# Classical Classification

FinTech

Lesson 11.1

# Software to Install for Module 11

Please install the following software during this class

Instructions can be found here

# Class Objectives

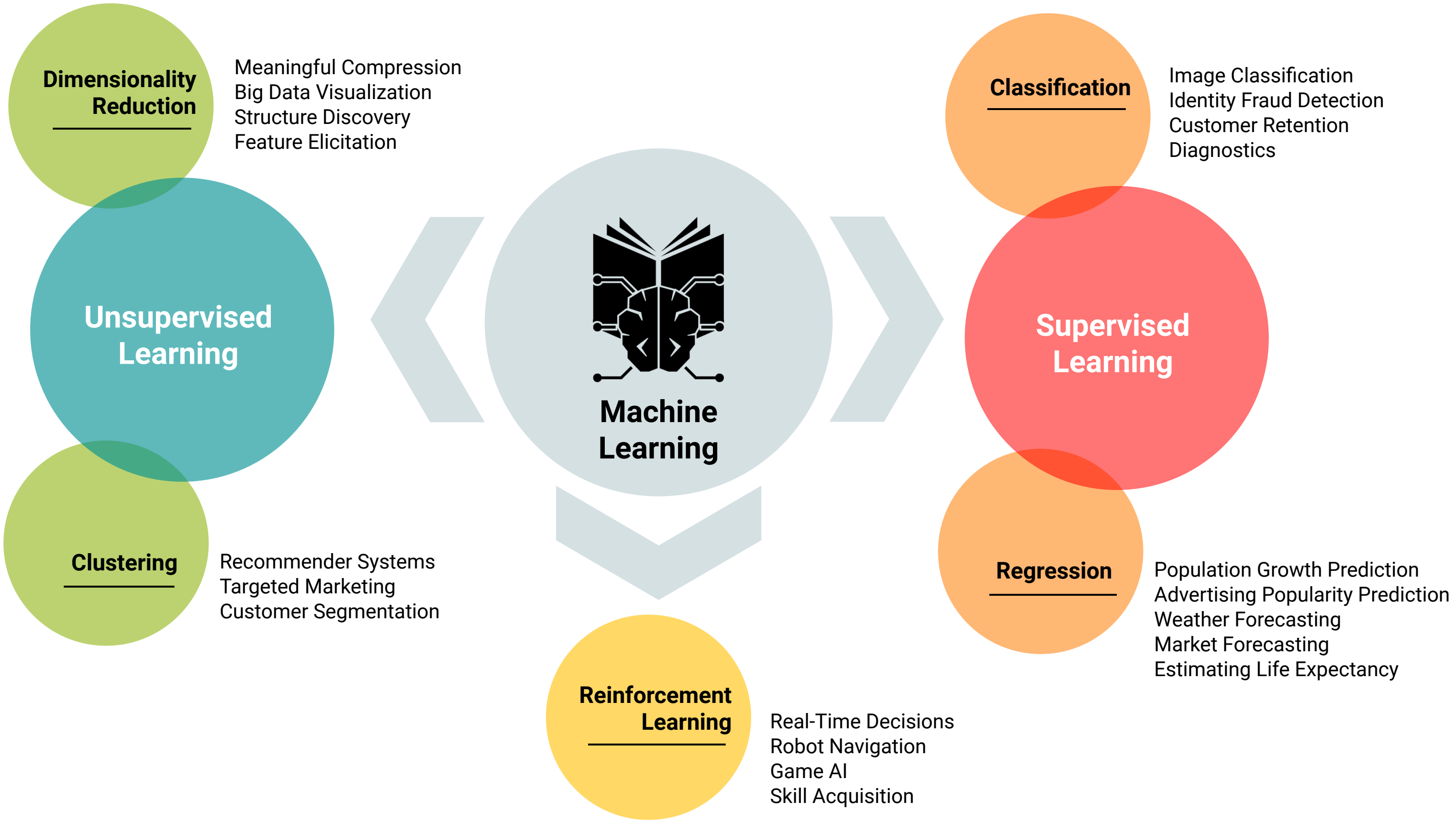
In today's class, we'll learn about classification algorithms

- Logistic/linear regression
- Support vector machines

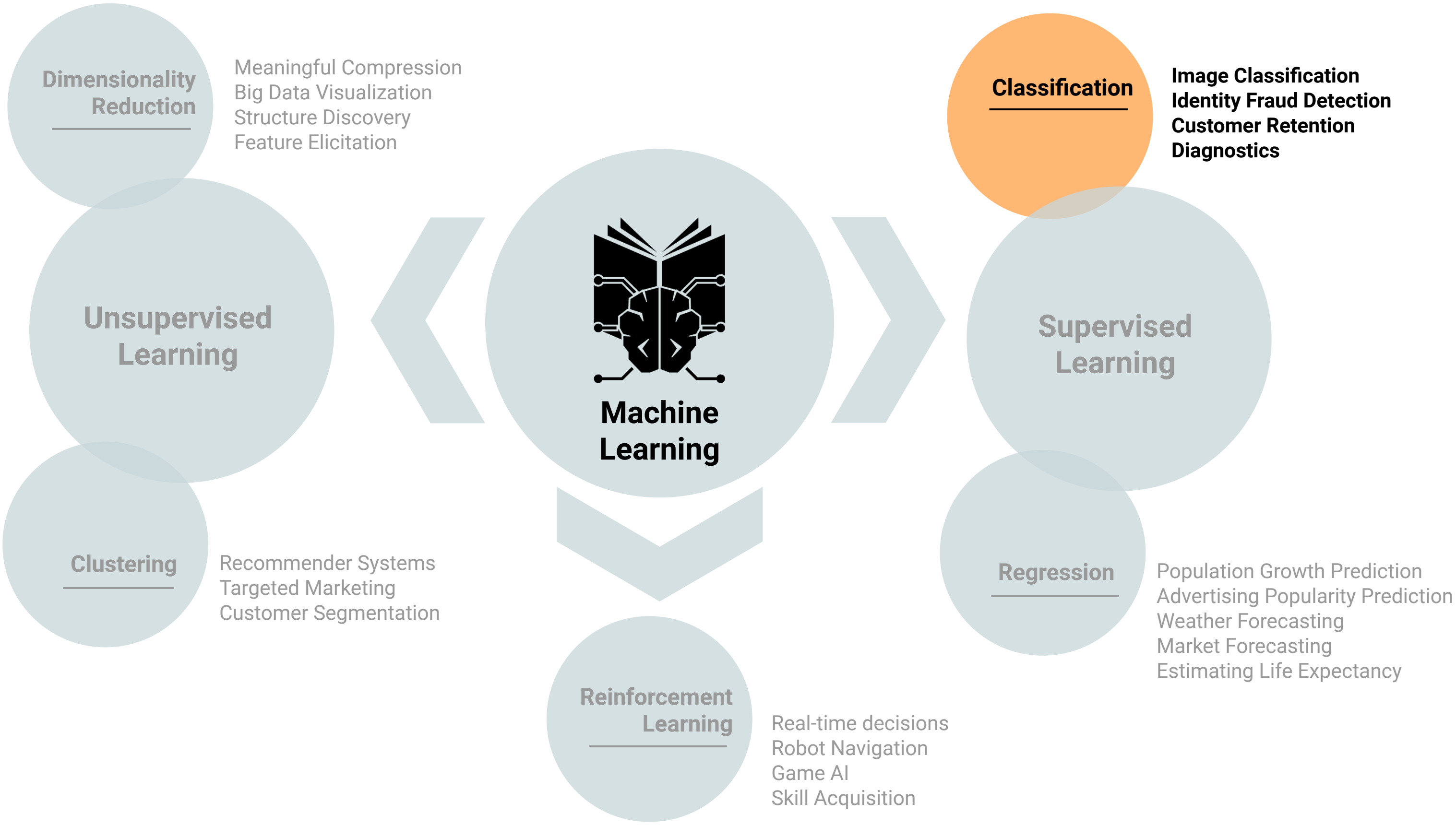
Use Cases:

- Fraud detection
- Price and ROI forecasting
- Medical disease/condition diagnosis

# This is the second week of machine learning!



# Intro to Classification



# Intro to Classification

Classification is the prediction of discrete outcomes. Outcomes are identified as labels, which serve to categorize bi-class and multi-class features.

| | vs. | |
|---|---|---|
| Spam | | Not Spam |
| Risk | | No Risk |
| 0 | | 1 |

**Classification** is the action or process of categorizing something according to shared qualities or characteristics.

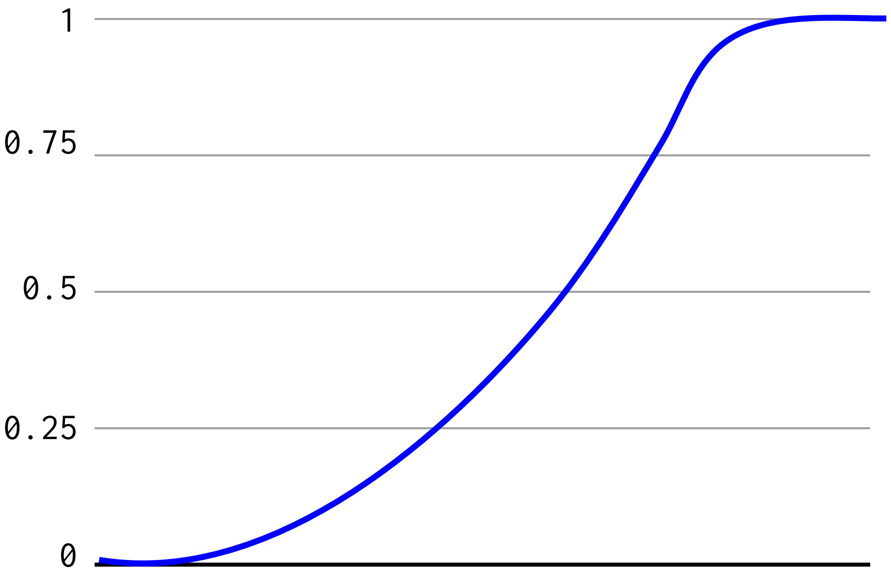
# Classification

Classification is used to forecast and predict financial outcomes, automate underwriting and insurance premiums, and detect and categorize health issues.

