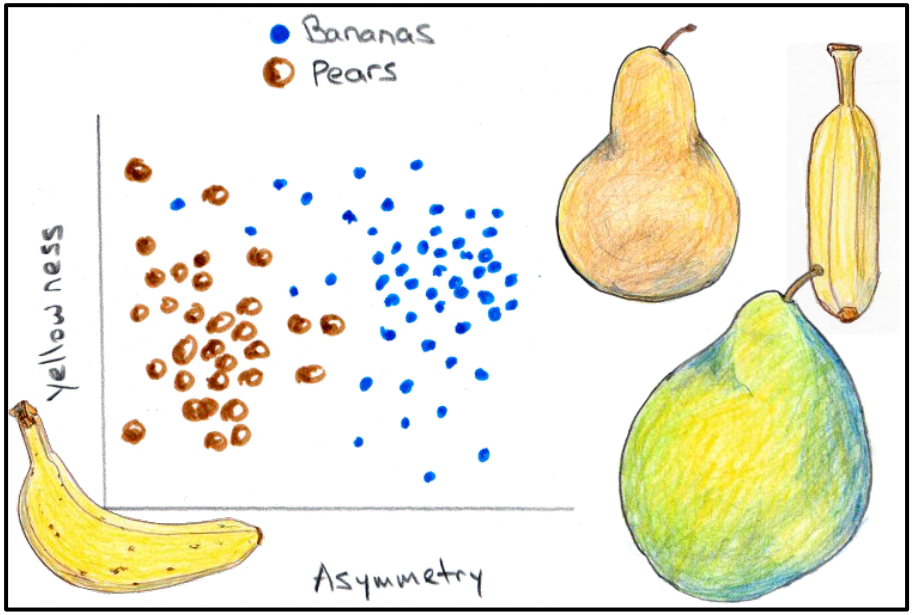
➔ **Regression**

We'll be revisiting regression to predict the location of data points based on old data.



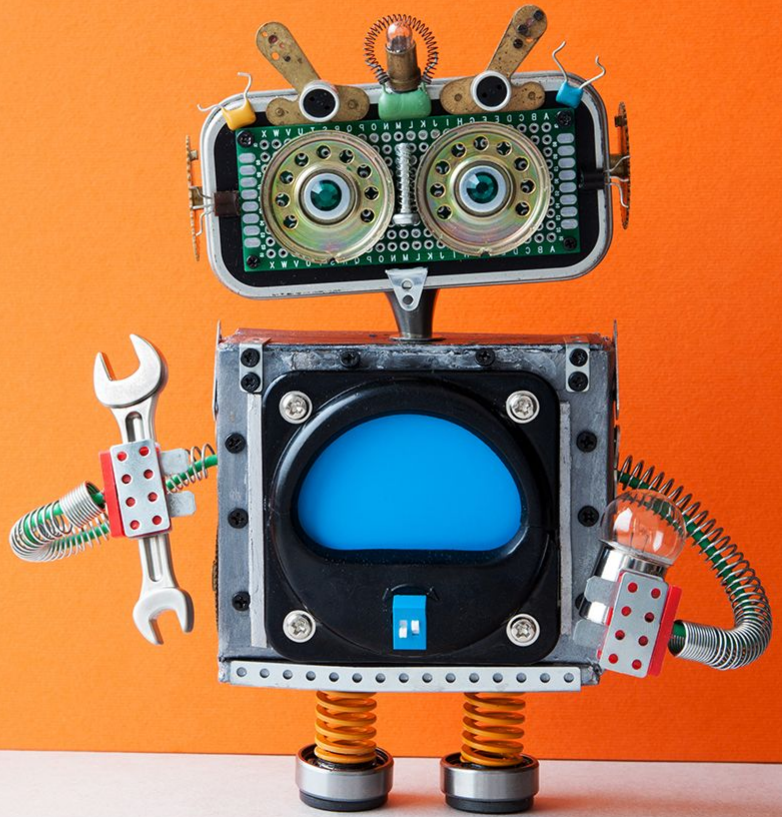
Instructor Do: Demystifying Machine Learning

➔ **Classification**



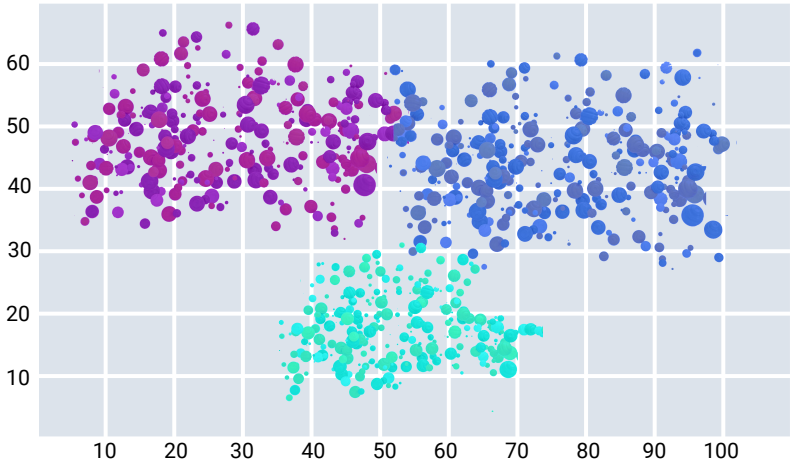
Instructor Do: Demystifying Machine Learning

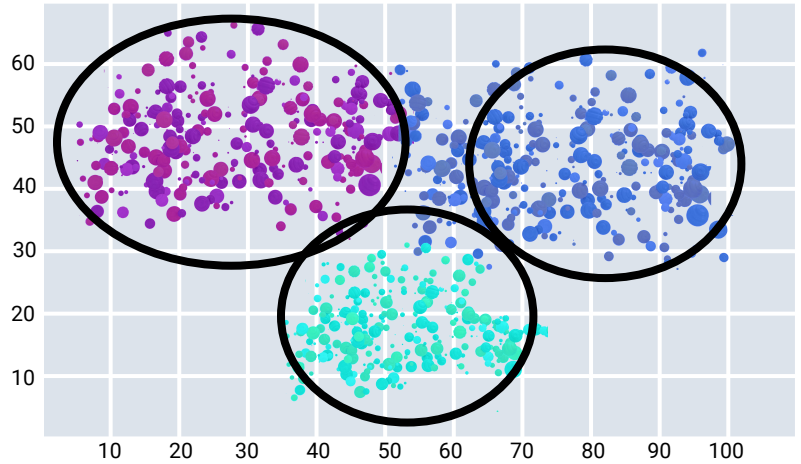
- **Algorithms for which the potential outcomes are unlabeled. Inferences are made directly from the data without feedback from known outcomes or labels.**



Instructor Do: Demystifying Machine Learning

- **Clustering**
  In this clustering problem, we expect our algorithm to group data points based on their mutual similarities of features.

