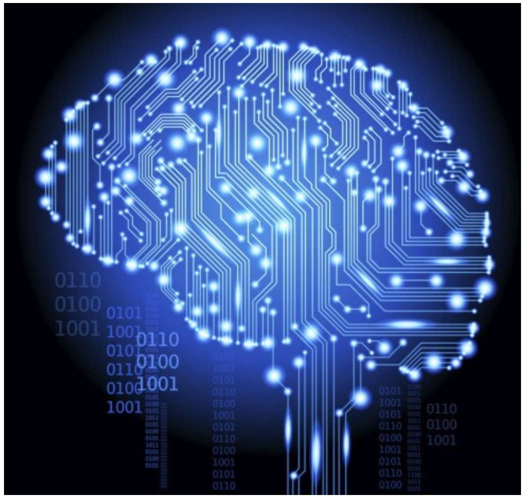
Machine Learning

Artificial Intelligence

Artificial Intelligence

- Intelligence demonstrated by machines or software, and the study of how to make them that way.
- Reasoning, knowledge, planning, learning, natural language processing, perception.
- What does it mean to "act intelligent"?



The Turing Test

- Operational definition of intelligence
- Proposed by Alan Turing in 1950
- Test of a computers ability to exhibit intelligent behavior equivalent to that of a human.