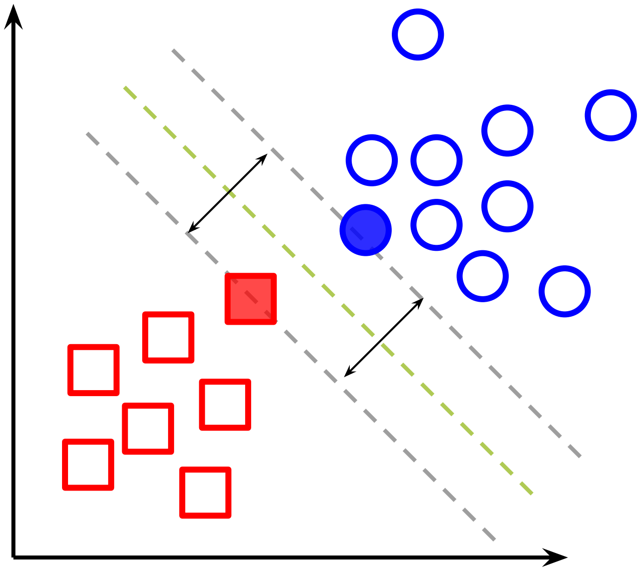
# Classification

There are multiple approaches to classification, which include:

## Logistic Regression




## Support Vector Machines



# Classification

Classification models have drastically improved financial efforts to properly categorize applicants, predict market decline, and categorize fraudulent transactions or suspicious activity.

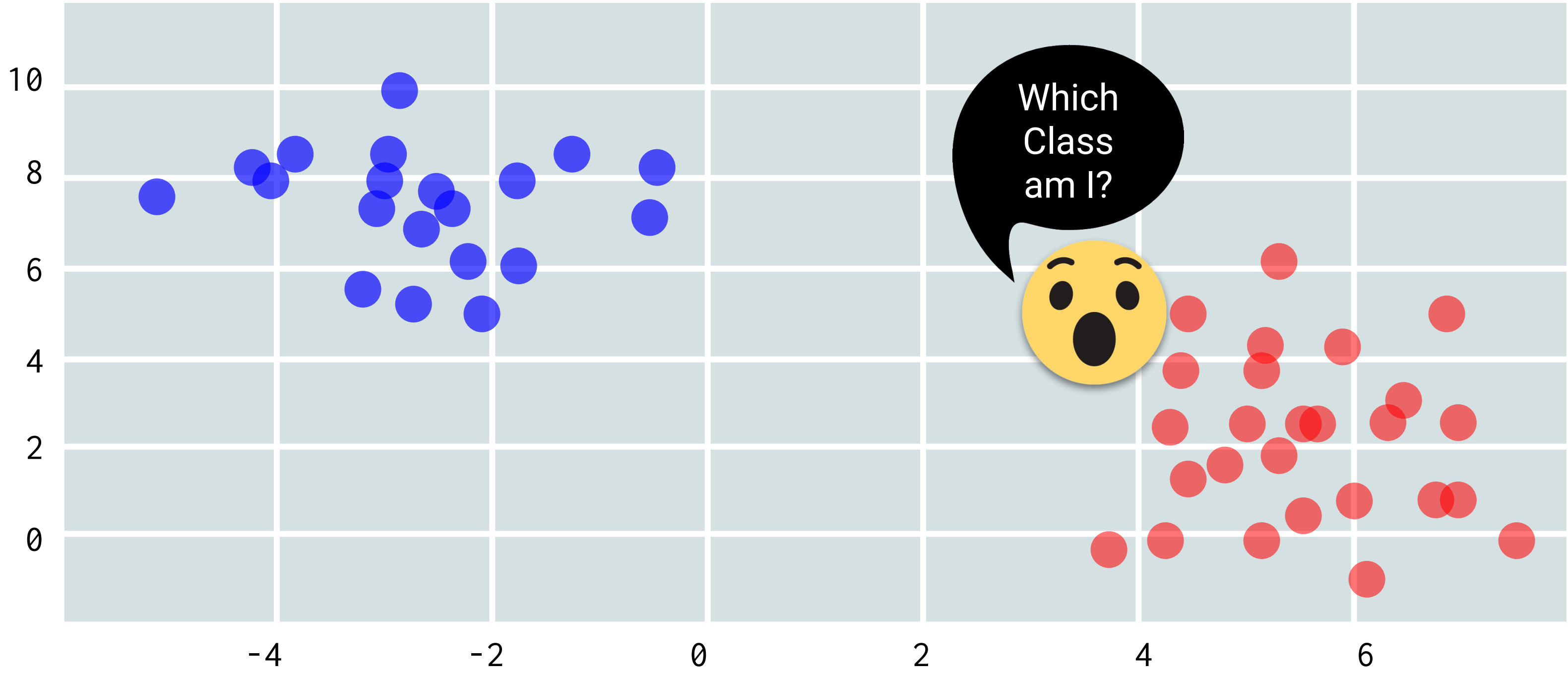
# Classification

FICO credit scoring uses a classification model for its cognitive fraud analytics platform. Classification engines have allowed the financial industry to become more effective and efficient at mitigating risk.

# Making Predictions with Logistic Regression

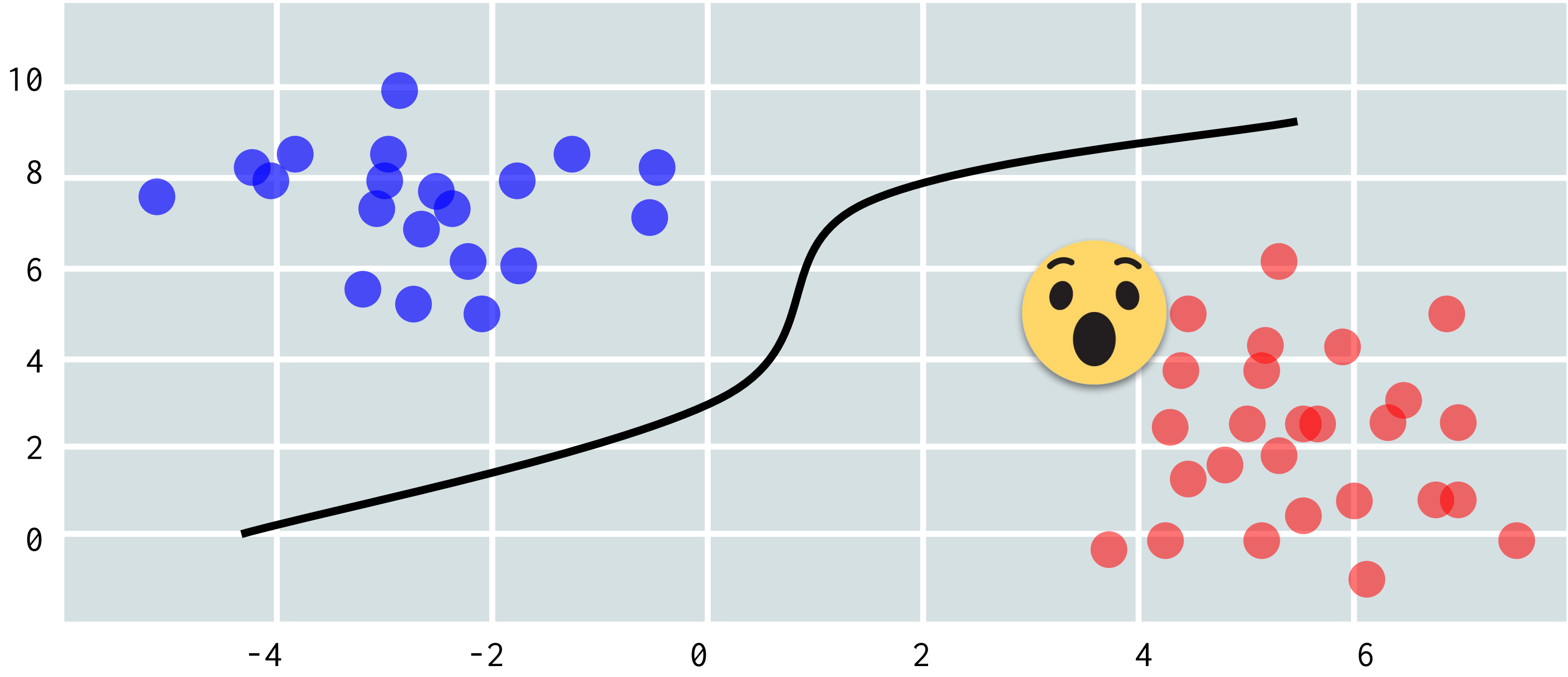
# Making Predictions with Logistic Regression

Logistic regression is a common approach used to classify data points and make predictions.



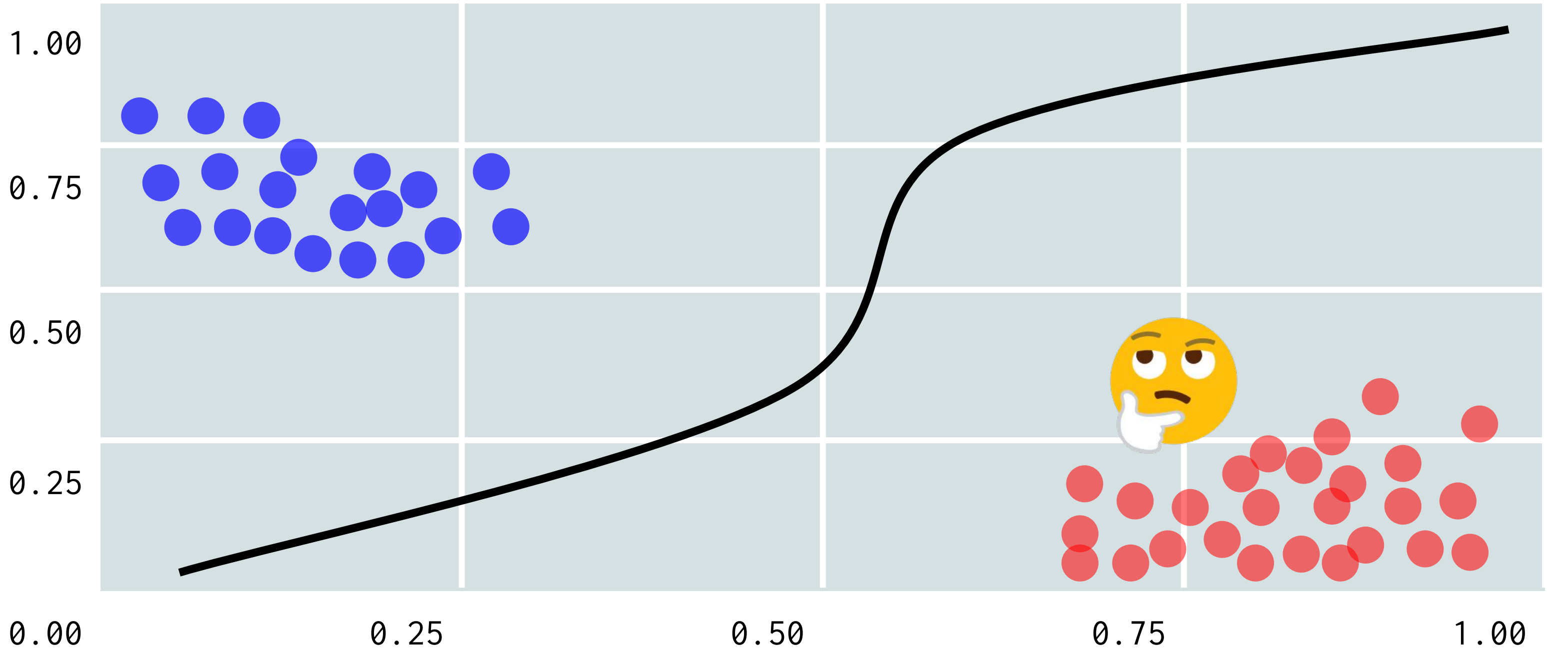
# Making Predictions with Logistic Regression

Predictions are made by creating linear paths between data points.



