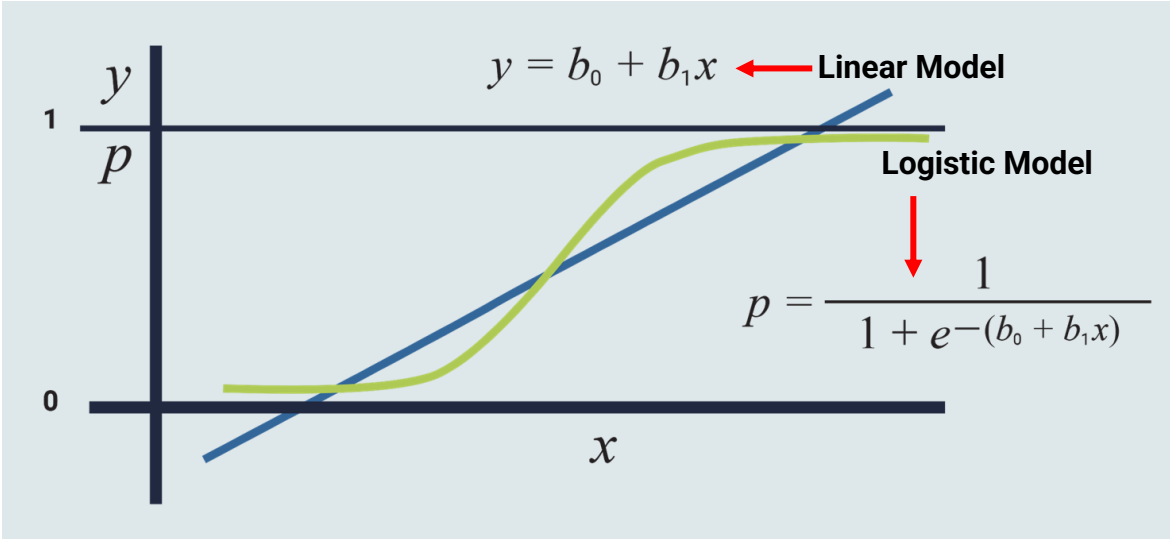
Logistic Regression

Logistic Regression

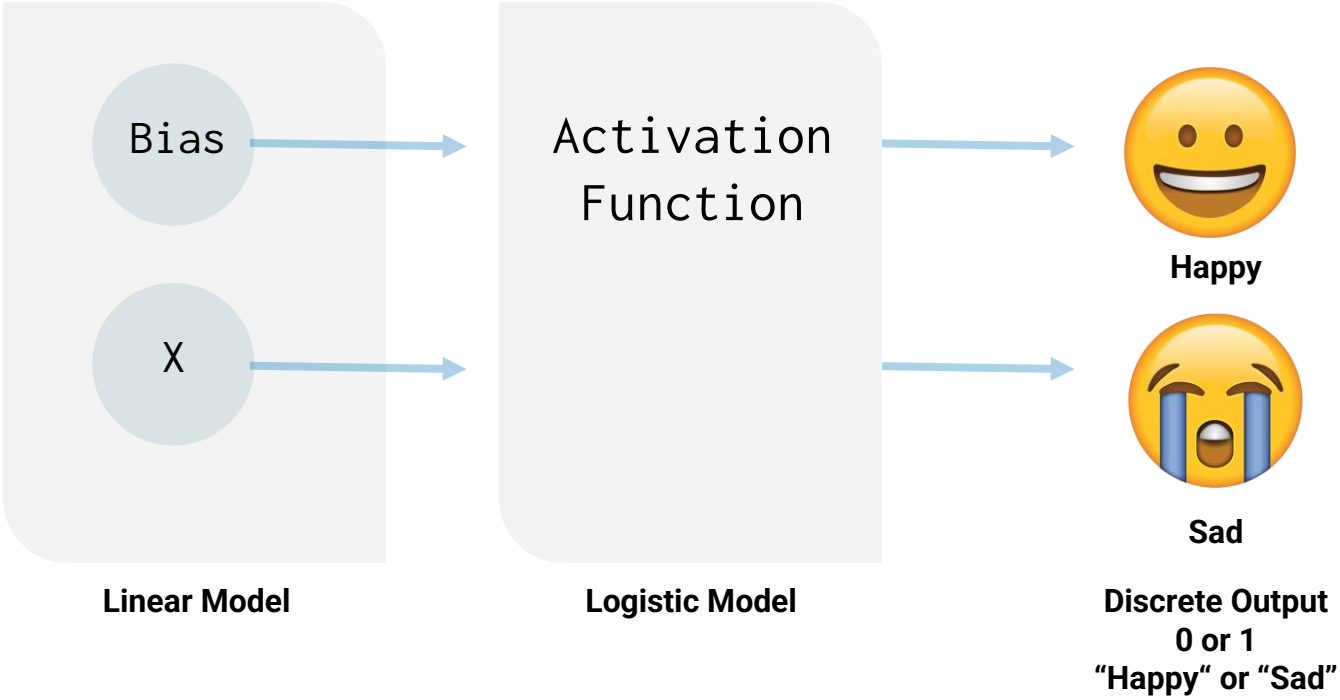
Logistic regression is a classification algorithm used to predict a discrete set of classes or categories (e.g., Yes/No, Young/Old, Happy/Sad).

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.”