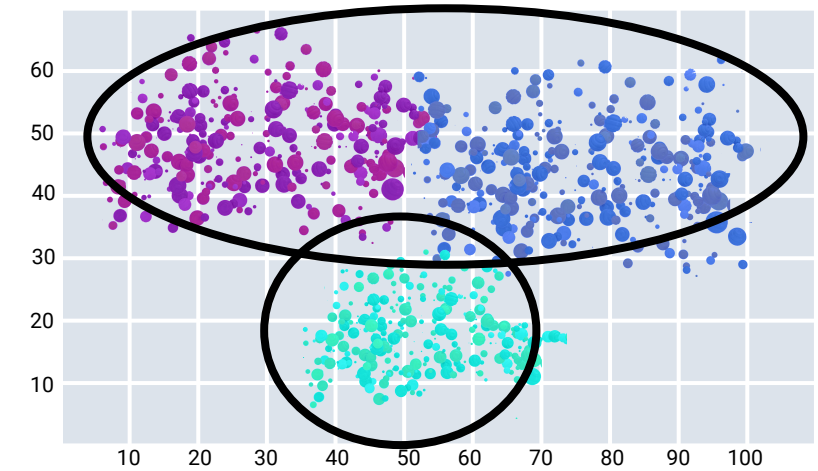




K=3

Instructor Do: Demystifying Machine Learning

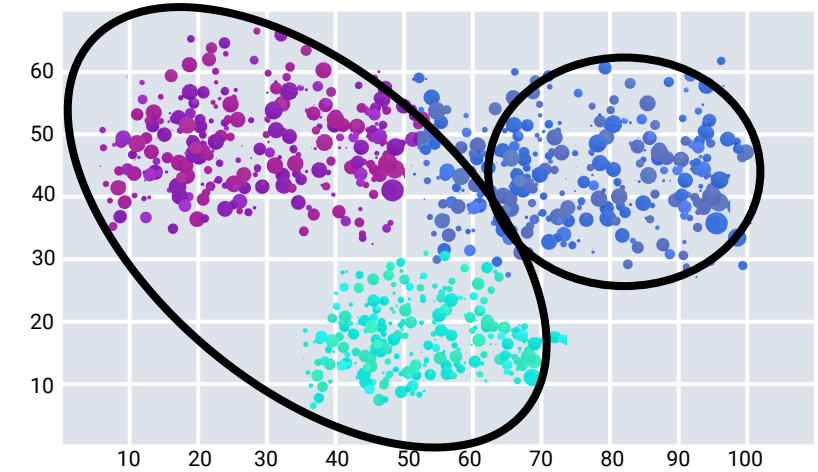
- **Clustering**
  But the problem is more complex.



K=2

Instructor Do: Demystifying Machine Learning

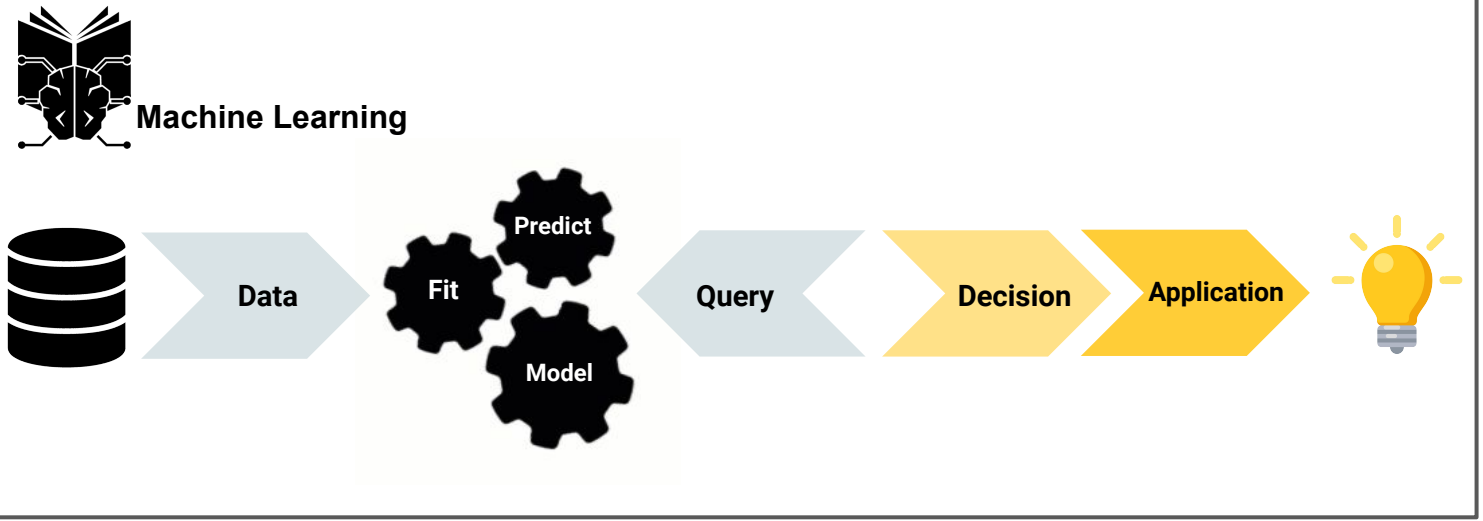
- **Clustering**
  Perhaps the clusters are not where we think they are.



K=2

Instructor Do: Demystifying Machine Learning

Regardless of the problem type, in Machine Learning we follow a familiar paradigm.

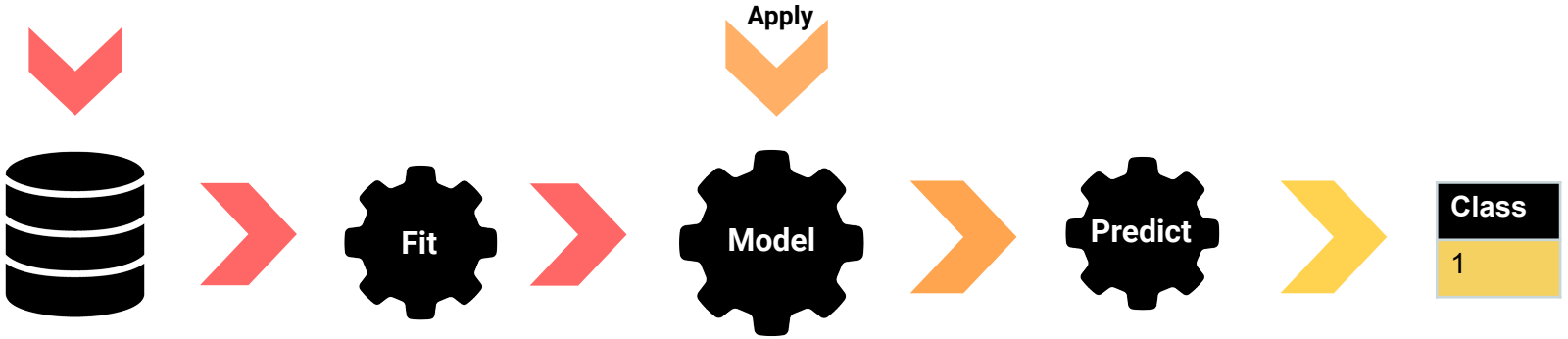


Instructor Do: Demystifying Machine Learning

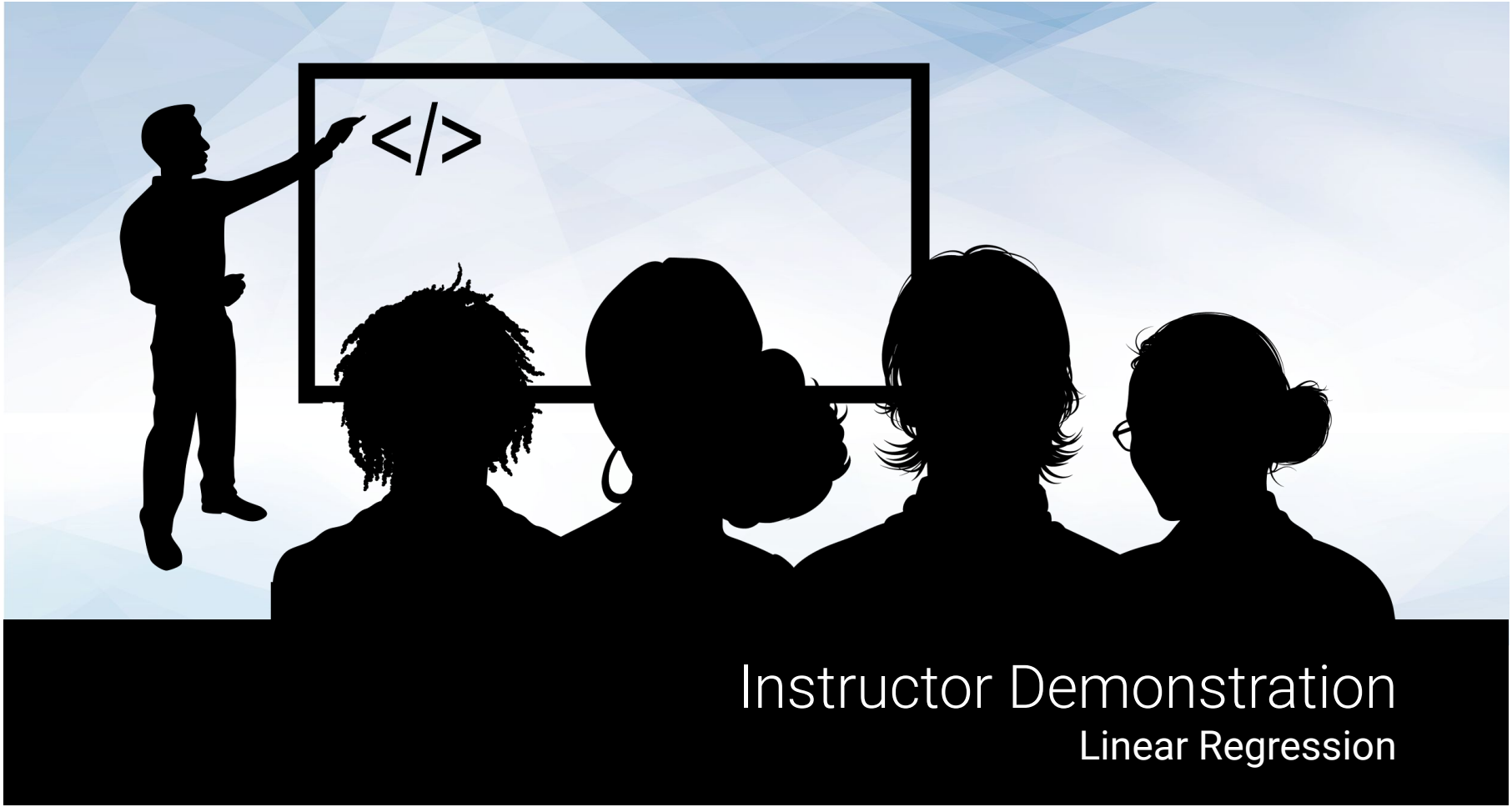
Regardless of the problem type, in Machine Learning we follow a familiar paradigm.