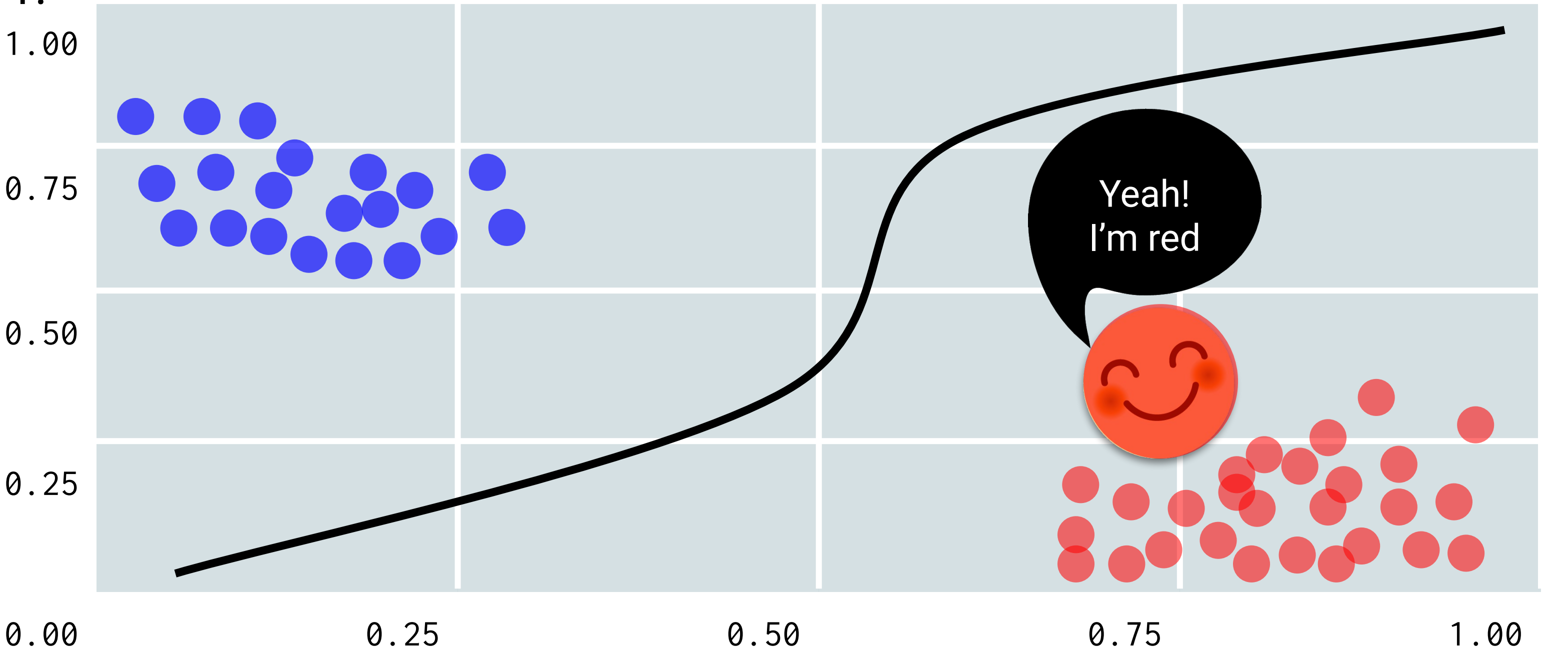
# Making Predictions with Logistic Regression

Data points along the trajectory are normalized between 0 and 1.



# Making Predictions with Logistic Regression

If a value is above a certain threshold, the data point is considered either of class 0 or 1.



# Logistic Regression Equation

$$ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x$$

$$=> P = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

# Logistic Regression Model

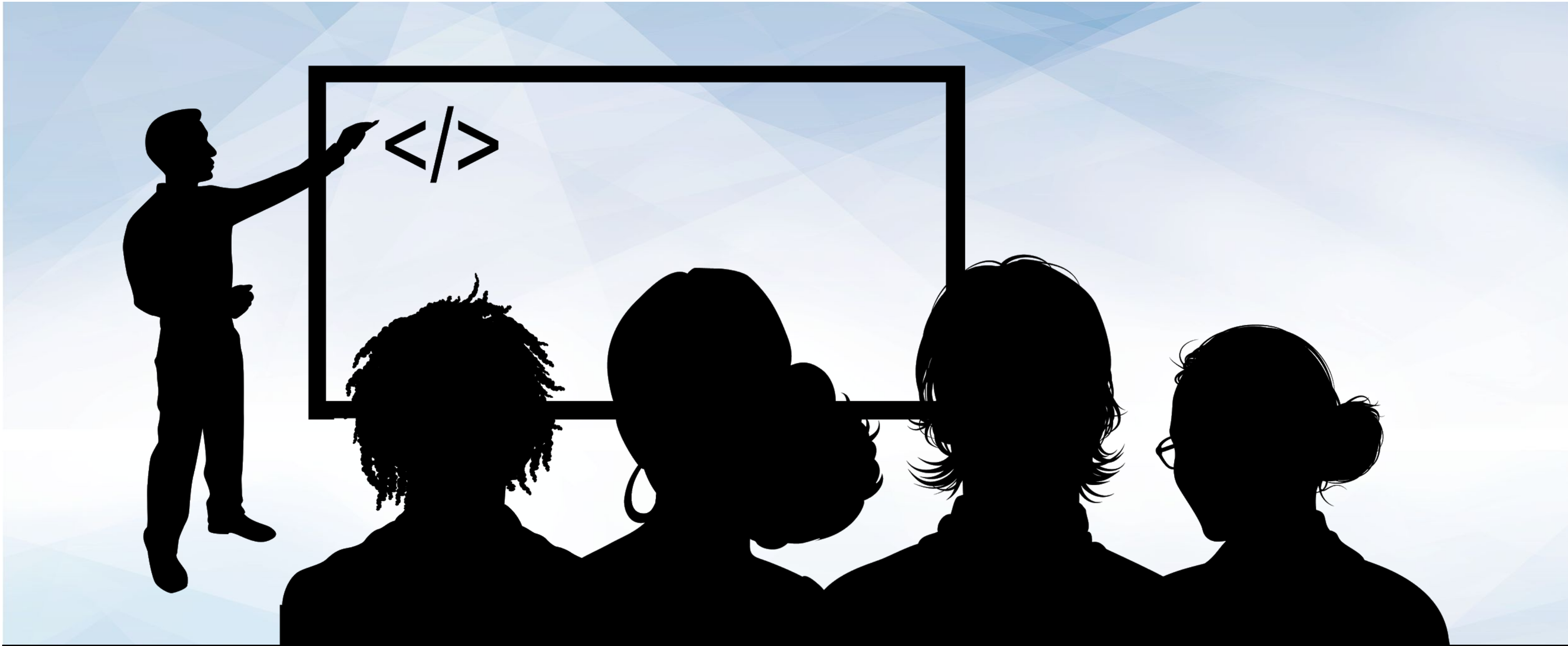
Running a logistic regression model involves four steps, which can be applied when running any machine-learning model:

01 Preprocess

02 Train

03 Validate

04 Predict




# Instructor Demonstration

## Logistic Regression using Scikit-Learn

# **Activity:** Predicting Diabetes

In this activity, you will use the sklearn library to execute logistic regression models in order to predict whether or not an individual has diabetes.

**Suggested Time:**
15 minutes

# **Time's Up!** Let's Review.

# **Review:** Predicting Diabetes

- How well did your model perform?
- How do you know? Did you count the results?
- If you had to diagnose a patient, how confident would you be in your model's prediction?

# Evaluating Logistic Regression Predictions

# How sure are you that models can actually predict diabetes?

75%

sure, as described by the scored accuracy.

# Would you feel comfortable giving a diagnosis of diabetes, based on the predictions of the model?

**No.** The prediction is not 100% accurate. There is room for error, as well as false positives.

# What is better: the false positive or false negative?

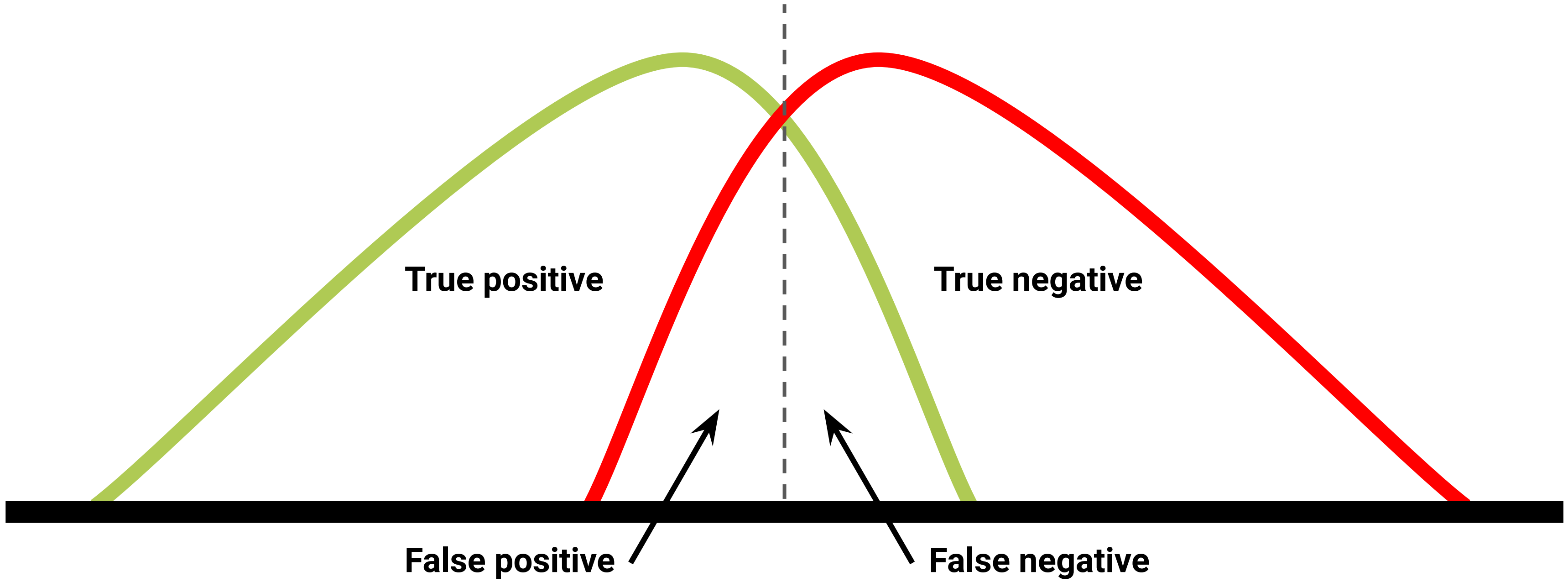
**False Positive.** Additional tests can be run to refine the prediction and filter out individuals who don’t have diabetes. That way, people who have it can get the treatment they need.