Alan Turing

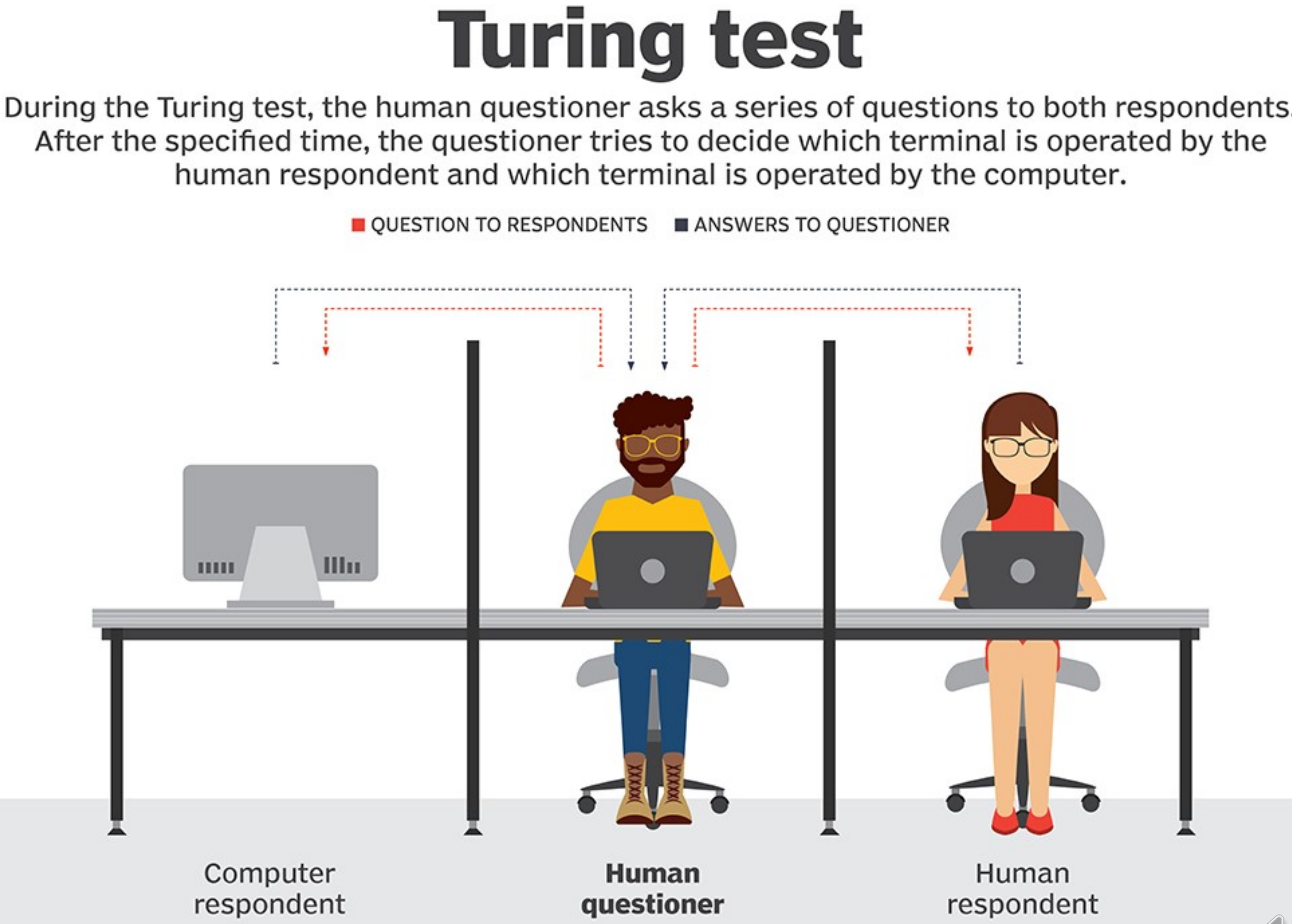
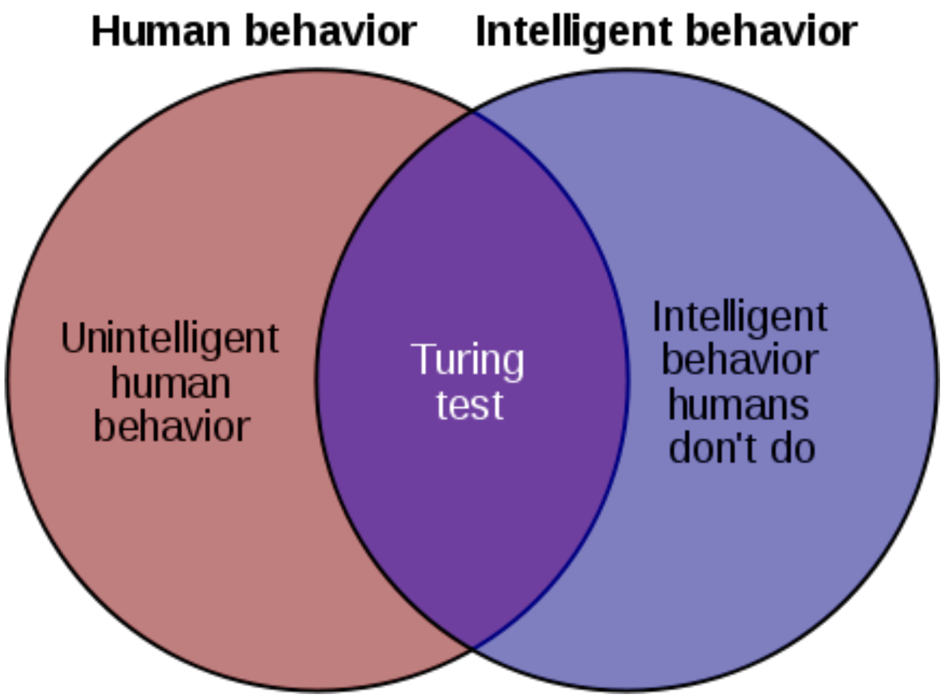


Image from: http://whatis.techtarget.com/definition/Turing-Test

The Turing Test: Problems

Human intelligence vs intelligence in general

Turing Test evaluates if a computer behaves like a human. Not necessarily the same as being intelligent.



The Turing Test: Problems

Consciousness vs. the simulation of consciousness

- Is the computer "thinking" or just following hard coded directions (simulating thought)
- Turing test only evaluates external behaviour, may not be possible to differentiate between consciousness and simulated consciousness based solely on external behaviour.

The Turing Test: Problems

Dependence on Human Questioner

- A naïve questioner may be easily fooled.
- Numerous cases of unsuspecting questioners being fooled by chatterbots.
- Humans tend to consider non-human objects as human (anthropomorphism).