A	B	C	Class
11	16	22	1
10	8	4	2
…	…	…	…

A	B	C	Class
10	15	23	?



Questions?



Instructor Demonstration

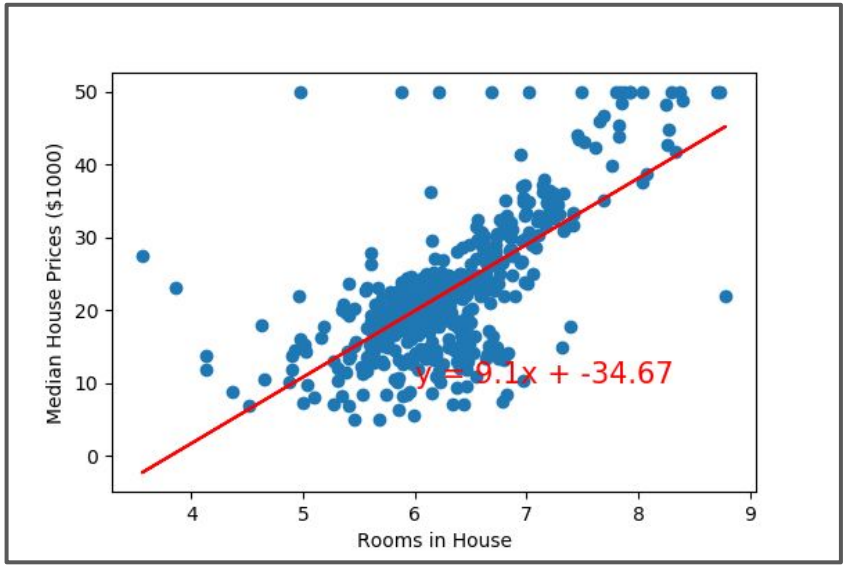
Linear Regression

What is **linear regression**?

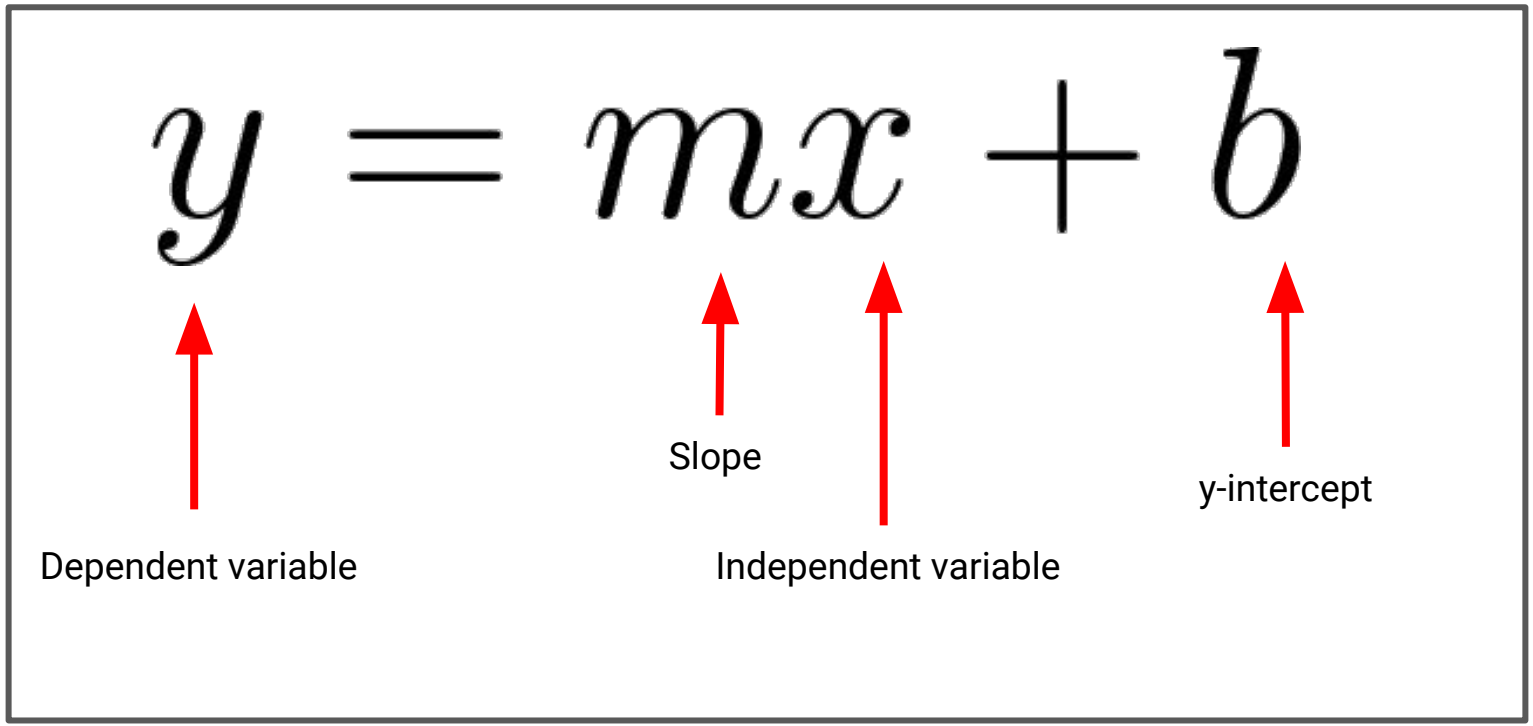
Instructor Do: Linear Regression

➔ Linear regression is used to model and predict a relationship.

- Predicts a dependent variable, given values from an independent variable.
- There are two basic types.
  - Simple linear regression.
  - Multiple linear regression.
- Both types predict an independent variable using the linear equation.