In addition to accuracy, a model must be measured for precision and recall, both of which can be used to eliminate false positives and false negatives.

# Accuracy, Precision, Recall

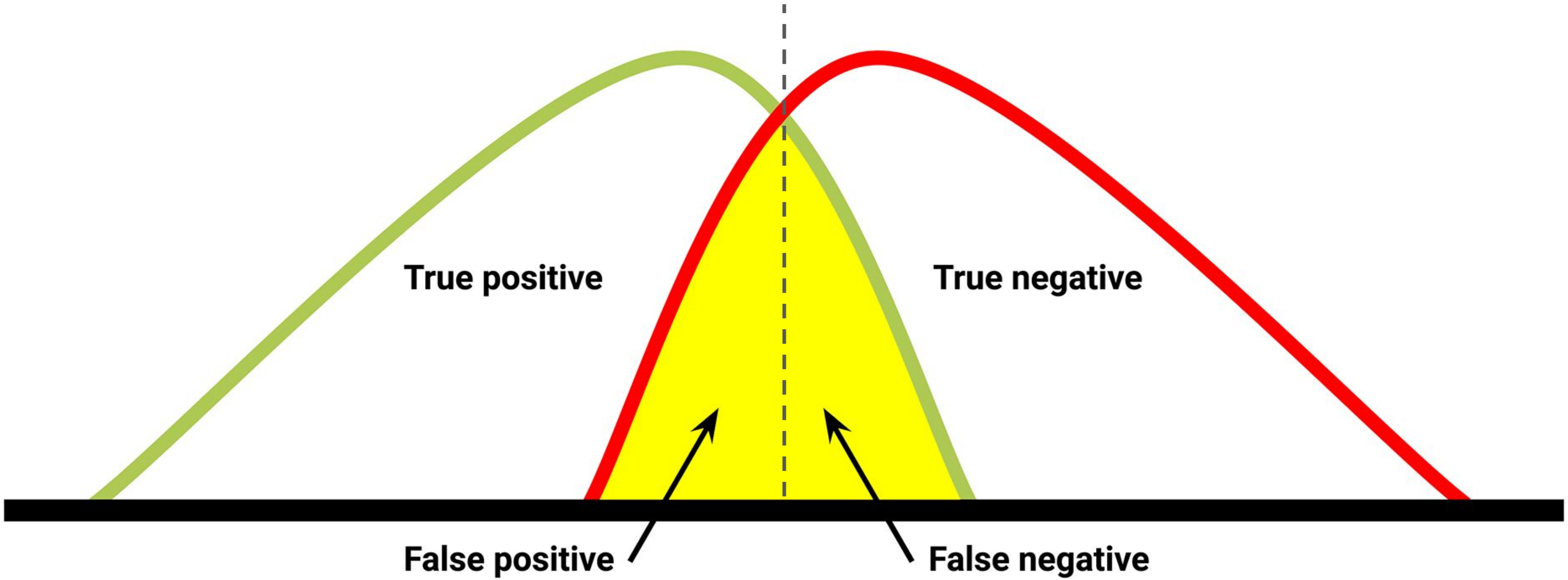
# Accuracy, Precision, Recall

Accuracy, precision, and recall are especially important for classification models that involve a binary decision problem. Binary decision problems have two possible correct answers: True Positive and True Negative.



# Accuracy, Precision, Recall

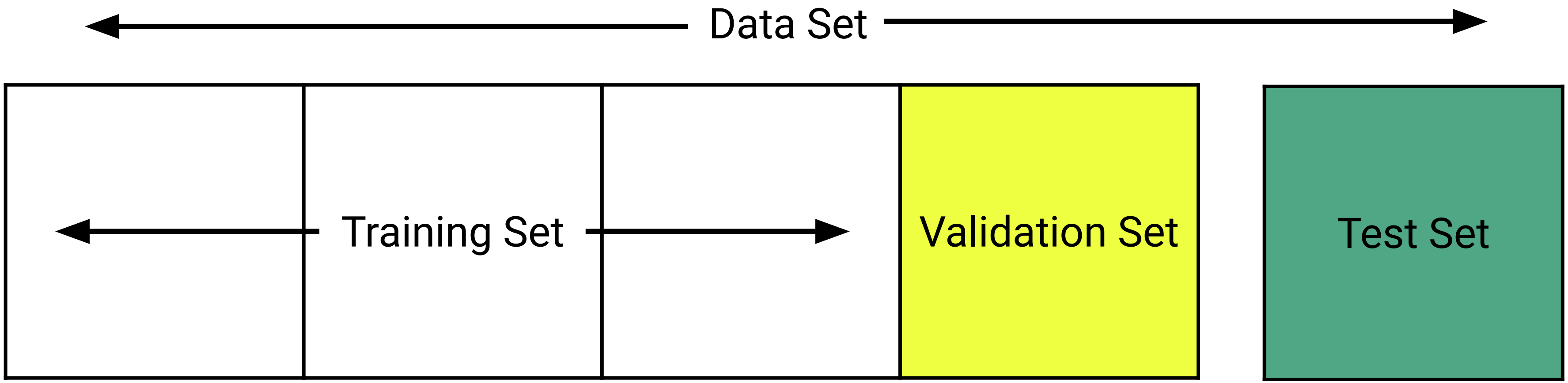
Inaccurate and imprecise models result in models returning false positives and false negatives.



**Accuracy** is how often the model is correct—the ratio of correctly predicted observations to the total number of observations.

# Accuracy

Scoring reveals how accurate the model is, but not how precise it is.



# Accuracy

Accuracy can be very susceptible to imbalanced classes. In the case of the homework assignment, the number of good loans greatly outweighs the number of at-risk loans. In this case, it can be really easy for the model to only care about the good loans because that has the biggest impact on accuracy. However, we also care about the at-risk loans, so we need a metric that can help us evaluate each class prediction.

**Calculation:**

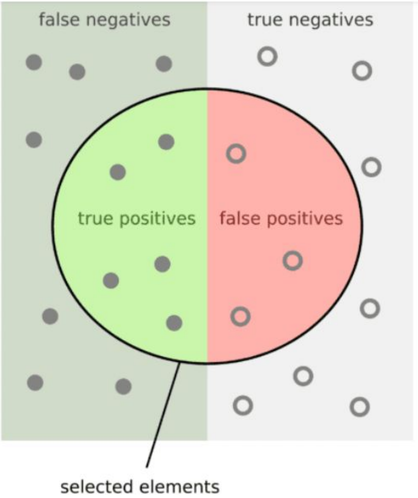
```
(TP + TN) / (TP + TN + FP + FN)
```

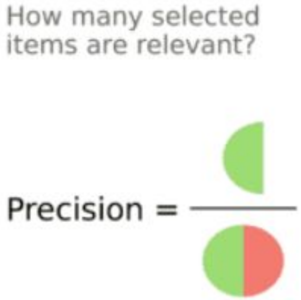
**Precision** is the ratio of correctly predicted positive observations to the total predicted positive observations.

# Precision

Another example: Of all of the individuals that were classified by the model as being a credit risk, how many actually were a credit risk?

**The question at hand:** Did we classify comprehensively and correctly?



# Precision

High precision relates to a low false-positive rate.

**Calculation:**

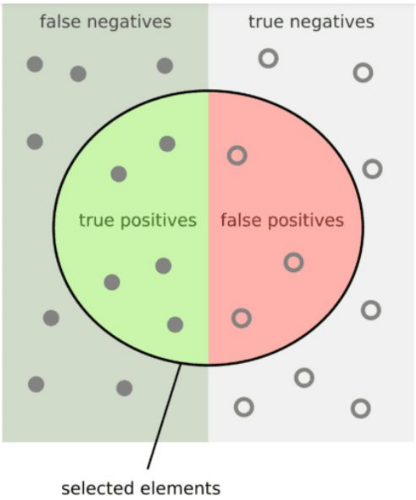
$$TP \ / \ (TP + FP)$$

**Recall** is the ratio of correctly predicted positive observations to all predicted observations for that class.

# Recall

Of all of the actual diabetes/credit risk samples, how many were correctly classified as having diabetes/being a credit risk?

**The question at hand: Did we classify all samples correctly, leaving little room for false negatives?**




# Recall

High recall relates to a more comprehensive output and a low false-negative rate.