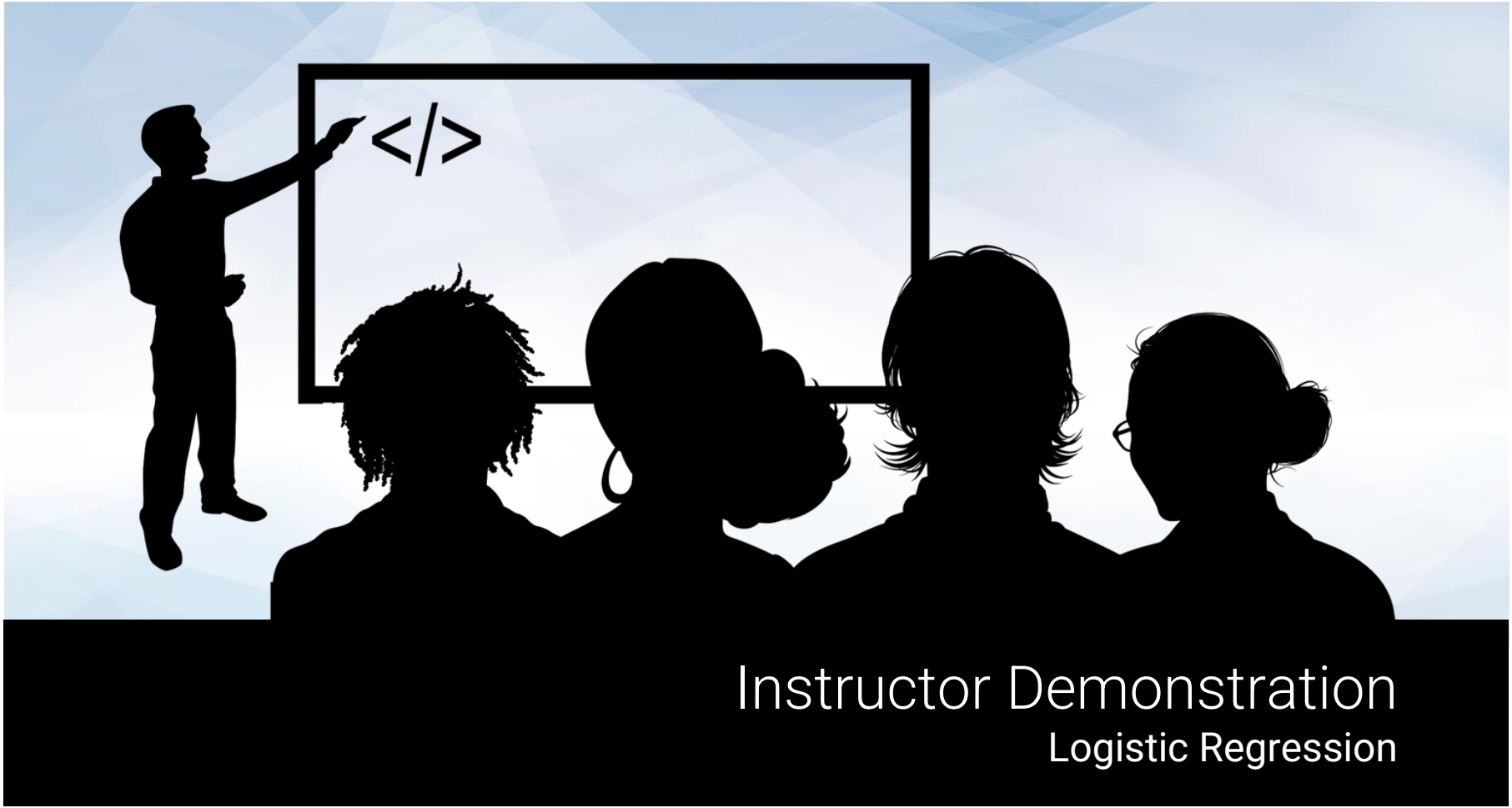


Logistic Regression

Logistic Regression is a statistical method to predict a discrete output or category.



Questions?



Instructor Demonstration

Logistic Regression

Activity: Voice Recognition

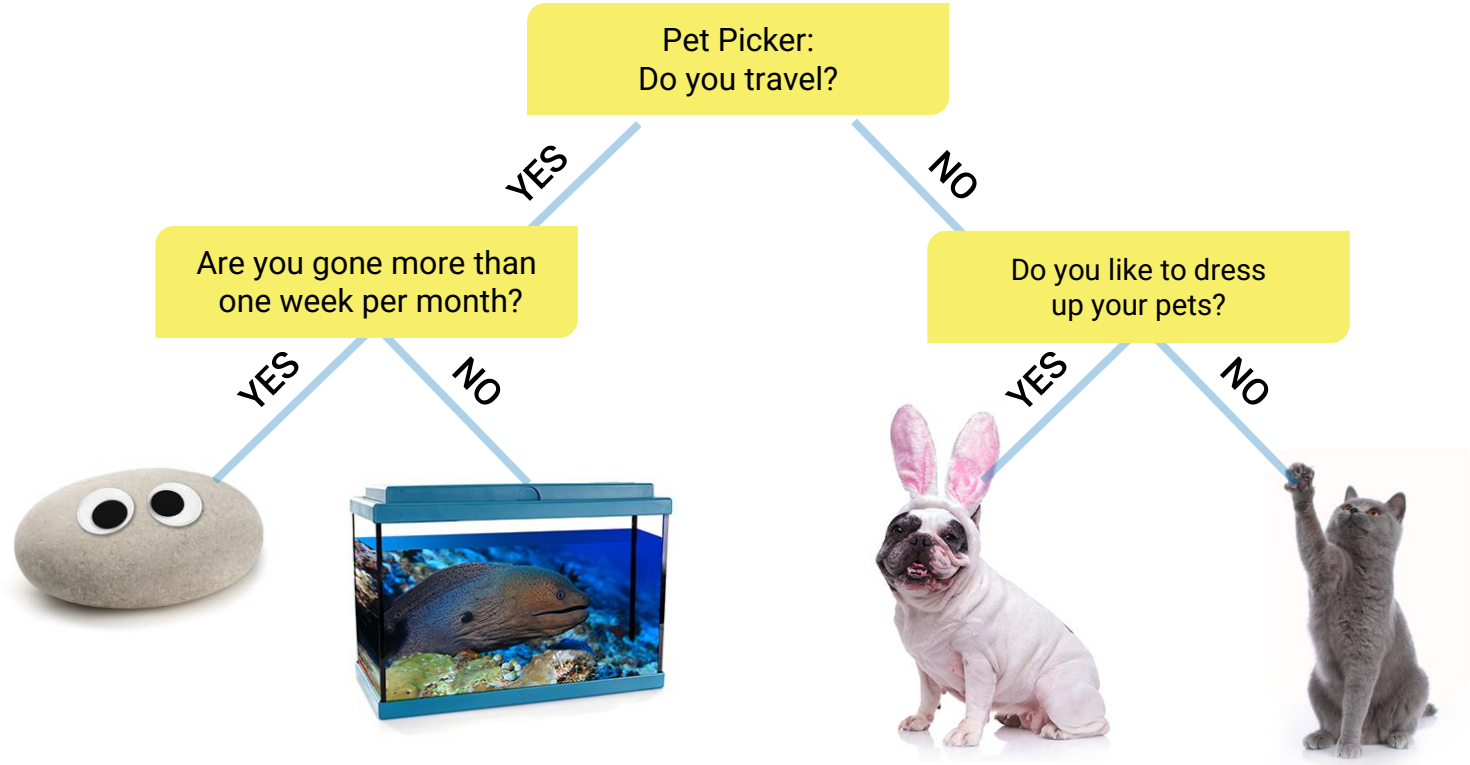
In this activity, you will apply logistic regression to predict the gender of a voice using acoustic properties of the voice and speech.

Suggested Time:
20 Minutes

Decision Trees & Random Forests

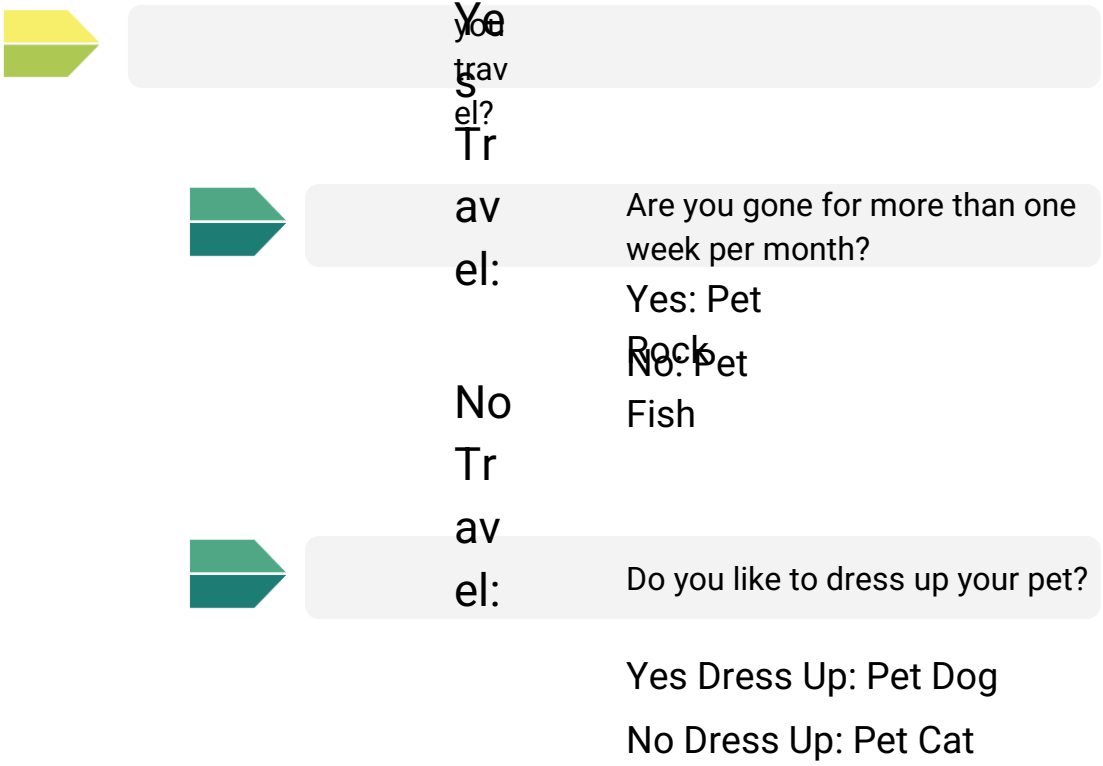
Decision Trees

Decision trees encode a series of true/false questions.