The Turing Test: Now

- Considered impractical and irrelevant by most mainstream AI researchers.
- Better to test to use less subjective tests based on the goal of the research: object recognition, natural language comprehension, automation, etc.

A Brief History of AI

Birth 1950-1956:

- Isaac Asimov's "I, Robot", rudimentary game AI, Turing test, term "artificial intelligence" coined.

Golden Years 1957-1974:

- MIT AI lab setup, assembly line robots, natural language programs, chatbots, first autonomous vehicle.

1st AI Winter 1974-1980:

- Limited computing power, reduced funding, lack of public interest.

A Brief History of AI

Boom 1980-1987:

- Expert systems, knowledge-based, 5th gen project

2nd AI Winter 1987-1993:

- Funding cuts, rise of PCs vs special hardware, robotics

Modern Era 1993-present:

- Increased computer power, specialized for problems, statistical methods, intelligent agents, modern machine learning approaches.

A Brief History of AI

Recent Accomplishments:

1997 IBM Deep Blue Chess

2005 Stanley robot DARPA

2011 Watson Jeopardy

2016 Google AlphaGo Go

Optional Reading: The History of Artificial Intelligence
https://en.wikipedia.org/wiki/History_of_artificial_intelligence

Optional Reading: Alibaba's AI Outguns Humans in Reading Test
https://www.bloomberg.com/news/articles/2018-01-15/alibaba-s-ai-outgunned-humans-in-key-stanford-reading-test

Machine Learning

- Branch of AI
- Systems that can learn from data
- Train with data
- Given some examples, generalize to categories.
- **Examples:**
  - **Spam detection:** Classify e-mail as spam vs non spam
  - **Object recognition:** Does this video/image contain a face? cat? car? etc.
  - **Facial recognition:** Identifying some by a picture of their face.
  - **Handwriting recognition**
  - **Speech recognition**

Machine Learning

Traditional Solutions

- Rule-based systems – no learning
- **Example:** Sentiment analysis based on a simple heuristic (>0 positive, =0 neutral, <0 negative)

Machine Learning

- Find patterns based on **features** of data
- Adapt to new and unforeseen situations
- Learn from mistakes
- Draw new conclusions

Classification

Classification is the process of assigning a class to something according to shared qualities or characteristics (**features**).

Classification Methods:

- **Decision Trees**
- *k*-Nearest Neighbors
- Support Vector Machines (SVM)
- Neural networks

The Classification Problem

Have a population that may be partitioned into classes.

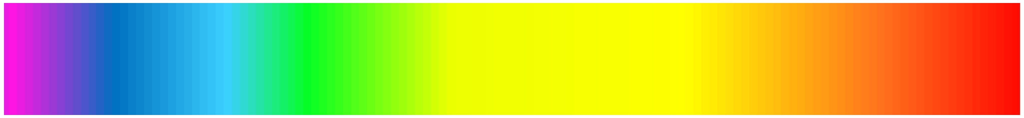
E.g. spam vs non-spam; handwritten digits 0,1,…9

Have a number of classified instances (examples), each of which has a number of attributes (**features**).

Look at the examples and come up with a method to classify things that have not yet been seen.