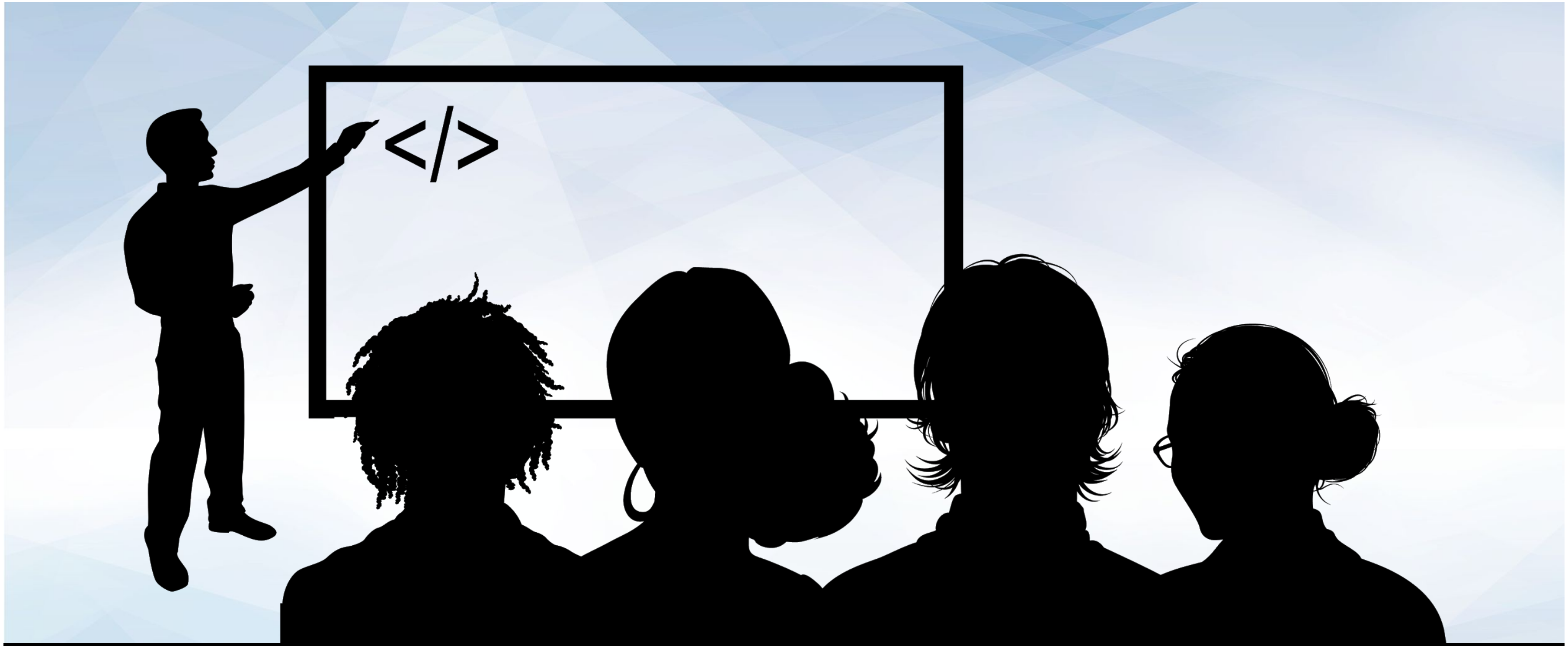
**Calculation:**

$$TP \ / \ (TP + FN)$$

# When to use Recall Vs. Precision?

- Diagnosing cancer patients
- Determining if a loan applicant will default
- Determining whether someone likes rock music

# Confusion Matrix & Classification Report




# Instructor Demonstration

## Confusion Matrix & Classification Report

A **confusion matrix** is used to measure and gauge the success of a model.

# Confusion Matrix

Confusion matrices reveal the number of true negatives and true positives (actuals) for each categorical class, which are then compared to the number of predicted values for each class.

| n=165 | **Predicted: No** | **Predicted: Yes** |
|---|---|---|
| **Actual = No** | 50 | 10 |
| **Actual = Yes** | 5 | 100 |

# Confusion Matrix

These values are then individually summed by column and row. The aggregate sums are then compared to gauge accuracy and precision. If the aggregates match, the model can be considered accurate and precise.

| n =165 | **Predicted: No** | **Predicted: Yes** | |
|---|---|---|---|
| **Actual = No** | 50 | 10 | **= 60** |
| **Actual = Yes** | 5 | 100 | **= 105** |
| | **= 55** | **= 110** | |

# Classification Report

A classification report identifies the **precision**, **recall**, and **accuracy** of a model for each given class.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| No Diabetes | 0.77 | 0.90 | 0.83 | 125 |
| Diabetes | 0.72 | 0.49 | 0.58 | 67 |
| accuracy | | | 0.76 | 192 |
| macro avg | 0.74 | 0.69 | 0.71 | 192 |
| weighted avg | 0.75 | 0.76 | 0.74 | 192 |

# **Activity:** Diagnosing the Model

In this activity, you will return to the model you created to predict diabetes, and use a confusion matrix and classification report to evaluate and diagnose the model.

**Suggested Time:**
10 minutes

# **Time's Up!** Let's Review.

# **Activity:** Build Loan Approver

In this activity, you will apply the machine learning concepts and technical skills learned so far to create a model for approving loans.

**Suggested Time:**
15 minutes

# **Time's Up!** Let's Review.

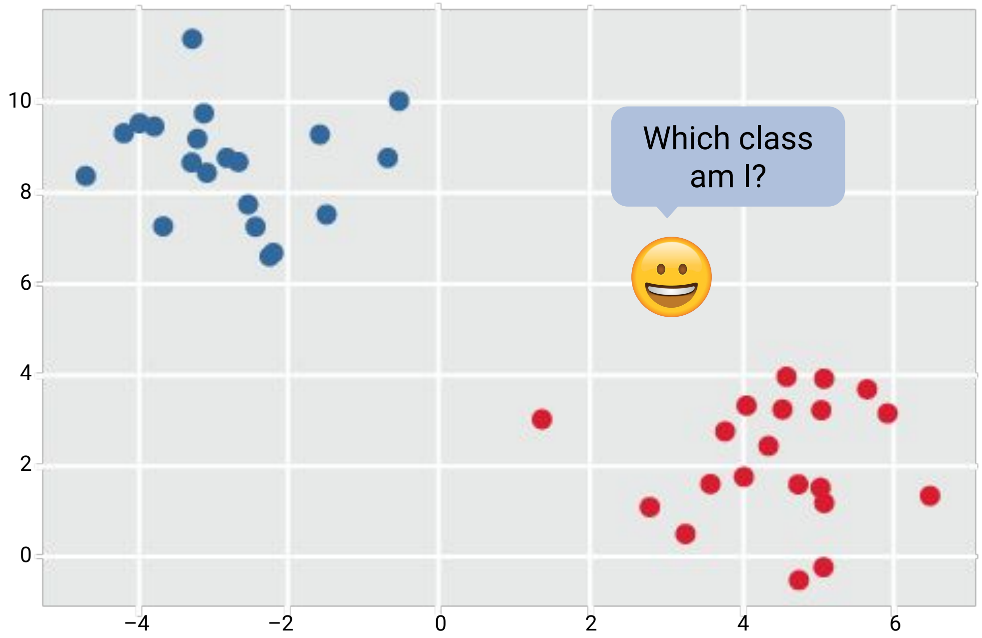

Break

# Support Vector Machines

**Support Vector Machines** (SVM) is a supervised learning model that can be used for classification and regression analysis. SVM separates classes of data points into multidimensional space.