Decision Trees

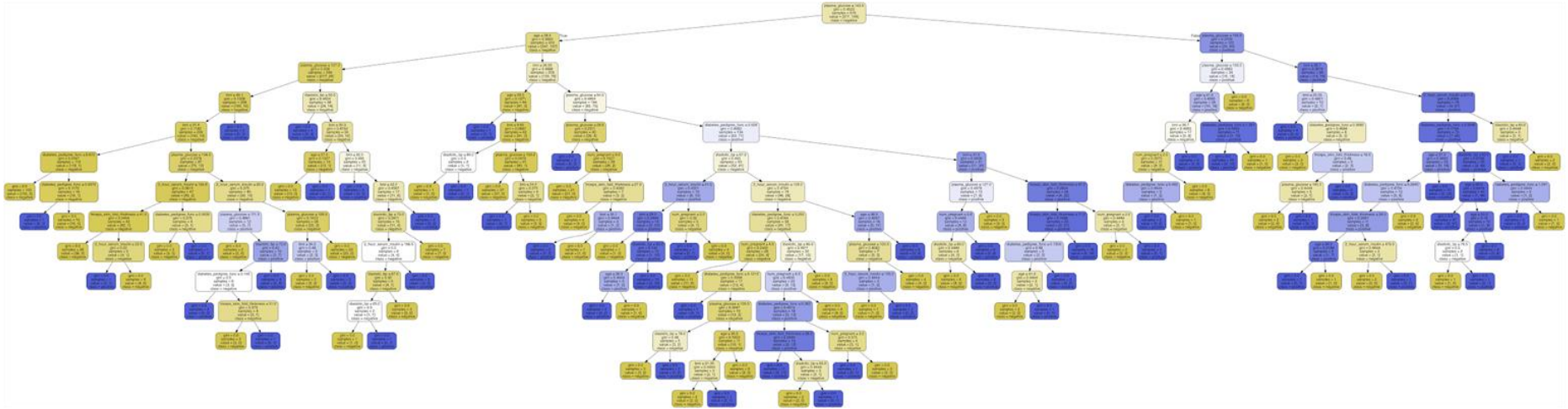
These true/false questions can be represented with a series of if/else statements




```
if (travel):
  if (time > week):
    print("Rock")
  else:
    print("Fish")
else:
  if (dress_up):
    print("Dog")
  else:
    print("Cat")
```

Decision Tree Complexity

Decision trees can become very complex and may not generalize well.

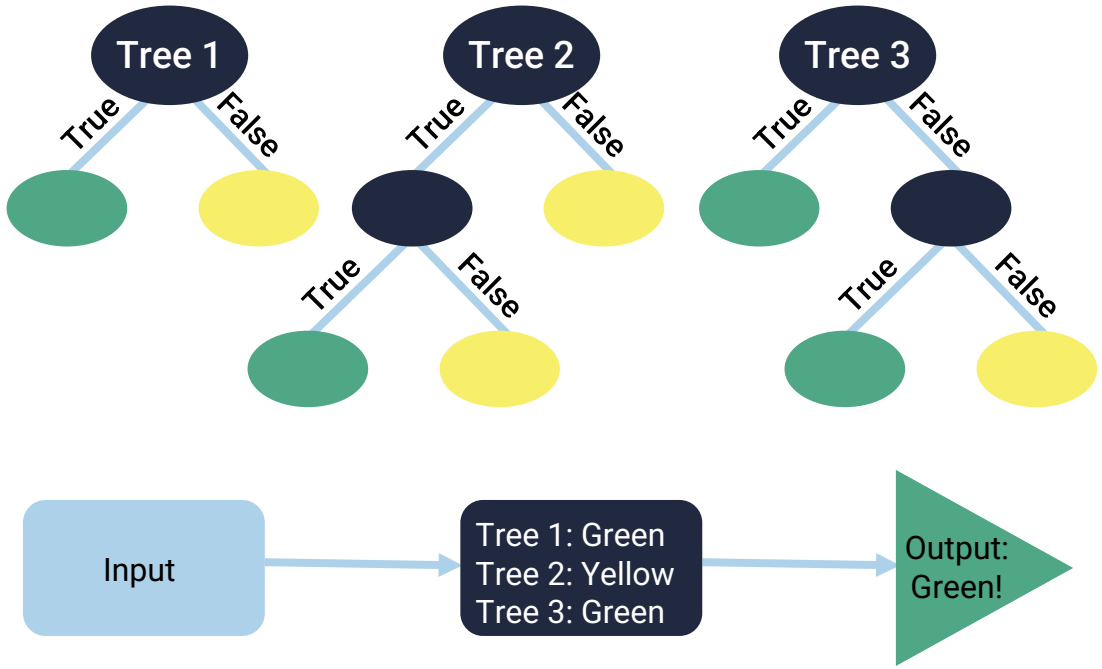


Random Forests

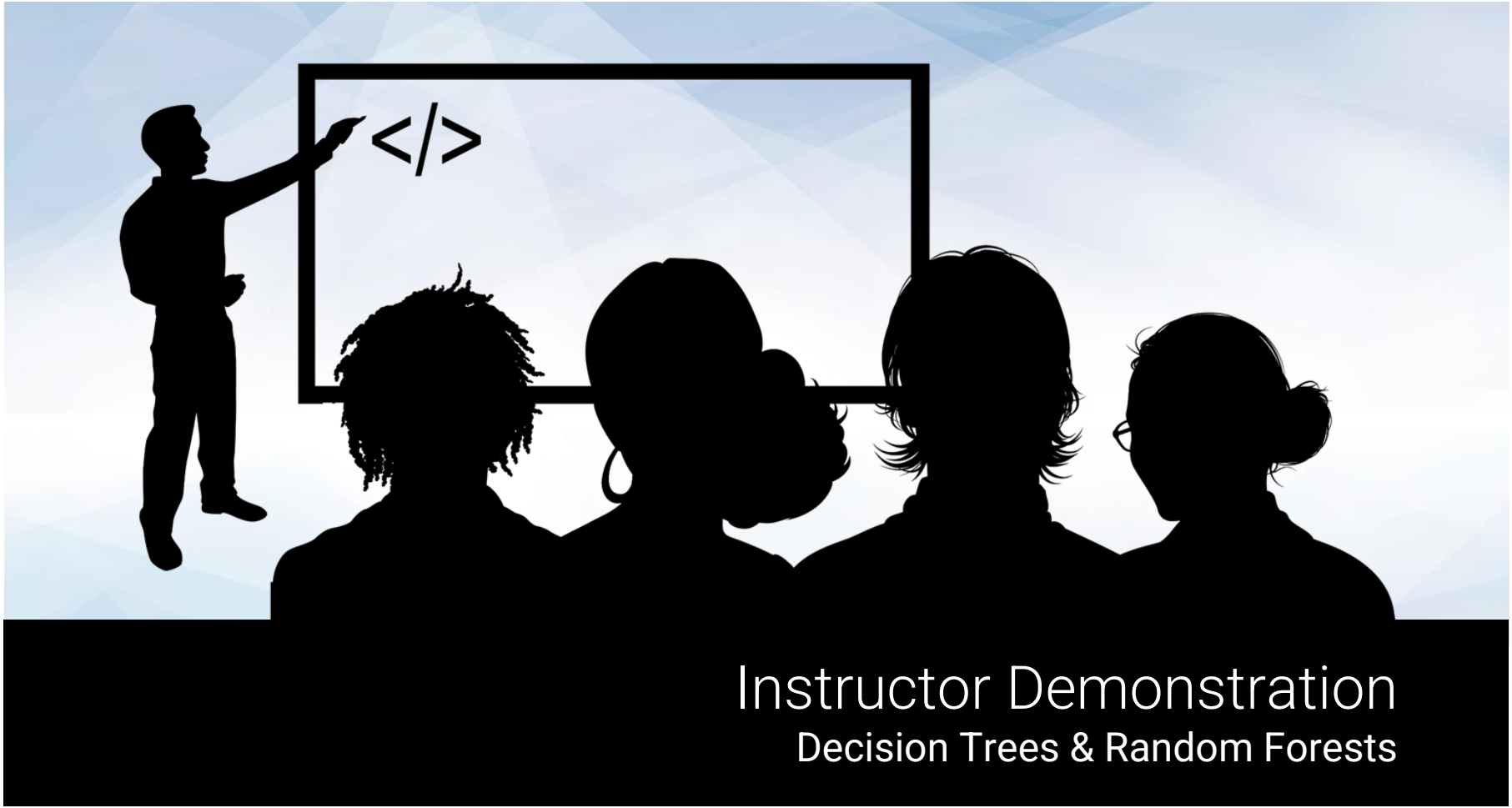
Instead of a single, complex tree like in the previous slide, a random forest algorithm will sample the data and build several smaller, simpler decisions trees (i.e., a forest of trees).