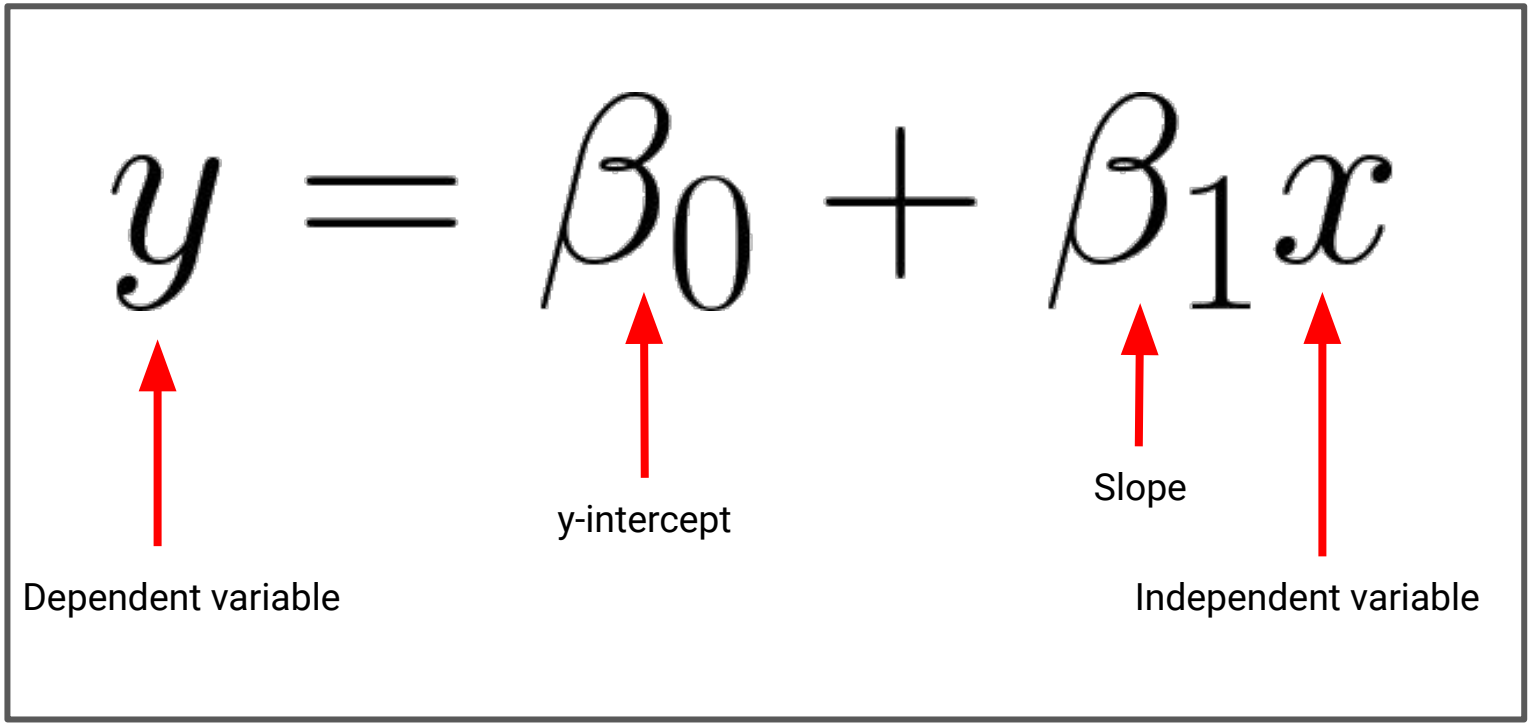


Instructor Do: Linear Regression

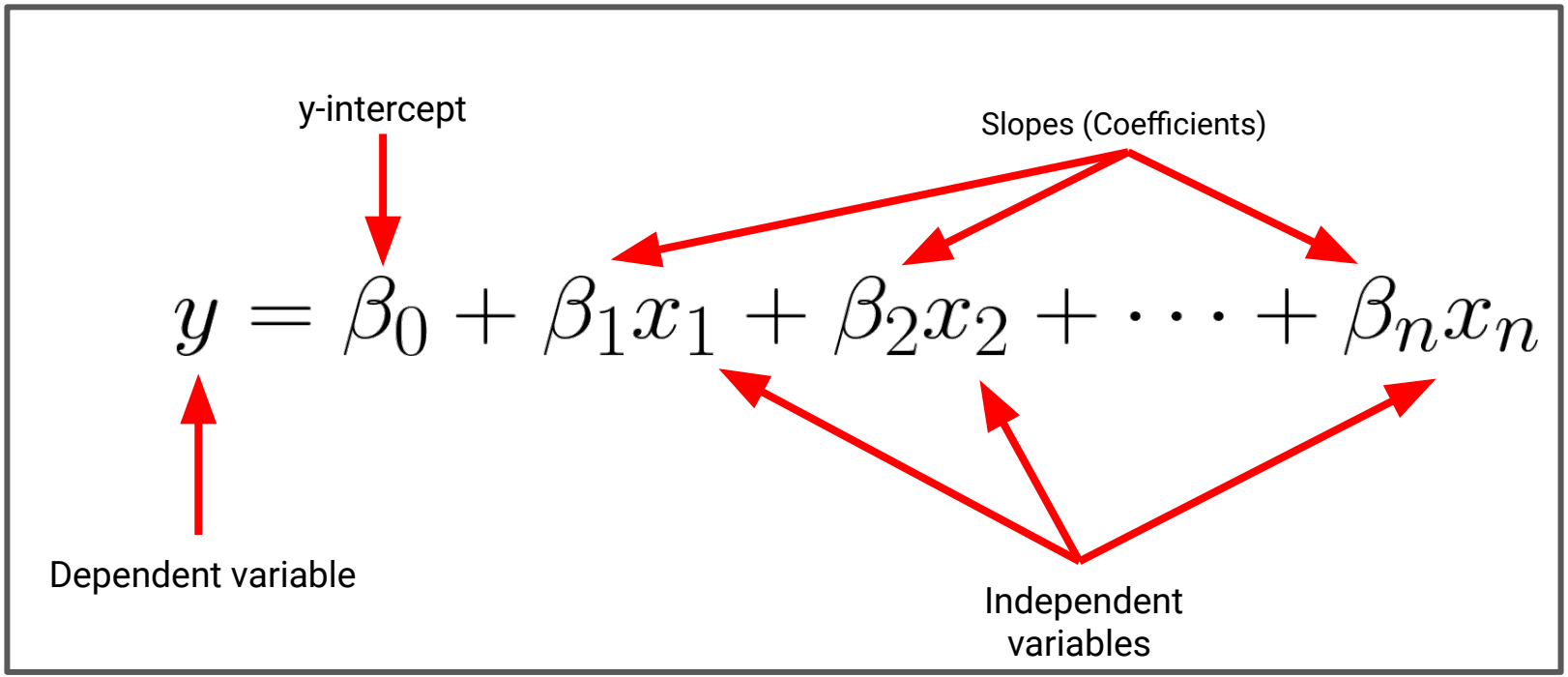


The equation of a line - Univariate in Greek!

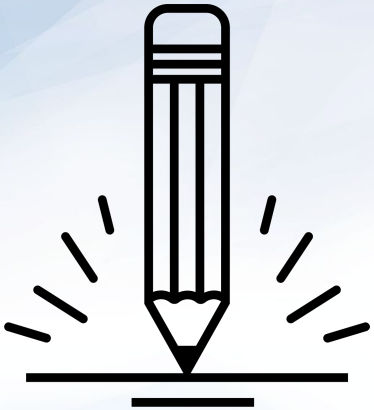
Instructor Do: Linear Regression



Instructor Do: Linear Regression



Linear regression is *FAST*



Activity: Linear Regression

In this activity, you will calculate two regression lines by using a dataset of California house prices. For the first regression line, you'll explore the relationship between median income levels and median home values. For the second regression line, you'll use all the available variables to predict the median home value.