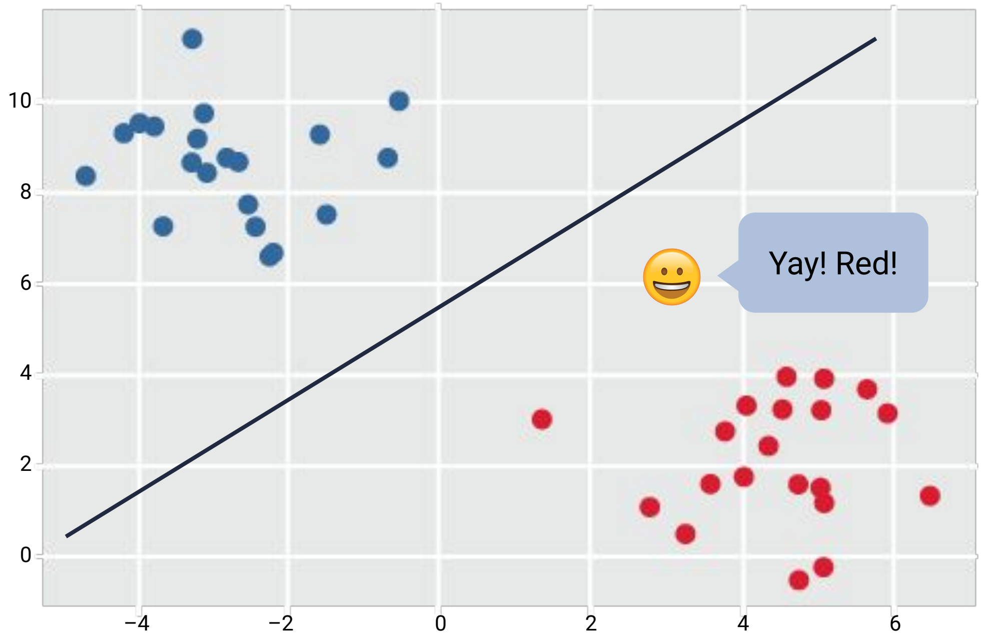
# Linear Classifiers

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



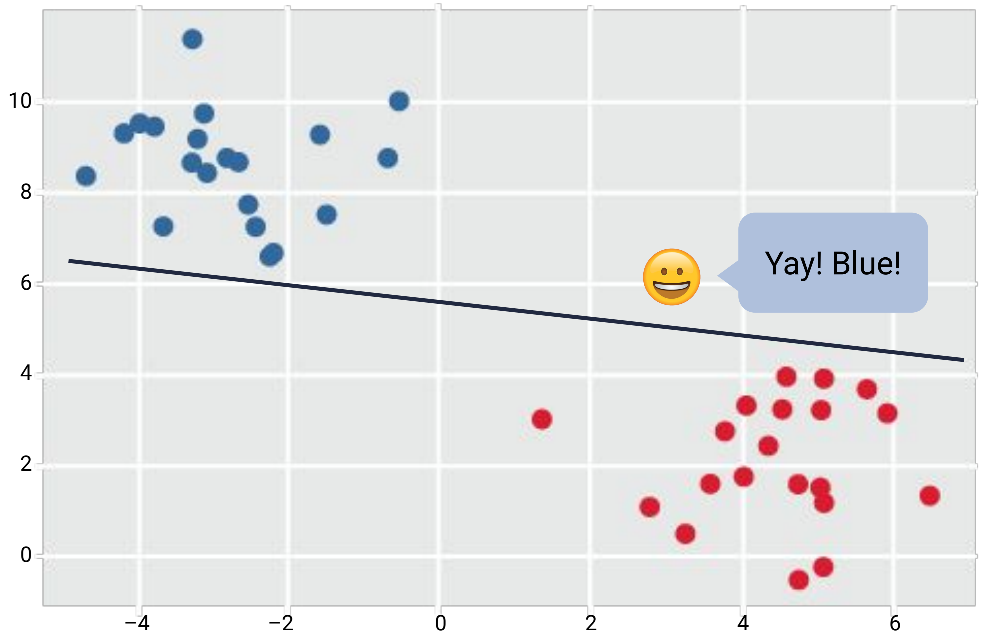
# Linear Classifiers

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



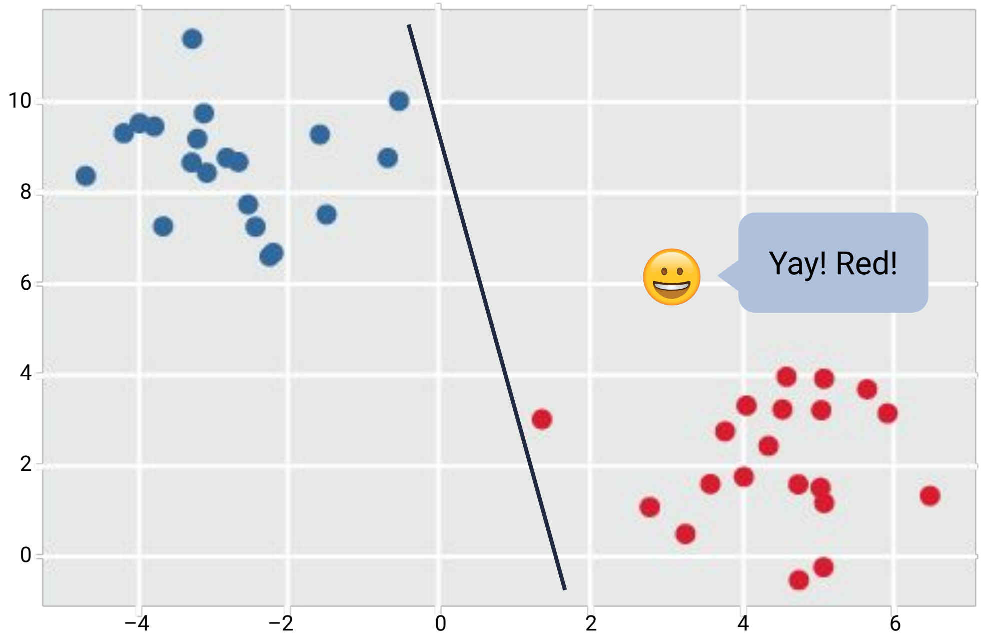
# Linear Classifiers

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



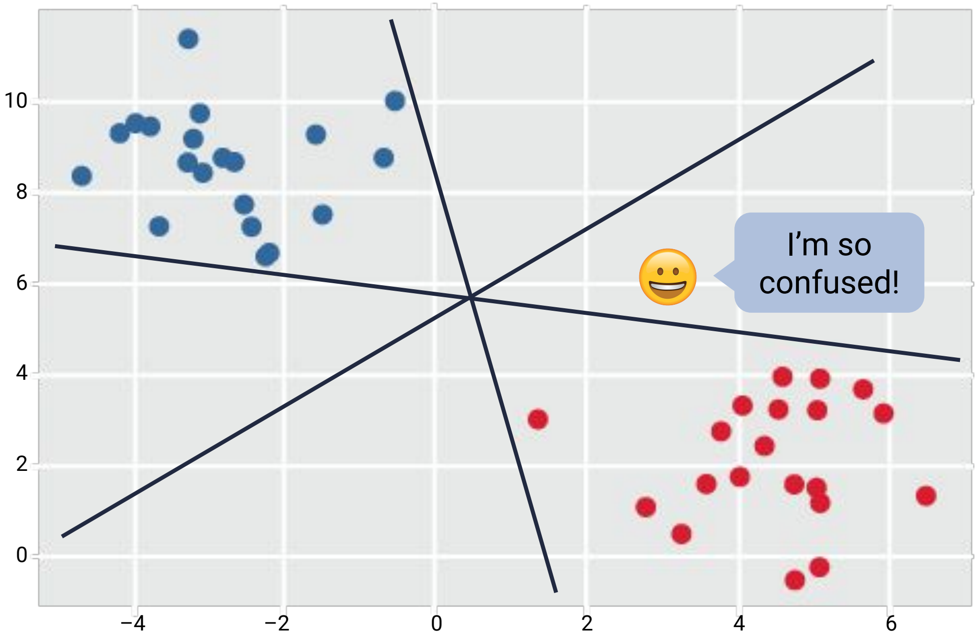
# Linear Classifiers

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



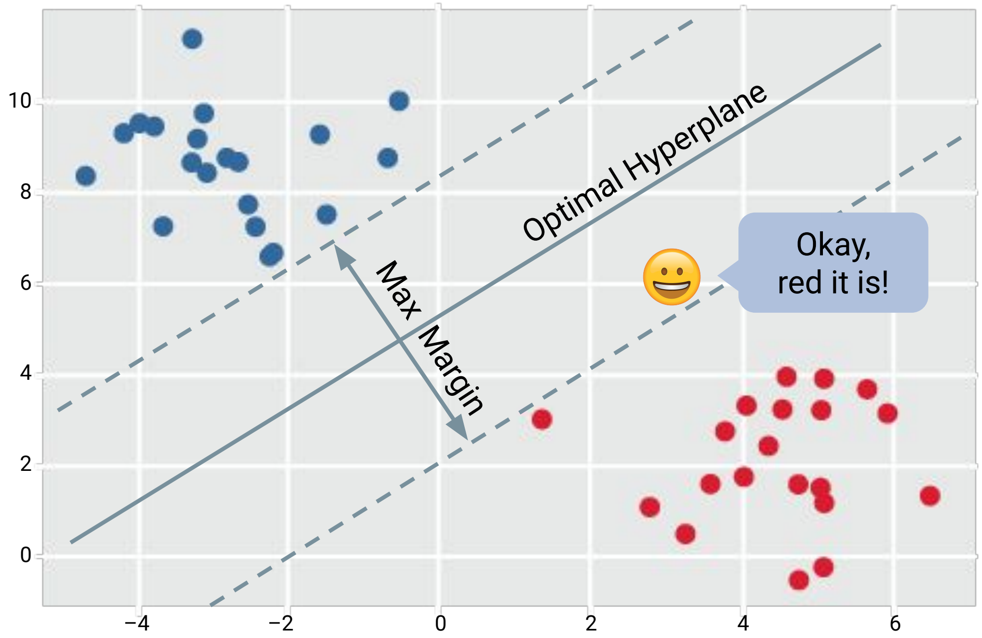
# Linear Classifiers

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



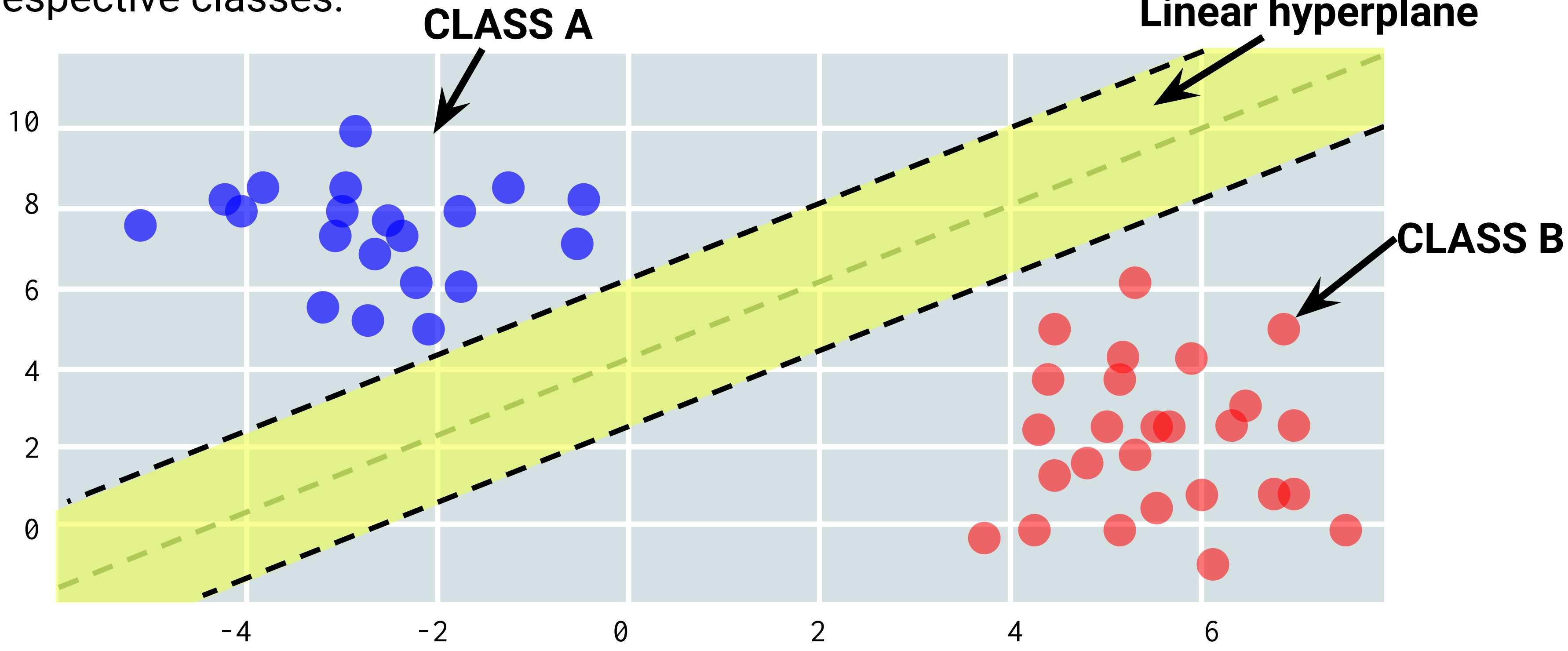
# Support Vector Machines

The SVM algorithm finds the optimal hyperplane that separates the data points with the largest margin possible.