Each tree is much simpler because it is built from a subset of the data.

Each tree is considered a “weak classifier” but when you combine them, they form a “strong classifier.”



Questions?



Instructor Demonstration

Decision Trees & Random Forests

Activity: Trees

In this activity, you will compare the performance of a decision tree to a random forest classifier using the Pima Diabetes DataSet.

Suggested Time:
15 Minutes

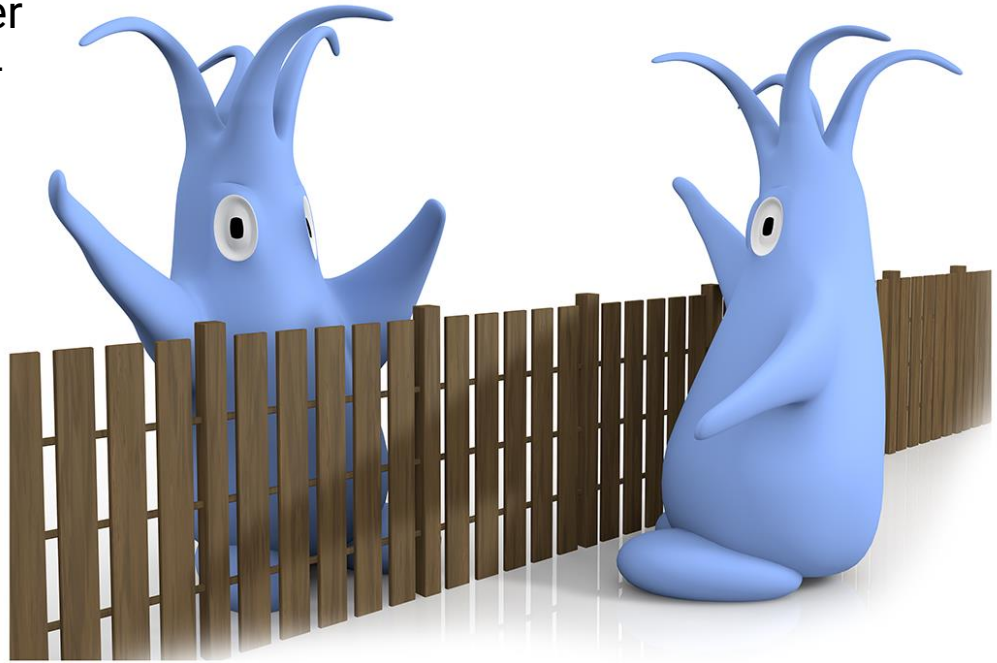
K Nearest Neighbors

K Nearest Neighbors Algorithm

K Nearest Neighbors (KNN) is a simple and robust algorithm for classification (and sometimes regression).

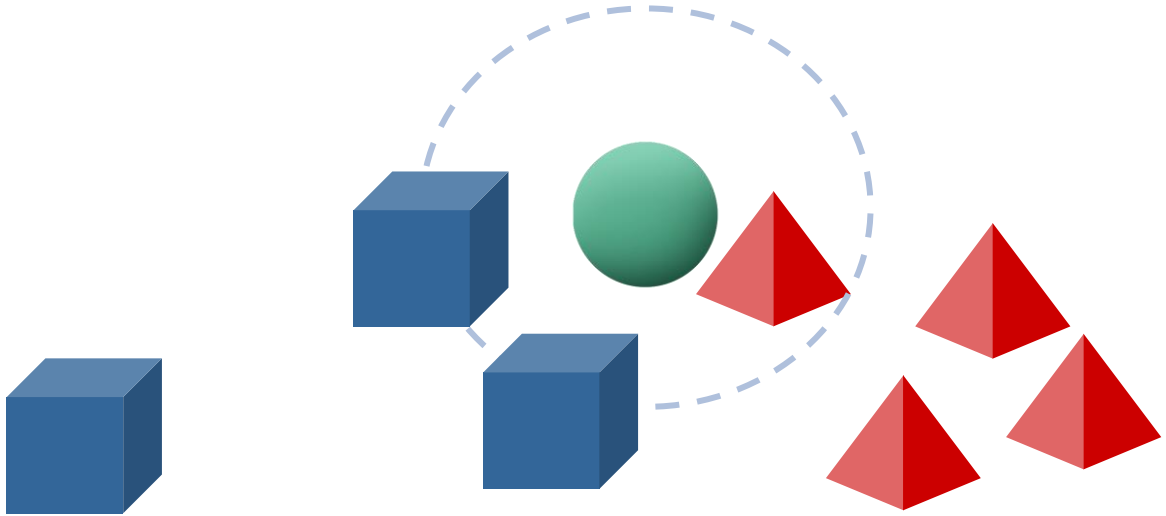
It has many benefits such as outlier insensitivity, ability to classify non-linear data, and high accuracy.

It does require a lot of memory.



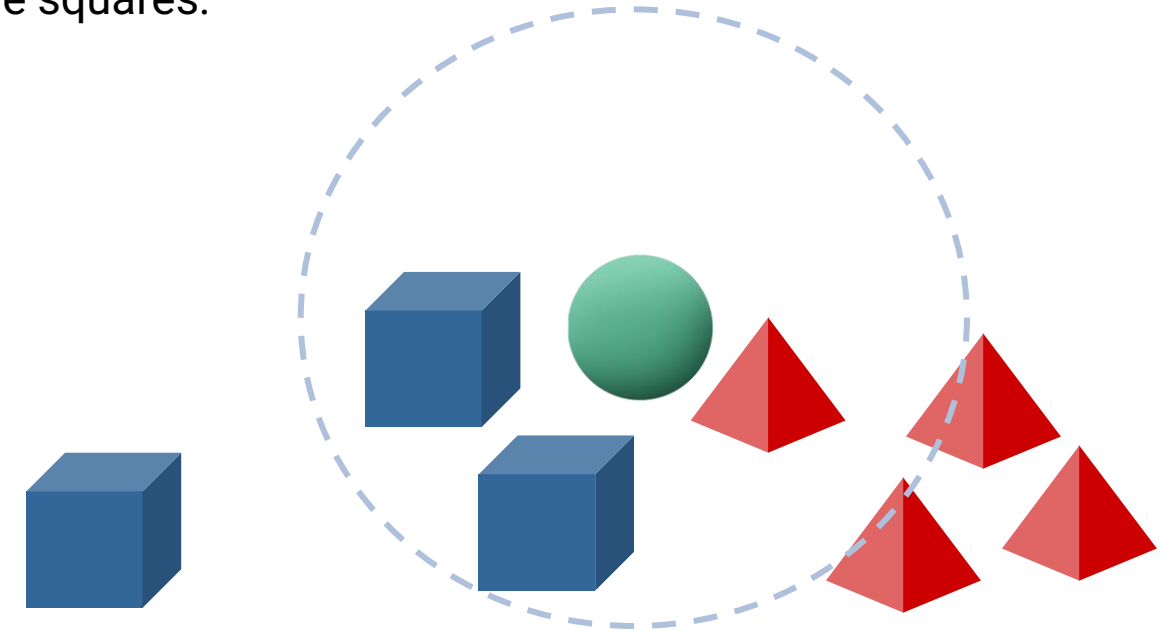
K = 1

When k = 1, this is simply the nearest neighbor. You find the point nearest to your new data point (the green circle) and that is the class that it will belong to. In this case, the closest neighbor is a red triangle, so the data point will belong to red.