Suggested Time:
15 Minutes

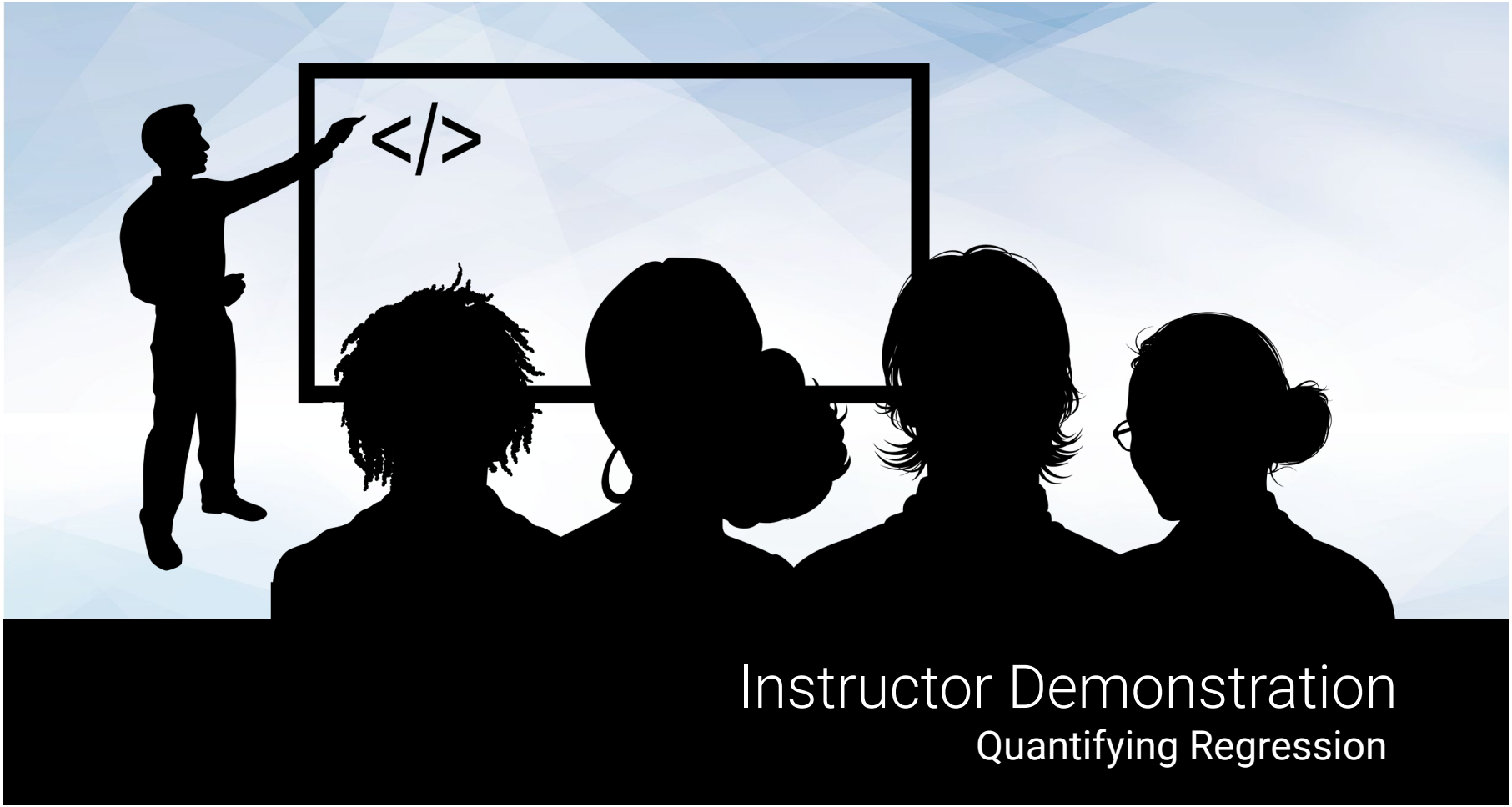
Instructions:

Activity: Linear Regression

- **For univariate regression:**
  - Load the housing data, and then separate the median income feature into one variable: `med_inc`.
  - Create a scatter plot of `med_inc` vs. `y` (median home values) to visually find out if any linear trend exists.
  - Use the linear regression model of Sklearn to fit the model to the data.
  - Print the weight coefficients and the y-axis intercept for the training model.
  - Calculate the `y_min` and `y_max` values by using `model.predict()`.
  - Plot the model fit line by using `[x_min[0], x_max[0]], [y_min[0], y_max[0]]`.

- **For multivariate regression**:
  - Use the linear regression model of Sklearn to perform multiple linear regression by using all eight features for `X` and median home value for `y`.
  - Compute the R2 score for the training and the testing data separately.
  - Plot the residuals for the training and the testing data.

Let's Review



Instructor Demonstration

Quantifying Regression

Common Scoring Metric

Instructor Do: Quantifying Regression

01

R^2 (R-Squared):

This is the baseline metric that many ML tools report on score. Higher R^2 values signify that the model is “highly predictive.” An R^2 value of >0.90 means that our model roughly accounts for 90% of the variability of the data.

02

MSE (Mean Squared Error):

This measures the average of the squares of the errors or deviations.

Instructor Do: Quantify Regression

We will cut a slice of this data (80%) to build our model, and then use this slice to predict the values for the remaining 20%.

Full Data Set (Historic)

N=1000

# bedrooms	# baths	Sq. feet (k)	Price (k)
2	1	1	200
3	2	1.5	250
…	…	…	…

Training Data Set

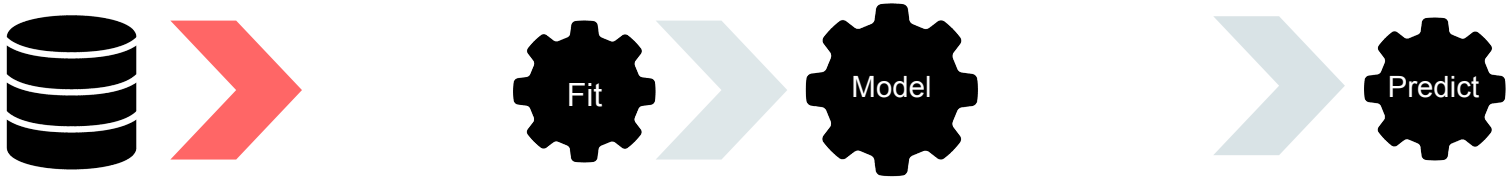
N=800

# bedrooms	# baths	Sq. feet (k)	Price (k)
4	3.5	3.2	450
2	2	1.5	220
…	…	…	…

Testing Data Set

N=200

# bedrooms	# baths	Sq. feet (k)	Price (k)
1	1	.5	60
5	3.5	4.2	780
…	…	…	…



Instructor Do: Quantifying Regression

We use the training data to fit the model to the data. This is the training step where we build a model that can predict our output (home price) for a given set of features (# bedrooms, # baths, square feet). Once the model is trained, we can use the model to make predictions.

Full Data Set (Historic)

N=1000

# bedrooms	# baths	Sq. feet (k)	Price (k)
2	1	1	200
3	2	1.5	250
…	…	…	…

Training Data Set

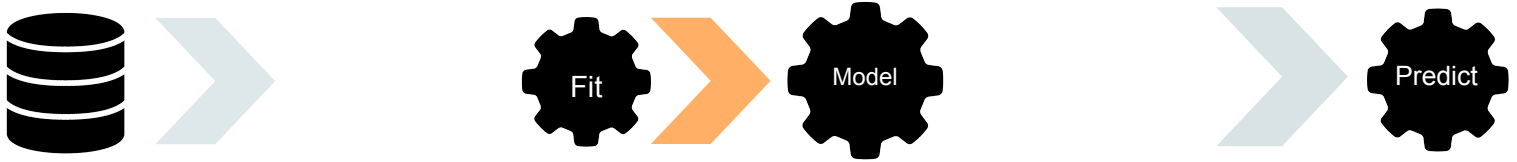
N=800

# bedrooms	# baths	Sq. feet (k)	Price (k)
4	3.5	3.2	450
2	2	1.5	220
…	…	…	…

Testing Data Set

N=200

# bedrooms	# baths	Sq. feet (k)	Price (k)
1	1	.5	60
5	3.5	4.2	780
…	…	…	…



Instructor Do: Quantifying Regression

We use the test data to make new home price predictions. We can then compare the home price of our prediction vs. the actual price. Based roughly on how often we are "correct," we get a score for the model as a whole. If the model scores well, we can trust it for future use. We train the model on the training data and score the model based on data that it has never seen before (test data).