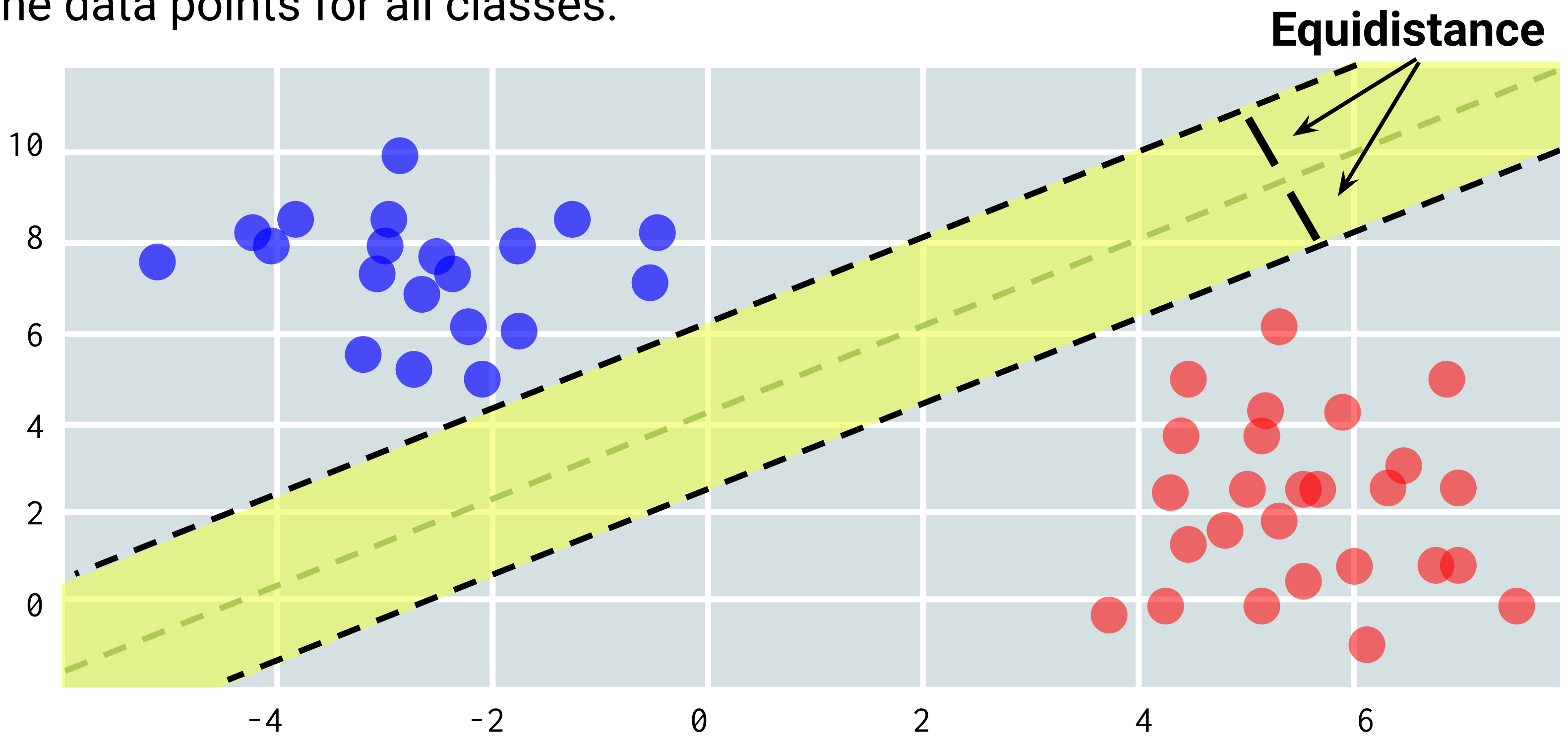
# Support Vector Machines

The space is segmented by a line or plane that groups data points into their respective classes.



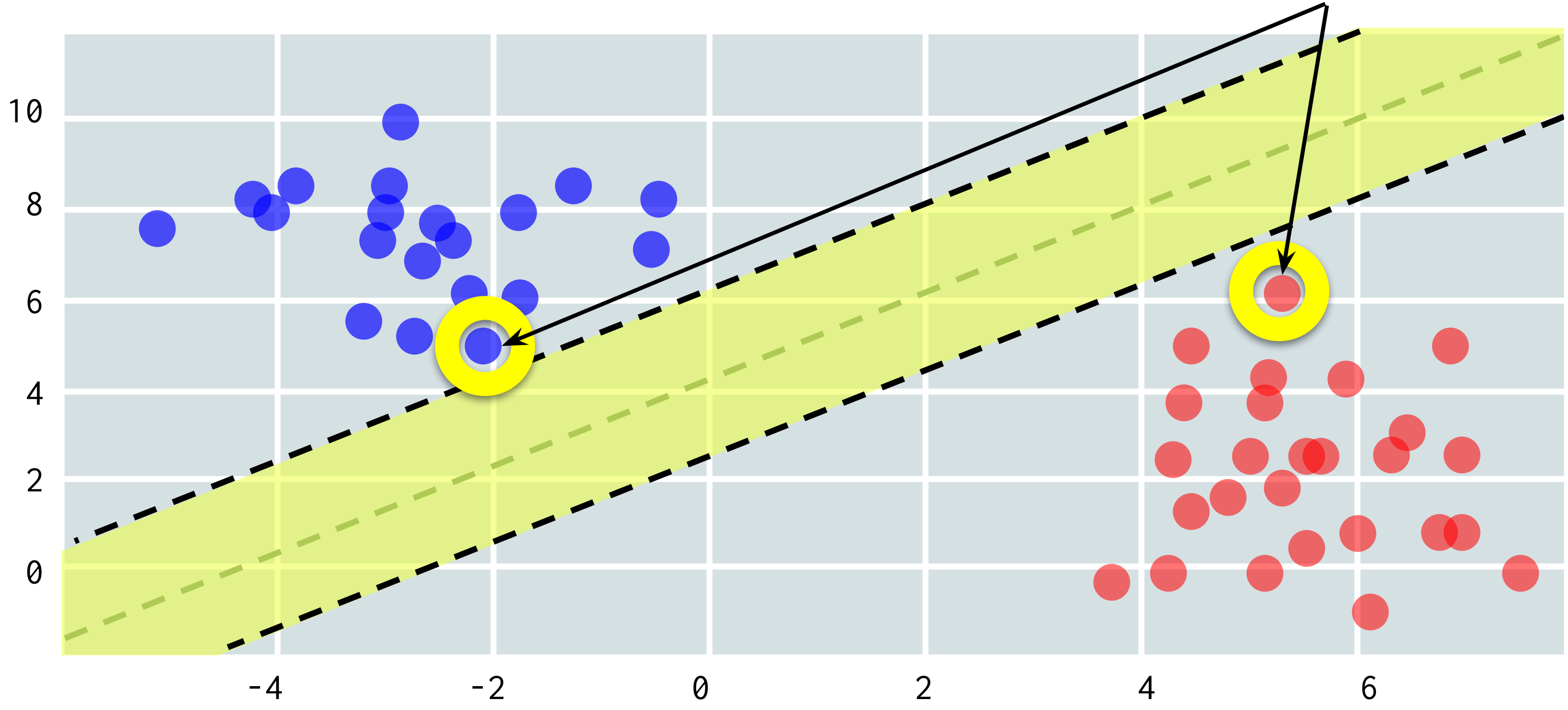
# Support Vector Machines

The goal with hyperplanes is to get the margin of the hyperplane equidistant to the data points for all classes.



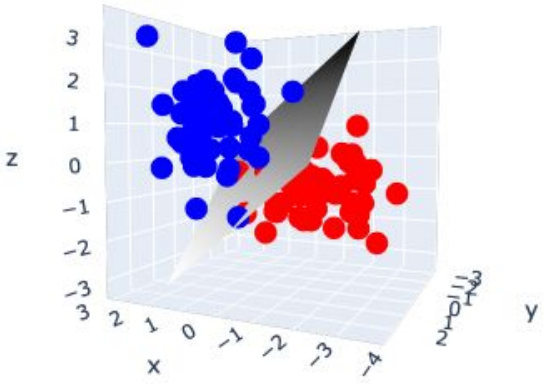
# Support Vector Machines

The data closest to/within the margin of the hyperplane are called support vectors, which are used to define boundaries of the hyperplane.



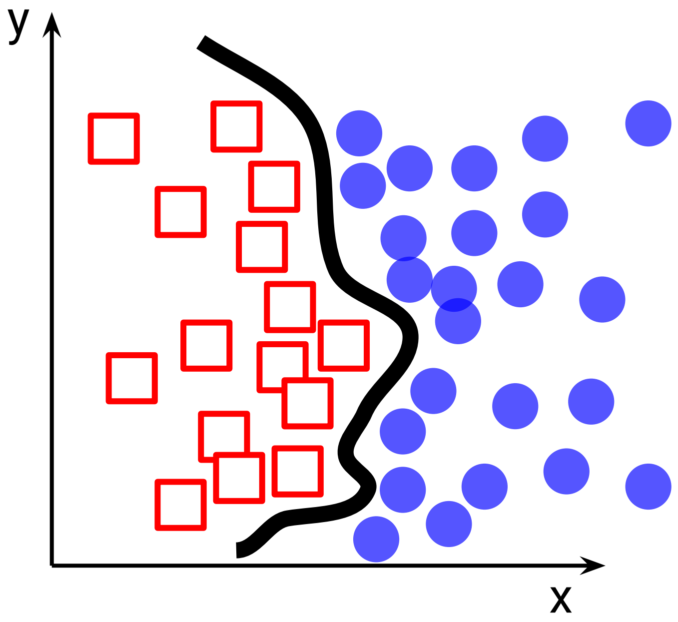
# Hyperplanes

Hyperplanes can be used to clearly delineate classes in multiple dimensions.