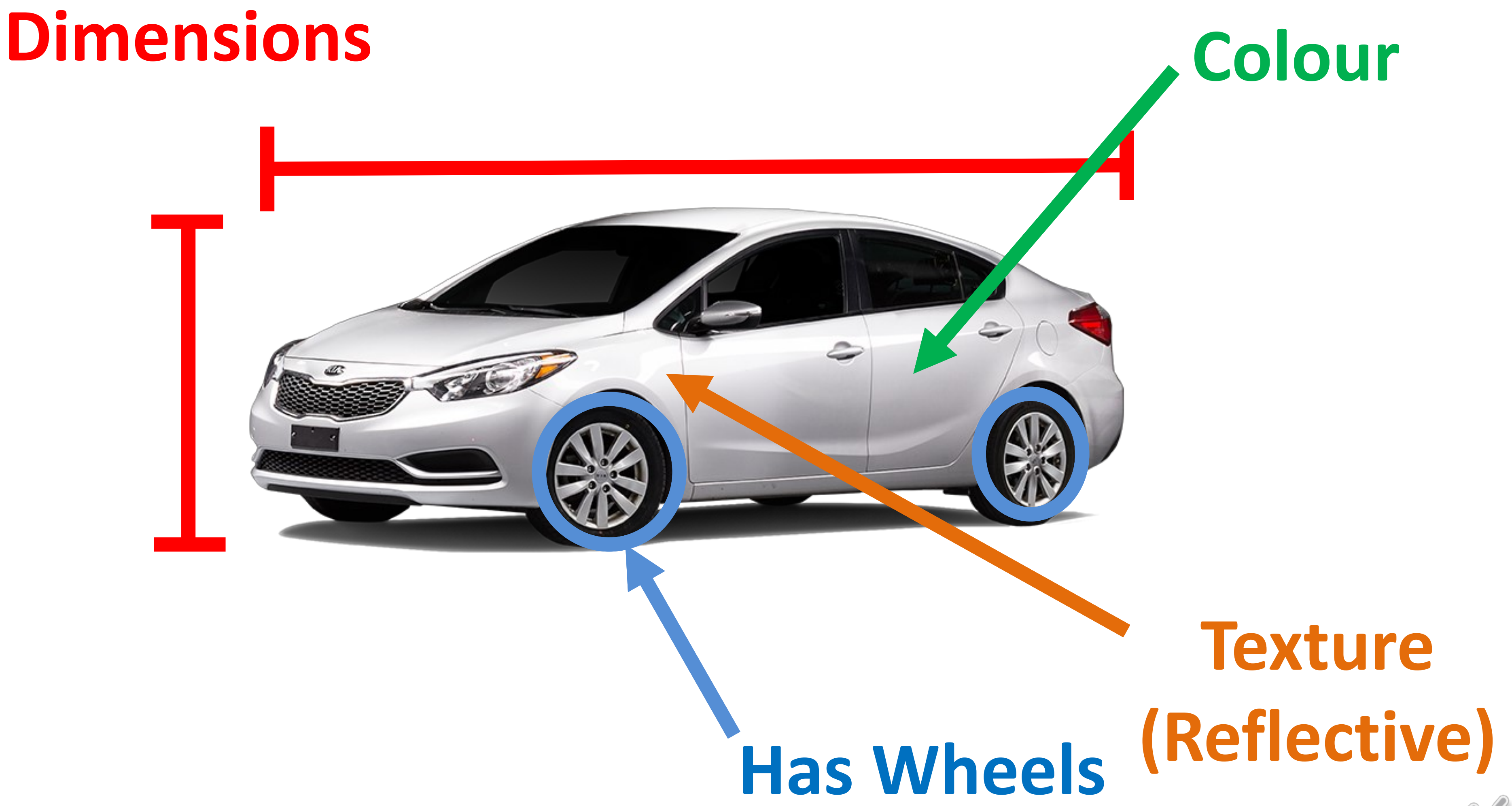
Features

- Individual measurable property or characteristic of object being observed.

- May be represented by many different values:
  - Boolean {Yes, No} or {True, False}
  - Labels {Car, Boat, Plane, Duck, Cat}
  - A color {  }
  - Integers {0, 1, 2, …}
  - Real numbers {1.34, 7.45, …}
  - Percentage (between 0 and 1, e.g. 0.123)
  - Distance, area, time, velocity, etc.

Features

Examples



Features

How do you select the appropriate set of **features** for classification?

We need to have some knowledge of what **features** make good predictors of **class** membership for the **classes** we are trying to distinguish.

For example, having wheels distinguishes people from cars, but doesn't distinguish cars from trains.

Supervised vs. Unsupervised

Two Approaches

Supervised Learning:

- Learn by example.
- Provide training examples that include both the inputs (the data) and the correct outputs (the classifications).

Unsupervised Learning:

- Learn solely based on the structure of the data.
- Provide only the inputs (the data) and try to derive the underlying structure and relationships between features of the data.

Toy Problem

Classify a picture as a **rubber duck** or **not a rubber duck**.

Classes:

- **Rubber duck.**
- **Not a rubber duck.**

Features: ?