Full Data Set (Historic)

N=1000

# bedrooms	# baths	Sq. feet (k)	Price (k)
2	1	1	200
3	2	1.5	250
…	…	…	…

Training Data Set

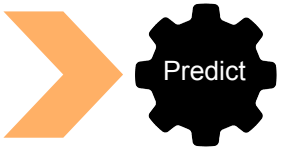
N=800

# bedrooms	# baths	Sq. feet (k)	Price (k)
4	3.5	3.2	450
2	2	1.5	220
…	…	…	…

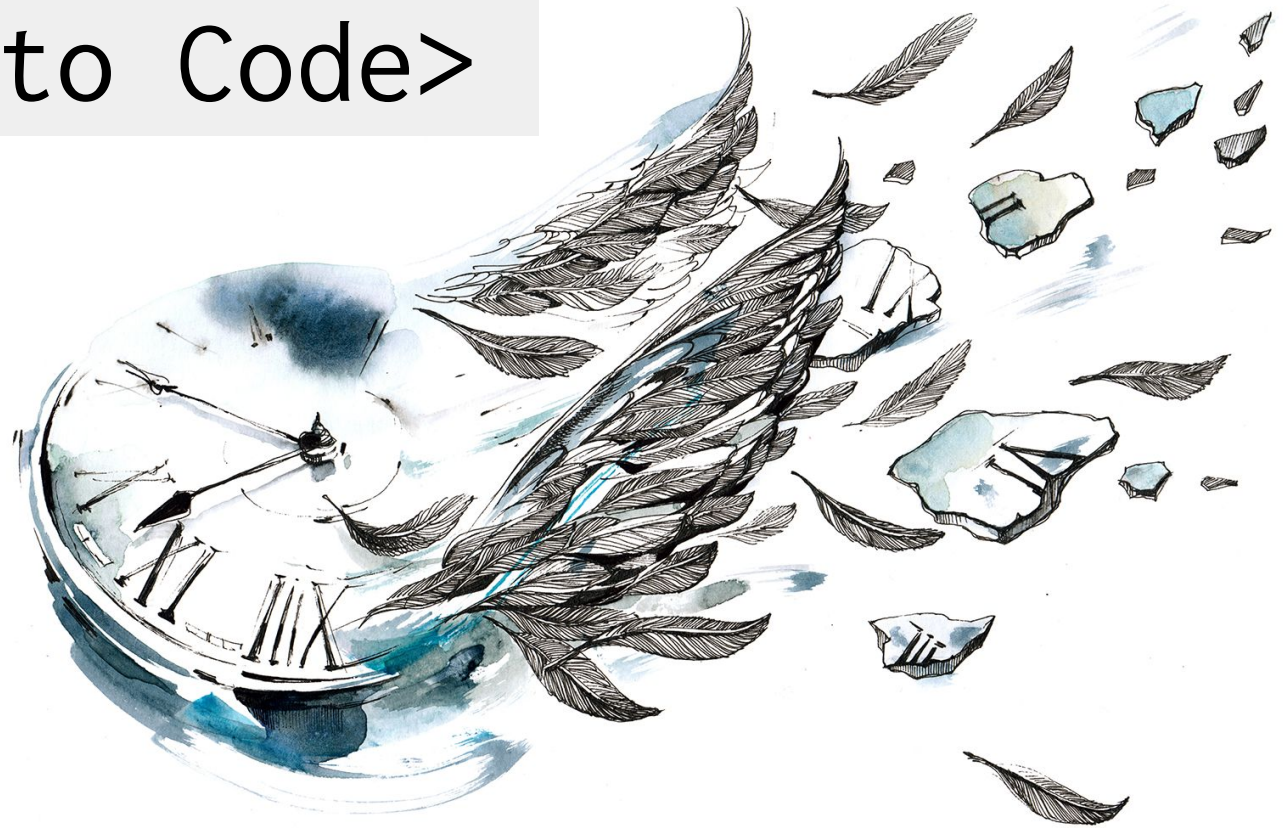
Testing Data Set

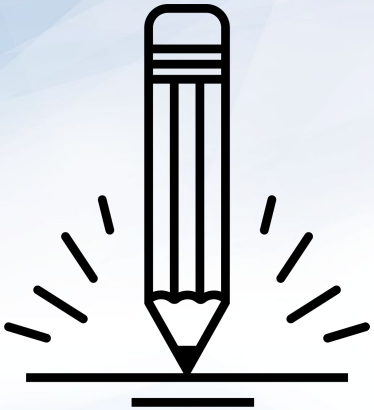
N=200

# bedrooms	# baths	Sq. feet (k)	Price (k)
1	1	.5	60
5	3.5	4.2	780
…	…	…	…



<Time to Code>





Activity: Brains!

In this activity, you will calculate a regression line to predict head size vs. brain weight.

Suggested Time:
15 Minutes

Instructions:

Activity: Brains!

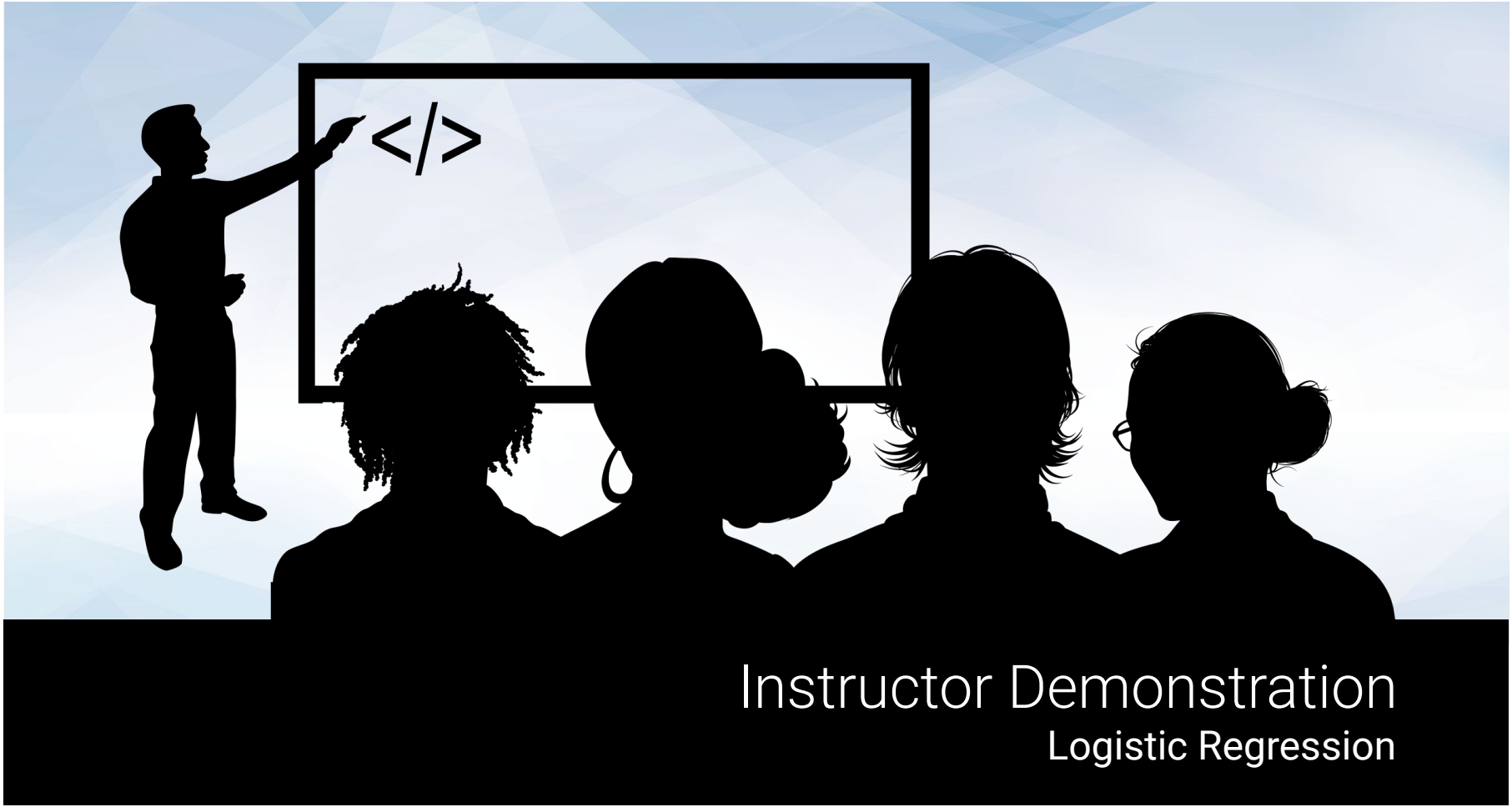
1. Start by creating a scatter plot of the data to visually find out if any linear trend exists.
2. Split the data into training and testing data by using the sklearn `train_test_split()` function.
3. Use the linear regression model of sklearn to fit the model to the training data.
4. Use the test data to make new predictions. Calculate the mean squared error (MSE) and the R-squared (R2) score for those predictions.
5. Use `model.score()` to calculate the R2 score for the test data.