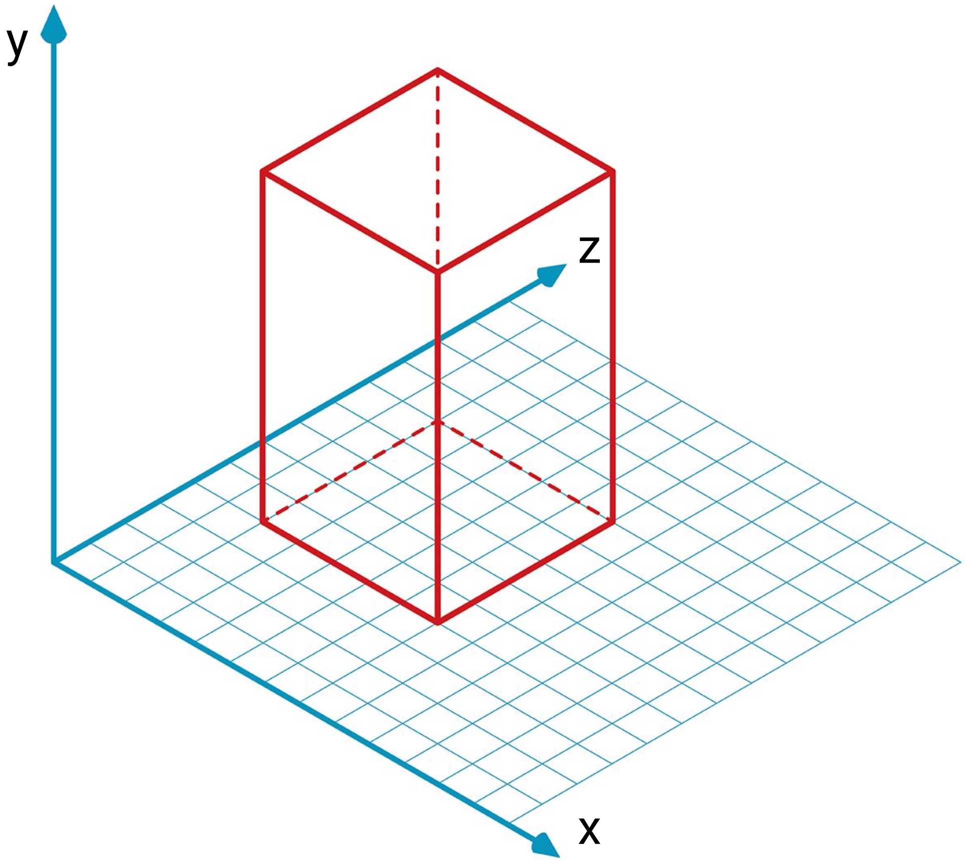
# Zero Tolerance with Perfect Partition

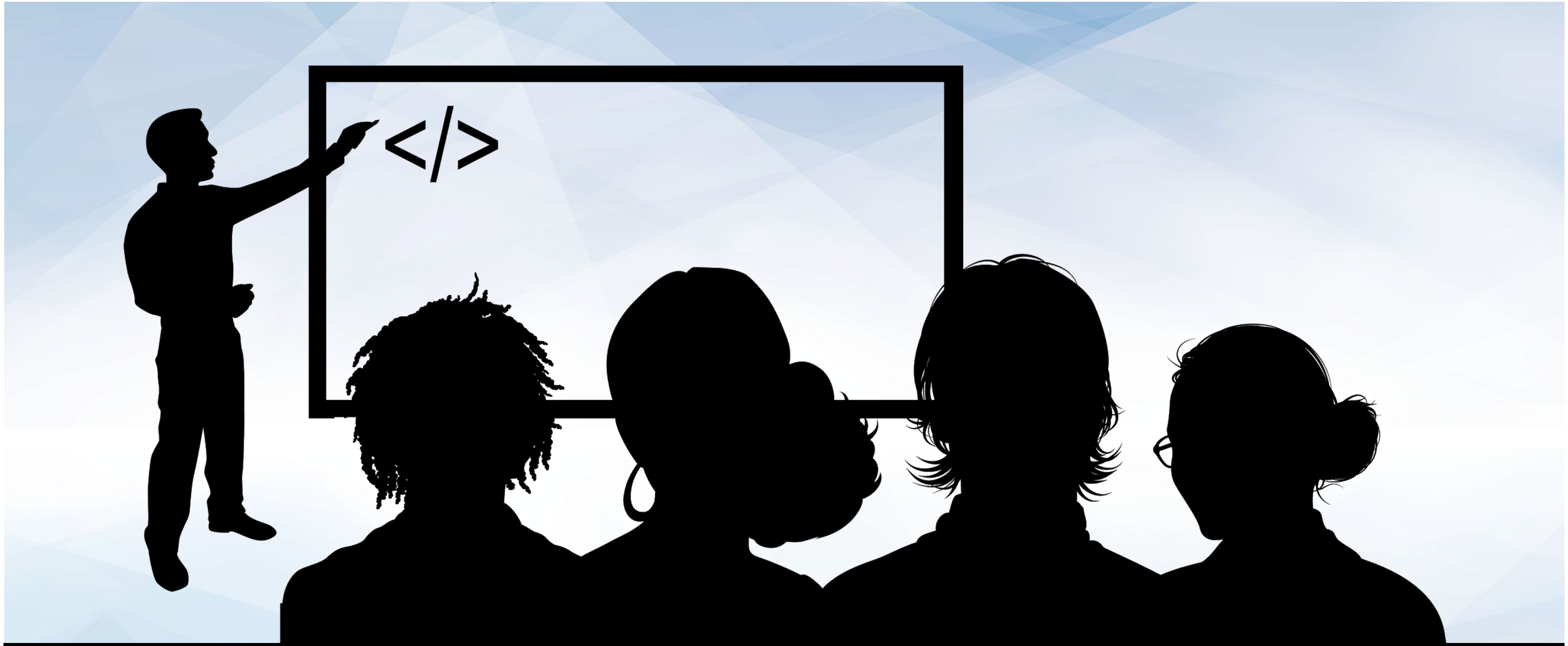
A hyperplane also supports what is considered **zero tolerance with perfect partition**, which is a nonlinear hyperplane that will position and orient the hyperplane to correctly classify overlapping or outlying data points.



# Zero Tolerance with Perfect Partition

In order to establish zero tolerance with perfect partition, the SVM model may introduce a new `z-axis` dimension for nonlinear hyperplanes.






# Instructor Demonstration

## SVM model with sklearn

# SVM model

Steps to implement an SVM model include:

01 Create the model with appropriate `kernel` parameters.

02 Fit the model.

03 Extract min and max decision boundaries and store in a mesh grid.

04 Execute the `decision_function` to get classifier scores for pre-existing data points.

05 Run the `predict` function to classify new data points.

# **Activity:** SVM Loan Approver

In this activity, you will update your loan approver with an SVM model and rerun the evaluation metrics.

**Suggested Time:**
15 minutes

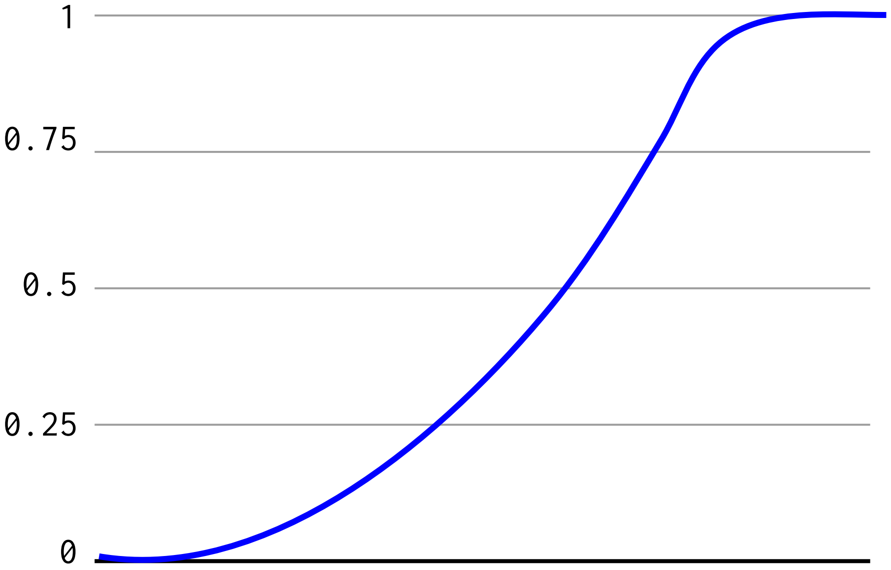
# **Time's Up!** Let's Review.

# Which Model is the Best?

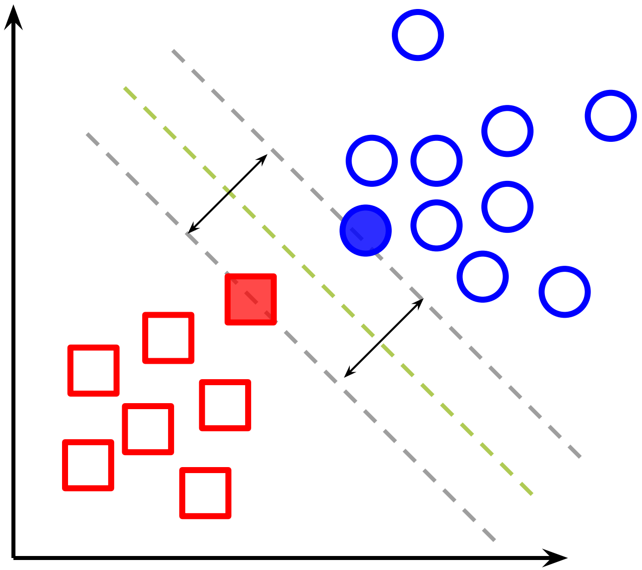
# Which Model is the Best?

Both the logistic regression and SVM models were able to predict outcomes. However, the important question is: **Which model performed best?**

## Logistic Regression




## Support Vector Machines



# What is the best approach to evaluate both models?

# Compare the confusion matrices and classification reports.

## Confusion Matrices:

| n=165 | Predicted: No | Predicted: Yes | |
|---|---|---|---|
| **Actual= No** | 50 | 10 | **=60** |
| **Actual= Yes** | 5 | 100 | **=105** |
| | **=55** | **=110** | |

## Classification Reports:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| No Diabetes | 0.77 | 0.90 | 0.83 | 125 |
| Diabetes | 0.72 | 0.49 | 0.58 | 67 |
| accuracy | | | 0.76 | 192 |
| macro avg | 0.74 | 0.69 | 0.71 | 192 |
| weighted avg | 0.75 | 0.76 | 0.74 | 192 |

# What is the best approach to evaluate both models?

## Logistic Regression Loan Approver Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| approve | 0.44 | 0.33 | 0.38 | 12 |
| deny | 0.50 | 0.62 | 0.55 | 13 |
| micro avg | 0.48 | 0.48 | 0.48 | 25 |
| macro avg | 0.47 | 0.47 | 0.47 | 25 |
| weighted avg | 0.47 | 0.48 | 0.47 | 25 |

## SVM Loan Approver Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| approve | 0.58 | 0.58 | 0.58 | 12 |
| deny | 0.62 | 0.62 | 0.62 | 13 |
| accuracy | | | 0.60 | 25 |
| macro avg | 0.60 | 0.60 | 0.60 | 25 |
| weighted avg | 0.60 | 0.60 | 0.60 | 25 |

# What is the best approach to evaluate both models?

The SVM model performed best. **Precision**, **recall**, and **accuracy** were all higher for the SVM loan approver.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| approve | 0.44 | 0.33 | 0.38 | 12 |
| deny | 0.50 | 0.62 | 0.55 | 13 |
| micro avg | 0.48 | 0.48 | 0.48 | 25 |
| macro avg | 0.47 | 0.47 | 0.47 | 25 |
| weighted avg | 0.47 | 0.48 | 0.47 | 25 |

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| approve | 0.58 | 0.58 | 0.58 | 12 |
| deny | 0.62 | 0.62 | 0.62 | 13 |
| accuracy | | | 0.60 | 25 |
| macro avg | 0.60 | 0.60 | 0.60 | 25 |
| weighted avg | 0.60 | 0.60 | 0.60 | 25 |

# What is the best approach to evaluate both models?

Recall percentage for deny is the same for the SVM and logistic regression loan approver, meaning both algorithms correctly predicted the same number of true positive denies.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| approve | 0.44 | 0.33 | 0.38 | 12 |
| deny | 0.50 | 0.62 | 0.55 | 13 |
| micro avg | 0.48 | 0.48 | 0.48 | 25 |
| macro avg | 0.47 | 0.47 | 0.47 | 25 |
| weighted avg | 0.47 | 0.48 | 0.47 | 25 |

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| approve | 0.58 | 0.58 | 0.58 | 12 |
| deny | 0.62 | 0.62 | 0.62 | 13 |
| accuracy | | | 0.60 | 25 |
| macro avg | 0.60 | 0.60 | 0.60 | 25 |
| weighted avg | 0.60 | 0.60 | 0.60 | 25 |

# Classification Homework

Due on Tuesday, August 25

# Questions?