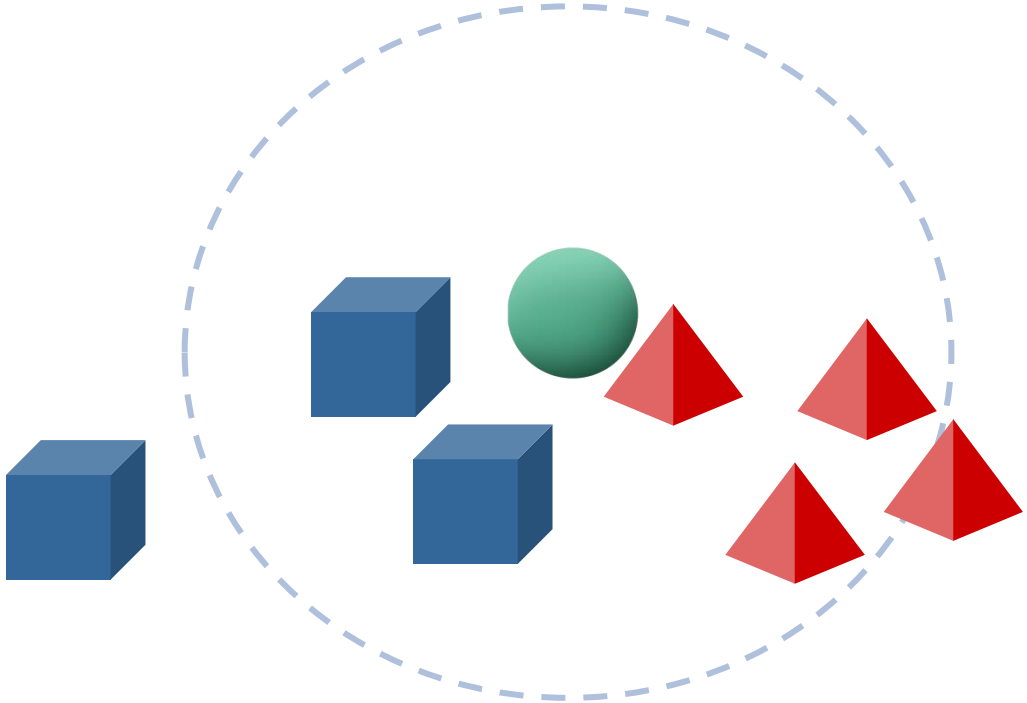
K = 3

When k = 3, we find the three closest neighbors. In this case, there are two blue squares and one red triangle, so the new data point will be grouped with the blue squares.



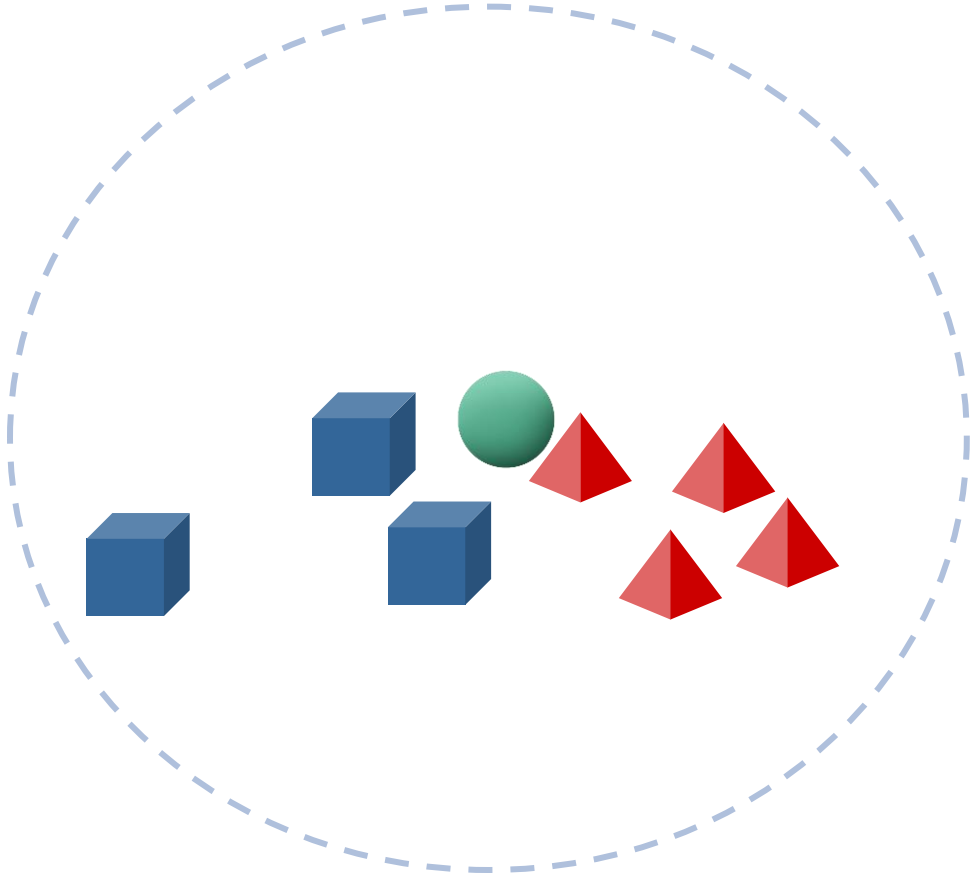
K = 5

When k = 5, there are three red triangles and two blue squares, so the new data point will belong to the red triangles.



K = 7

Finally, when k = 7, the majority are red triangles, so the new data point belongs to red.