Let's Review

Countdown timer

15:00

(with alarm)



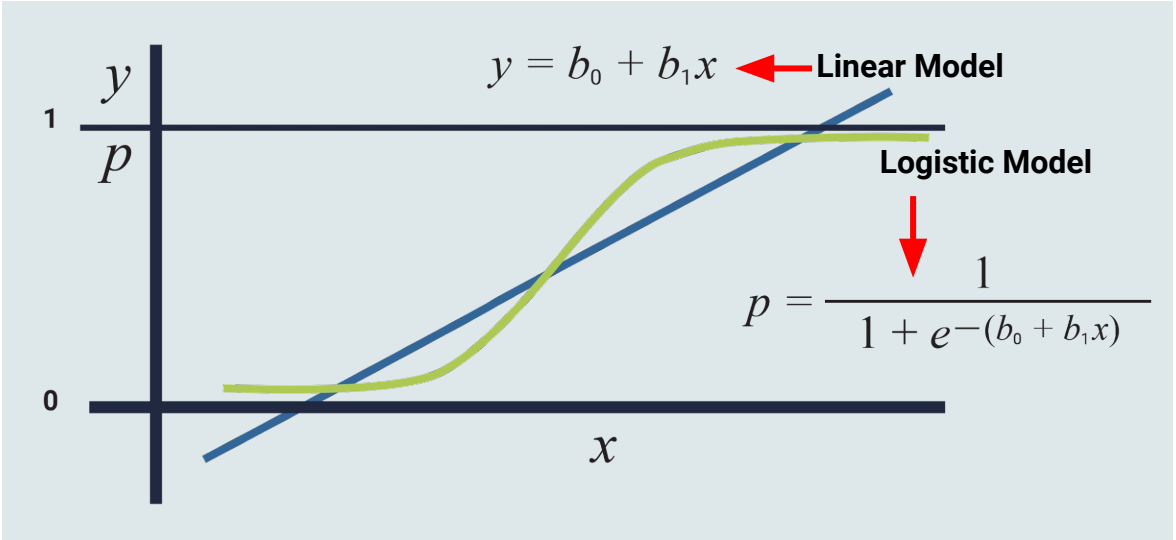
Instructor Demonstration

Logistic Regression

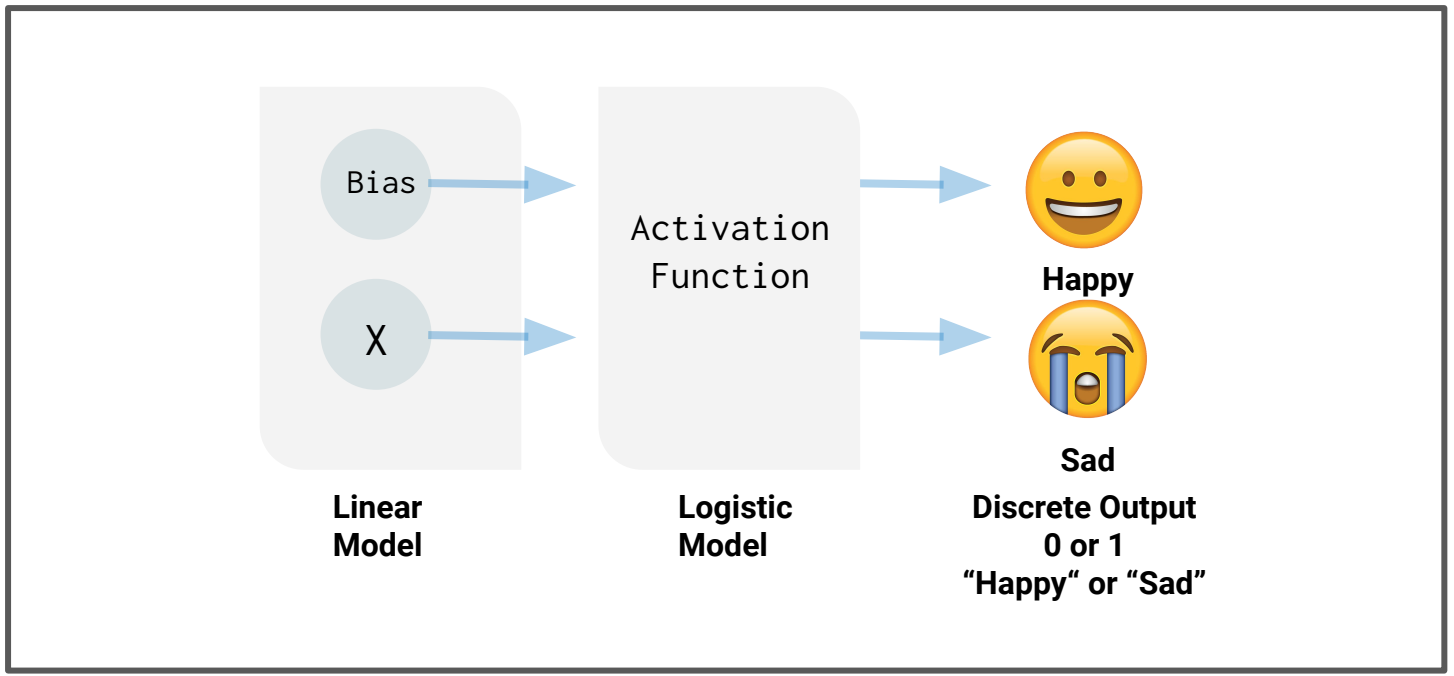
Instructor Do: Logistic Regression

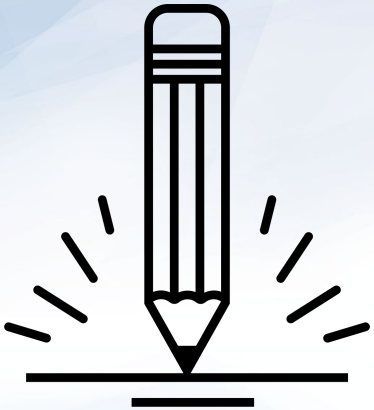
Logistic regression is a classification algorithm used to predict a discrete set of classes or categories (e.g., Yes/No, Young/Old, Happy/Sad).

Unlike linear regression, which outputs continuous numerical values (for example, age), logistic regression applies an activation function, such as the sigmoid function, to return a probability value of 0 or 1. This can then be mapped to a discrete class like “Young” or “Old.”



Instructor Do: Logistic Regression





Activity: Counterfeit Catcher

In this activity, you will apply logistic regression to predict whether a particular note is counterfeit or legitimate by using computed features from digitized images.