What features do you think might be useful for this problem?

Toy Problem: Features

Images contain many features we could choose from.

Toy Problem: Features

Color

Average Color

Red = 241

Green = 207

Blue = 103

Toy Problem: Features

Color

Most Common Colors

Color	Color Code	Percentage
	#ffffff	0.288382
	#f0c000	0.194902
	#d8a800	0.140980
	#ffd800	0.080245
	#c09000	0.040490
	#f06018	0.036667
	#ffd818	0.036029
	#f0d818	0.030245
	#f06000	0.030098
	#fffff0	0.027843

Toy Problem: Features

Shape

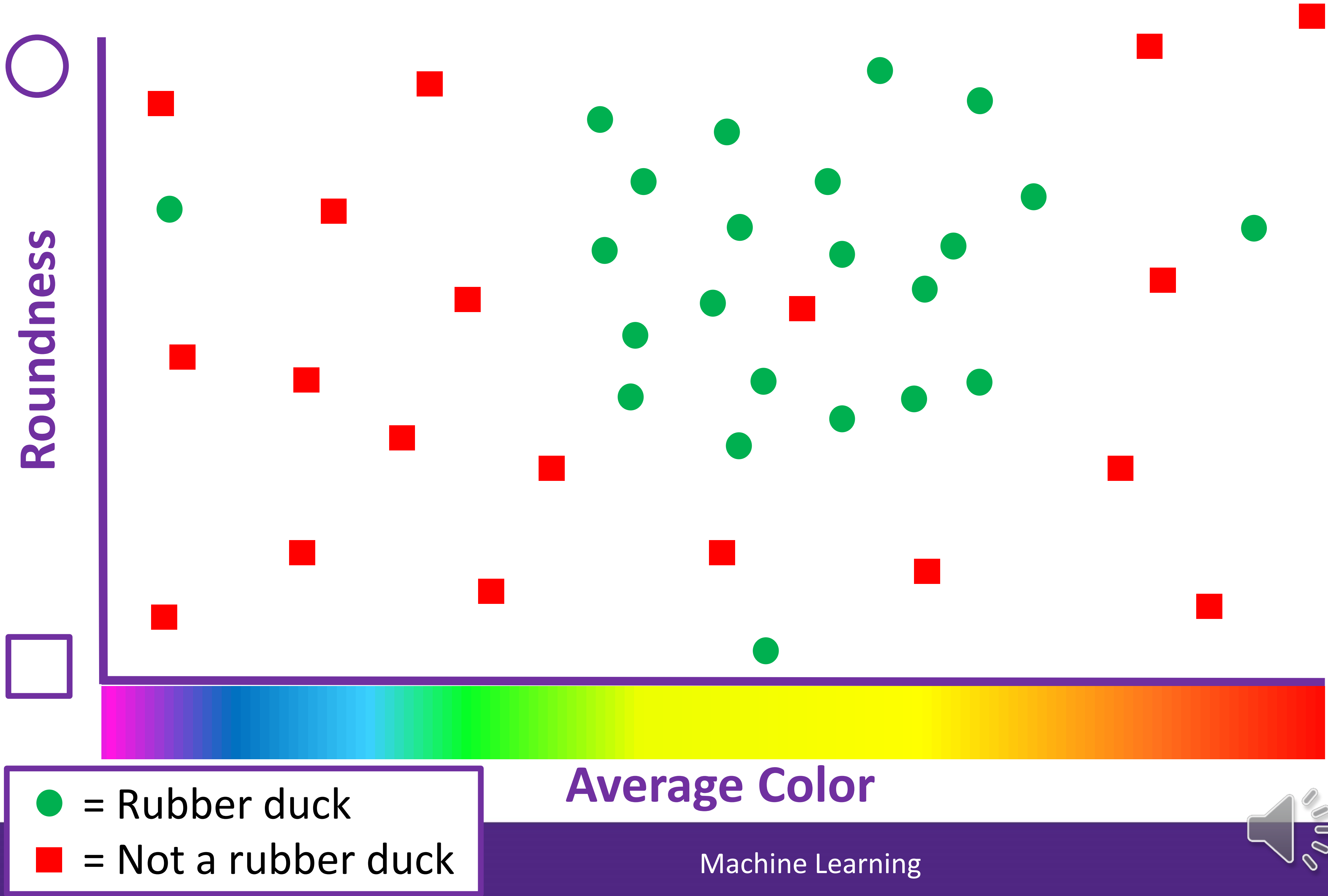
Roundness of Silhouette

0.75

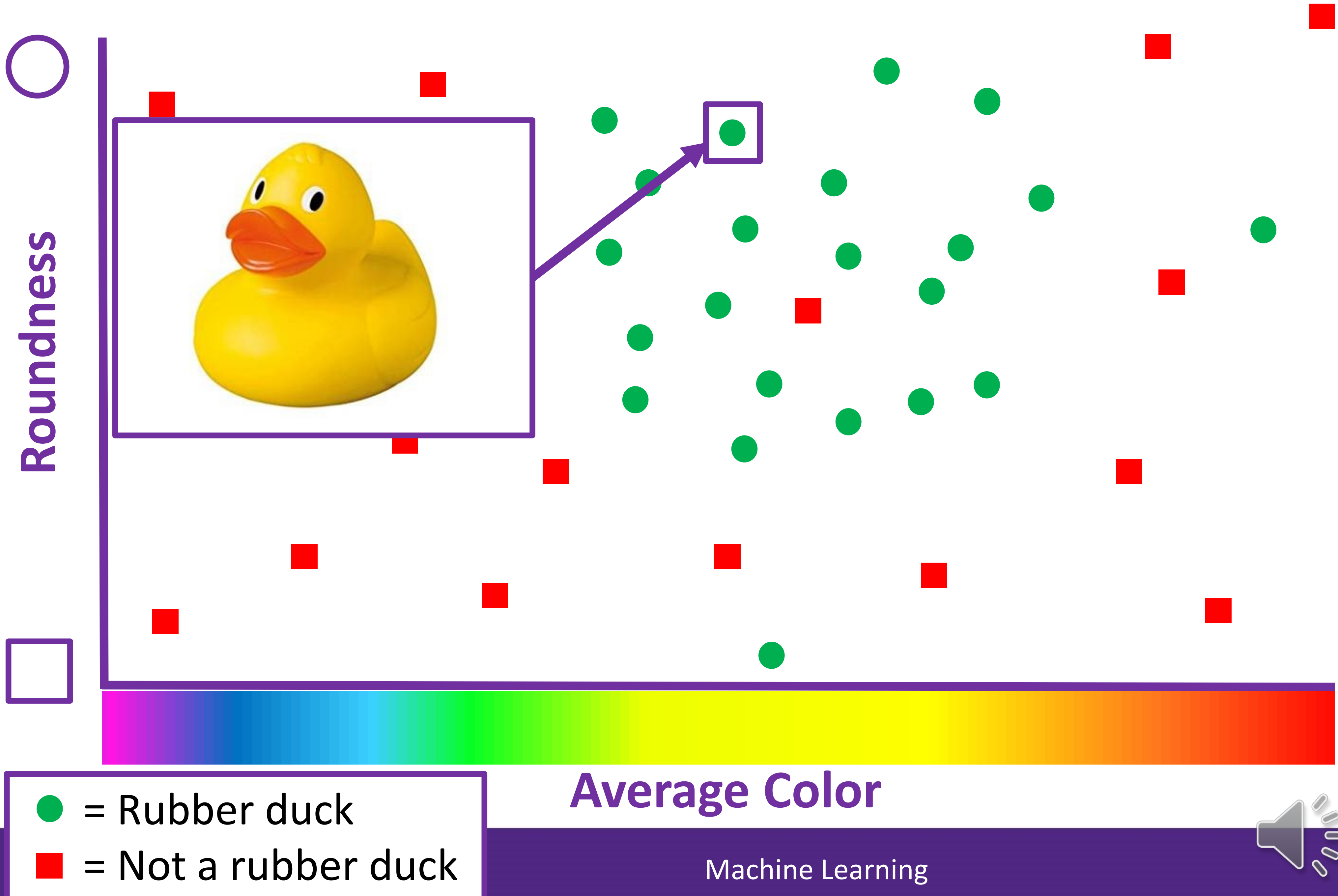
Toy Problem: Features

Thousands of possibilities but we will keep it simple for this example and just use **average color** and **"roundness"**.