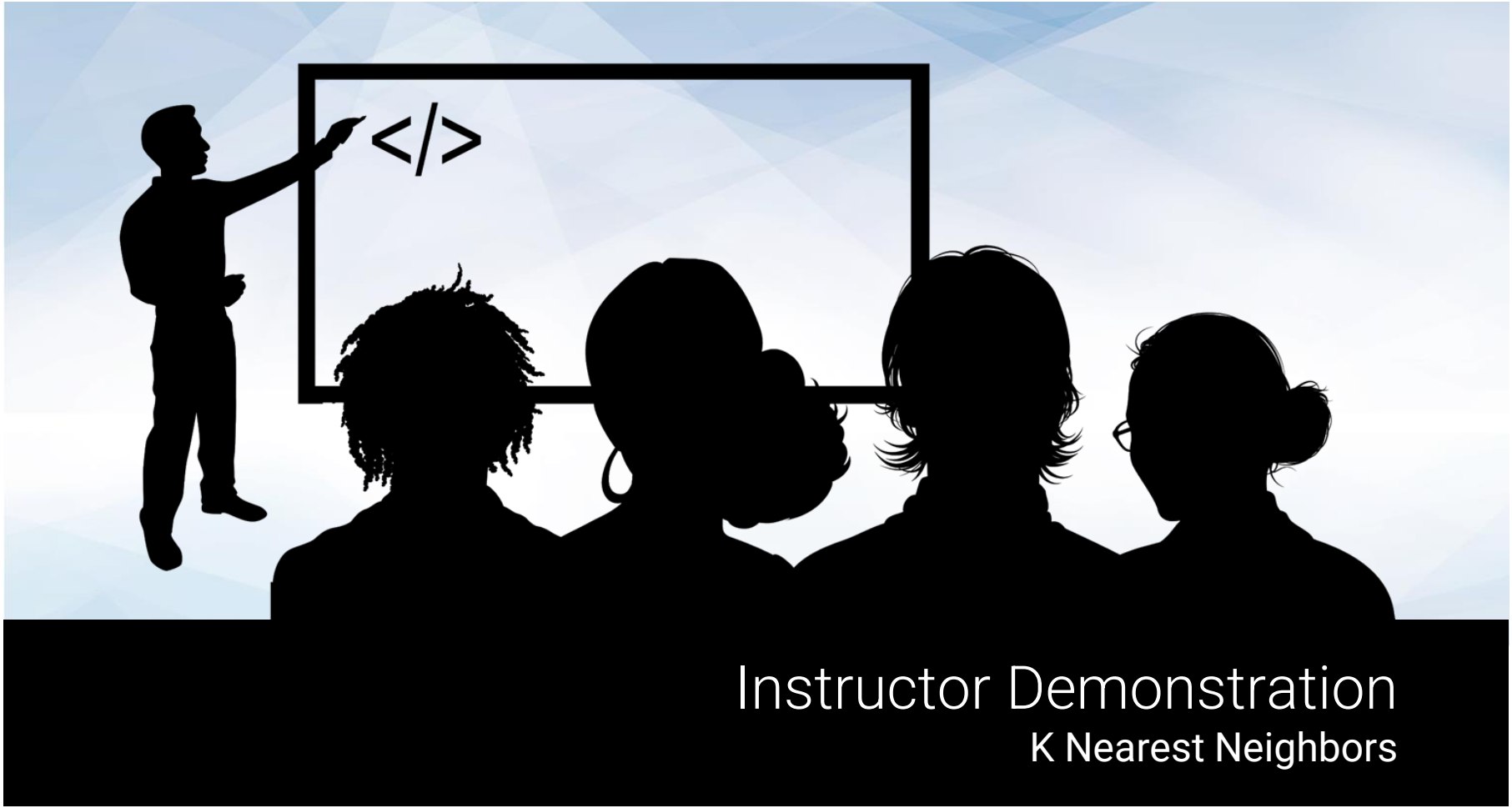


Choosing K

Because k can vary your results, the easiest technique for choosing a k value is to loop through a range of k and calculate the score. Choose the lowest value of k where the score starts to stabilize. **Note**: We only use odd numbers so there are no ties between classes.

```
for k in range(1, 20, 2):
        knn = KNeighborsClassifier(n_neighbors=k)
        knn.fit(data, labels)
        score = knn.score(data, labels)
        print(f"K: {k}, Score: {score}")
```

Questions?



Instructor Demonstration

K Nearest Neighbors

Activity: KNN

In this activity, you will determine the best k value in KNN to predict diabetes for the Pima Diabetes DataSet.

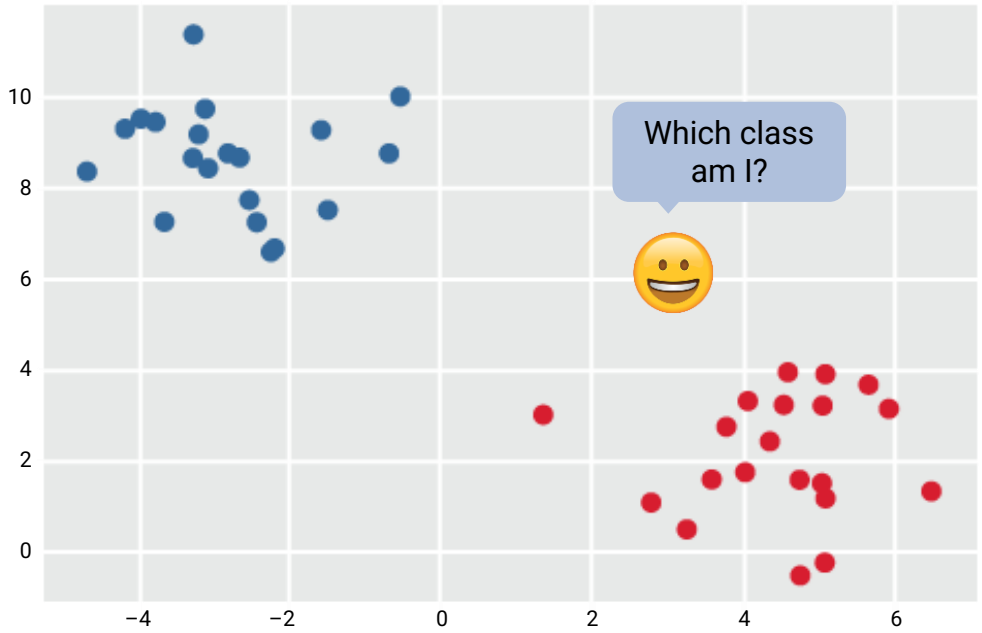
Suggested Time:
15 Minutes

BREAK

Support Vector Machine

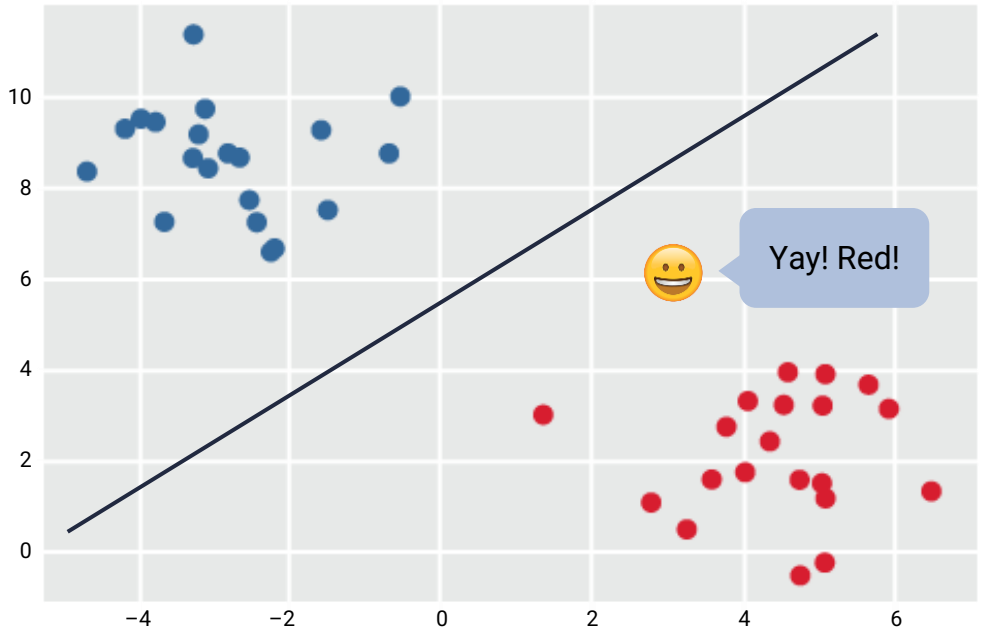
Linear Classifiers

Linear classifiers attempt to draw a line that separates the data, but which line best separates the groups?