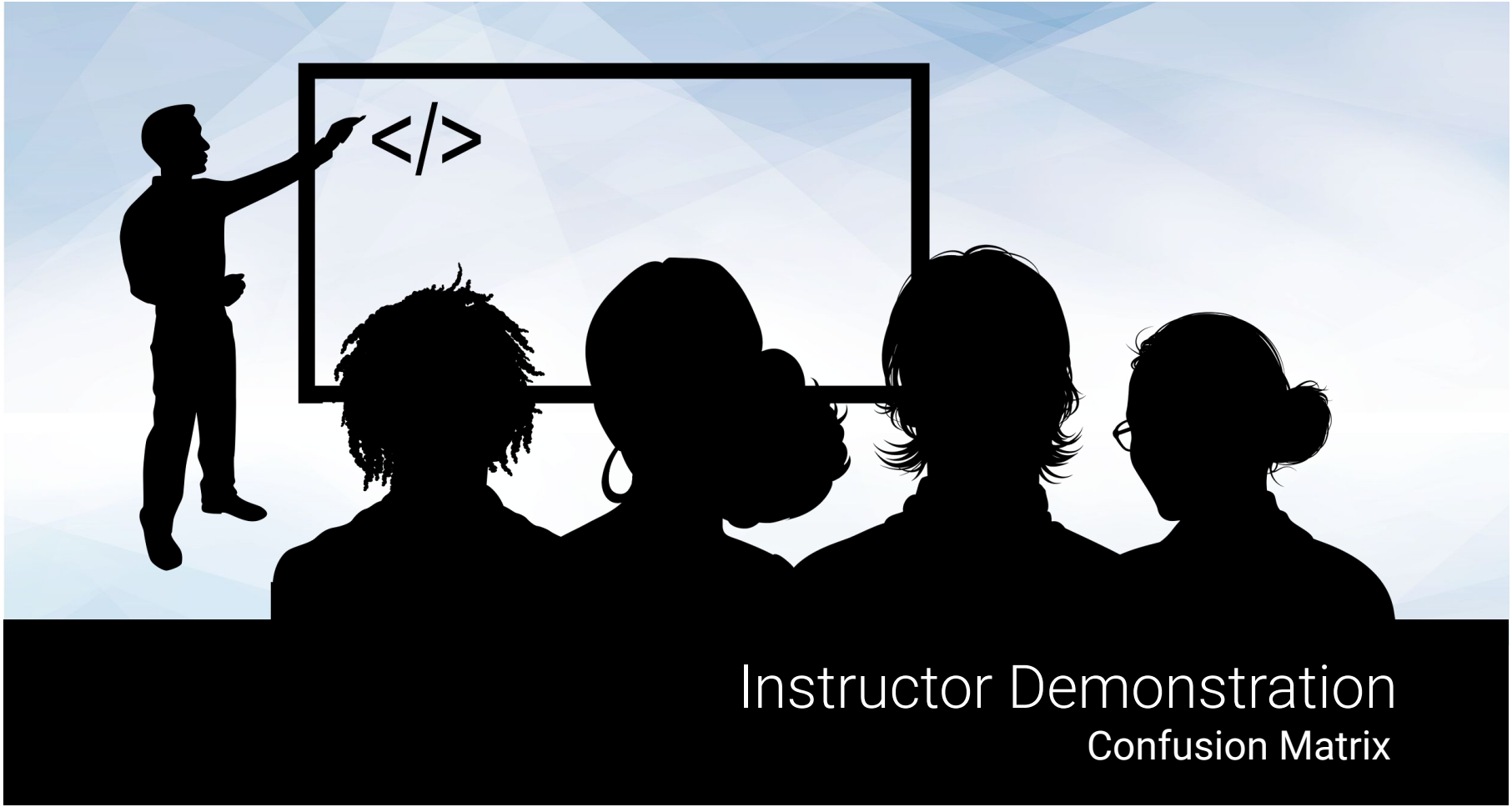
Suggested Time:
20 Minutes

Instructions:

Activity: Counterfeit Catcher

1. Split your data into training and testing data.
2. Create a logistic regression model with sklearn.
3. Fit the model to the training data.
4. Make 10 predictions, and then compare them to the testing data labels.
5. Compute the accuracy score for the training data and the testing data separately.

Let's Review



Instructor Demonstration

Confusion Matrix

Instructor Do: Confusion Matrix

- A Confusion Matrix compares the predicted values from a model against the actual values. The entries of the confusion matrix are the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

	Predicted True	Predicted False
Actually True	113 (True Positives)	12 (False Negatives)
Actually False	31 (False Positives)	36 (True Negatives

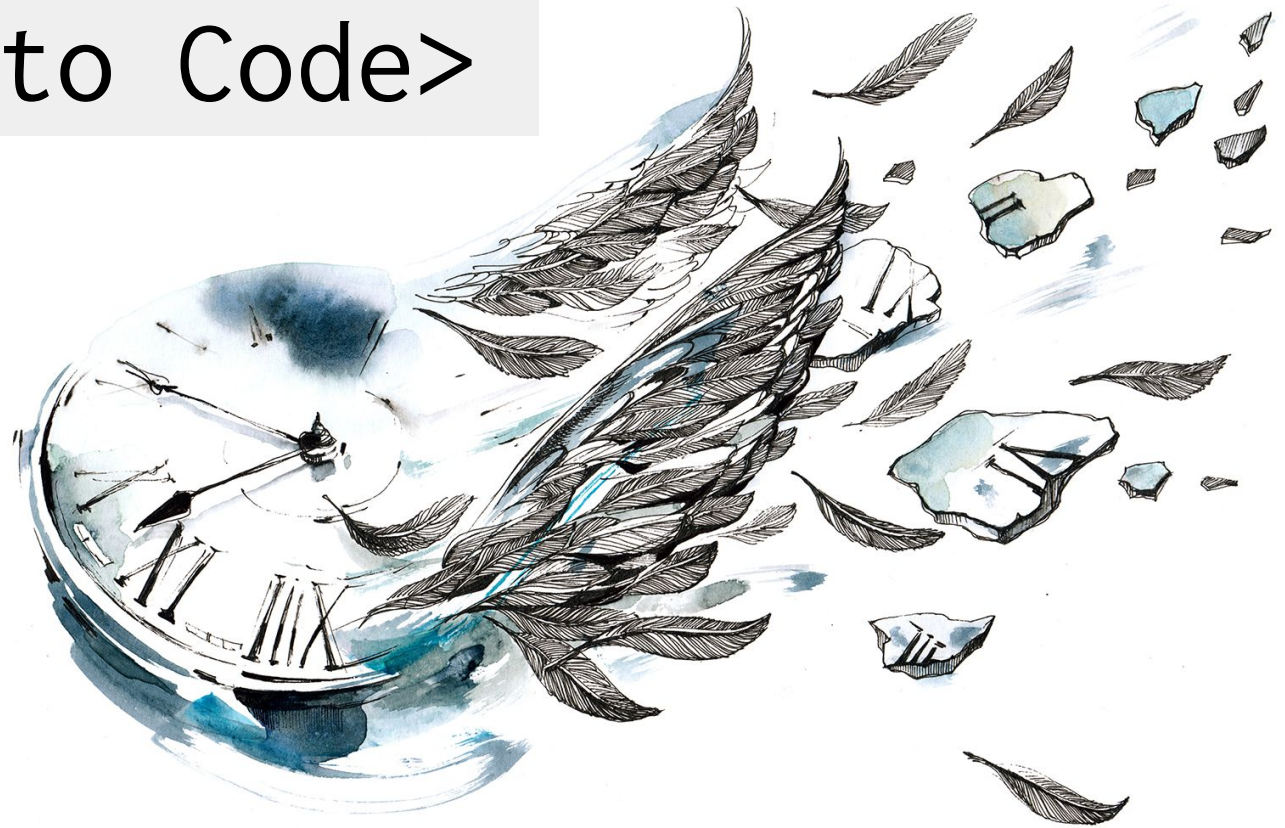
Instructor Do: Confusion Matrix

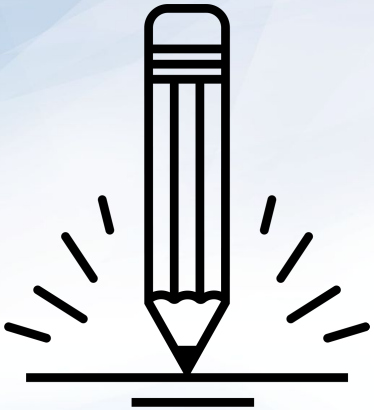
- We can calculate measures like the accuracy of a model from the values in the confusion matrix.

$$\text{Accuracy} = TP + TN / (TP + TN + FP + FN)$$

- We can also calculate other measures that gives us more information that just the accuracy, such as:
  - Precision
  - Sensitivity
  - F1 Score
  - Positive Predictive Value
  - Negative Predictive Value
  - Threat Score
  - False Omission Rate

<Time to Code>





Activity: Create a Confusion Matrix

In this activity, you will create a confusion matrix from the results of the previous activity and then manually calculate the accuracy of the model.

Suggested Time:
10 Minutes

Instructions:

Activity: Create a Confusion Matrix

1. Open Stu_Confusion_Matrix.ipynb (or use the solved notebook from the previous activity), and then create a confusion matrix for the testing dataset.
2. From the values in the confusion matrix, manually calculate the accuracy of the model.

Let's Review