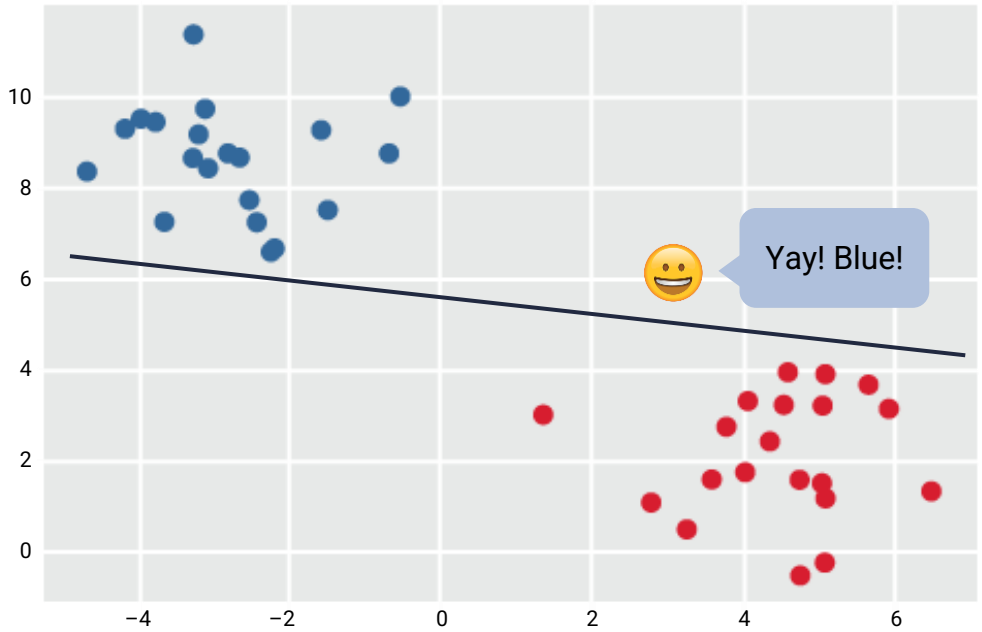
Linear Classifiers

Linear classifiers attempt to draw a line that separates the data, but which line best separates the groups?



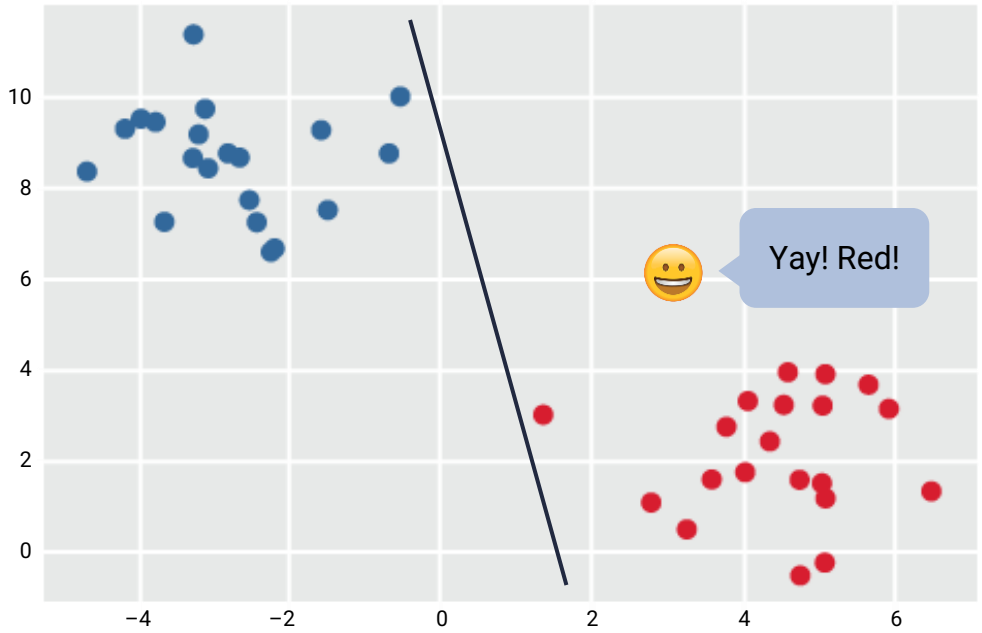
Linear Classifiers

Linear classifiers attempt to draw a line that separates the data, but which line best separates the groups?



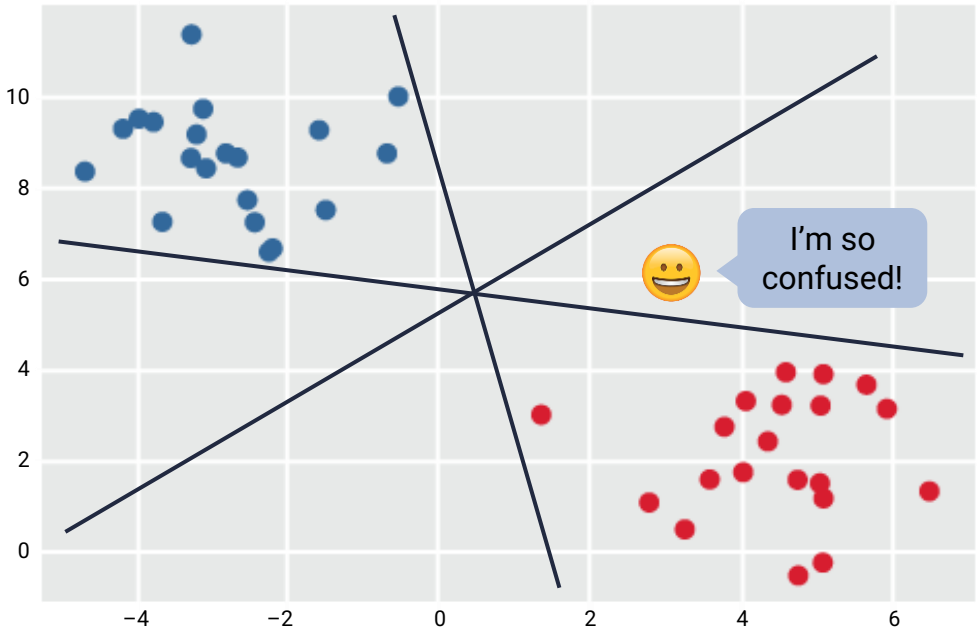
Linear Classifiers

Linear classifiers attempt to draw a line that separates the data, but which line best separates the groups?



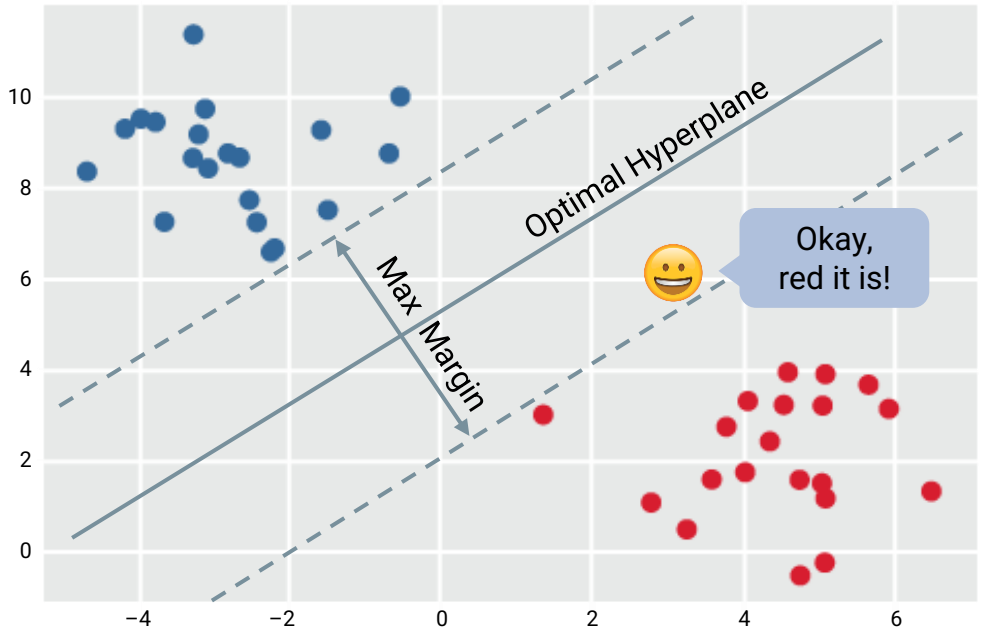
Linear Classifiers

Linear classifiers attempt to draw a line that separates the data, but which line best separates the groups?

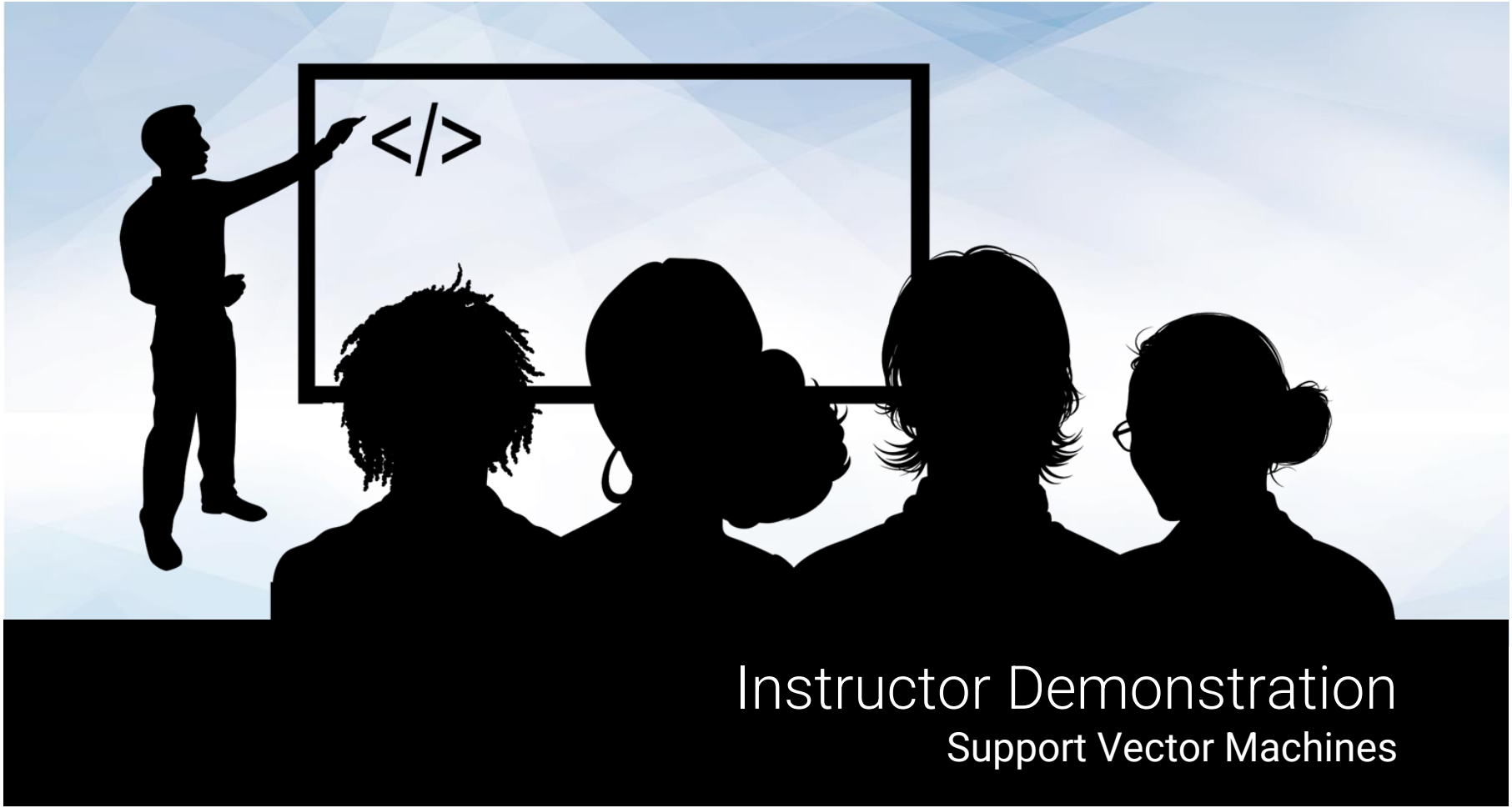


Support Vector Machines

The Support Vector Machines (SVM) algorithm finds the optimal hyperplane that separates the data points with the largest margin possible.



Questions?



Instructor Demonstration

Support Vector Machines

Precision and Recall

Data Boot Camp

Lesson 21.2

Precision & Recall

Confusion Matrix

A confusion matrix is a table used to describe the performance of a classifier by comparing the predicted and actual values. Consider the following matrix where the classes are “Cancer” or “No Cancer.”

	Predicted Class		
Actual Class		Cancer	No Cancer
	Cancer	True Positive	False Negative
	No Cancer	False Positive	True Negative

- True Positive (TP): The predicted class and the actual class are the same. Both predicted Cancer.
- True Negative (TN): The predicted class and the actual class are the same. Both predicted No Cancer.
- False Negative (FN): The actual class was Cancer, but the prediction was No Cancer.
- False Positive (FP): The actual class was No Cancer, but the prediction was Cancer.

Accuracy

Accuracy is the ratio of correctly predicted observations to the total number of observations.

Accuracy = (TP + TN) / all observations

	Predicted Class		
Actual Class		Cancer	No Cancer
	Cancer	True Positive	False Negative
	No Cancer	False Positive	True Negative

Accuracy

Accuracy is the ratio of correctly predicted observations to the total number of observations.

$$\text{Accuracy} = 85/100 = 0.85$$

	Predicted Class		
Actual Class		Cancer	No Cancer
	Cancer	25	10
	No Cancer	5	60

Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations (i.e., of all the samples we classified as Cancer, how many are actually Cancer?).

$$\text{Precision} = TP / (TP + FP)$$

	Predicted Class		
Actual Class		Cancer	No Cancer
	Cancer	True Positive	False Negative
	No Cancer	False Positive	True Negative

Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations (i.e., of all the samples we classified as Cancer, how many are actually Cancer?).

Precision = 25/30 = .8333

	Predicted Class		
Actual Class		Cancer	No Cancer
	Cancer	25	10
	No Cancer	5	60

Recall

Recall is the ratio of correctly predicted positive observations to the total predicted positive observations (i.e., of all the actual Cancer samples, how many did we classify as Cancer?).

$$\text{Recall} = TP / (TP + FN)$$

	Predicted Class		
Actual Class		Cancer	No Cancer
	Cancer	True Positive	False Negative
	No Cancer	False Positive	True Negative

Recall

Recall is the ratio of correctly predicted positive observations to the total predicted positive observations (i.e., of all the actual Cancer samples, how many did we classify as Cancer?).

Recall = 25/35 = .714

	Predicted Class		
Actual Class		Cancer	No Cancer
	Cancer	25	10
	No Cancer	5	60

F1 Score

The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$F1 = 2 * ((precision * recall) / (precision + recall))$$

	Predicted Class		
Actual Class		Cancer	No Cancer
	Cancer	True Positive	False Negative
	No Cancer	False Positive	True Negative

F1 Score

The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$F1 = 2 * ((.8333 * .714) / (.8333 + .714)) = 0.77$$

	Predicted Class		
Actual Class		Cancer	No Cancer
	Cancer	25	10
	No Cancer	5	60

Questions?

Activity: SVM

In this activity, apply a support vector machine classifier predict diabetes for the Pima Diabetes DataSet.

Suggested Time:
15 Minutes