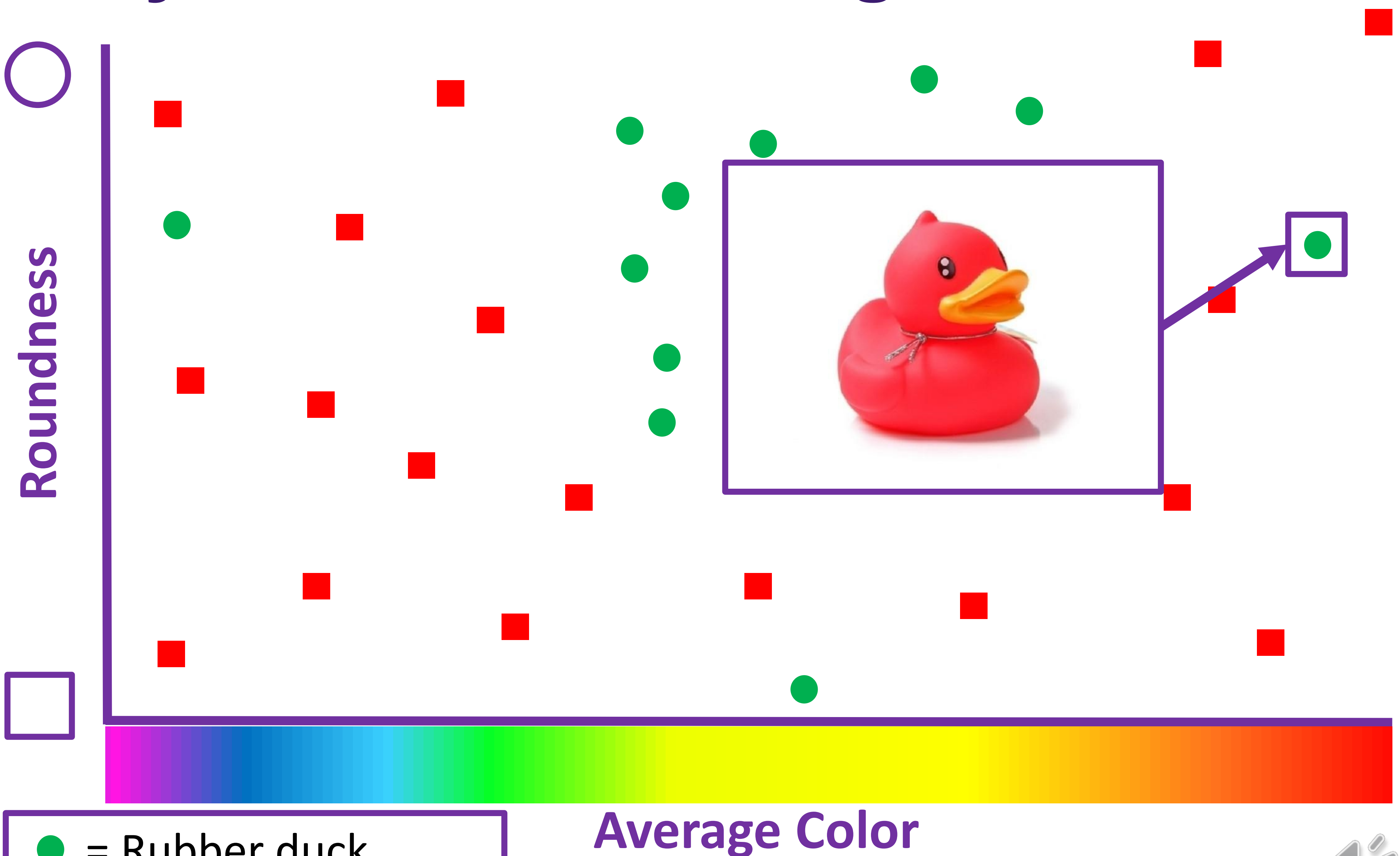
Toy Problem: Training Set



Toy Problem: Training Set

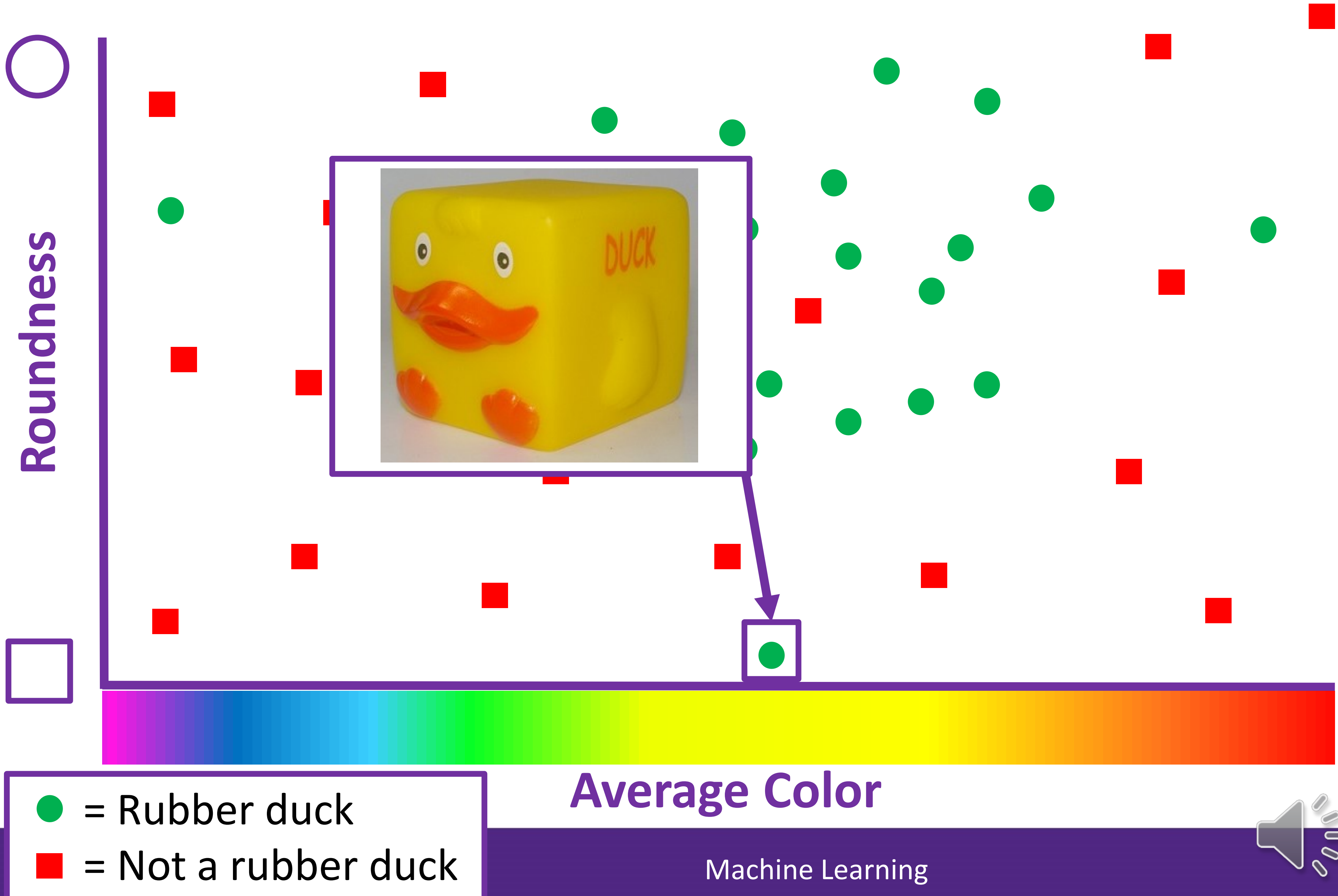


Toy Problem: Training Set

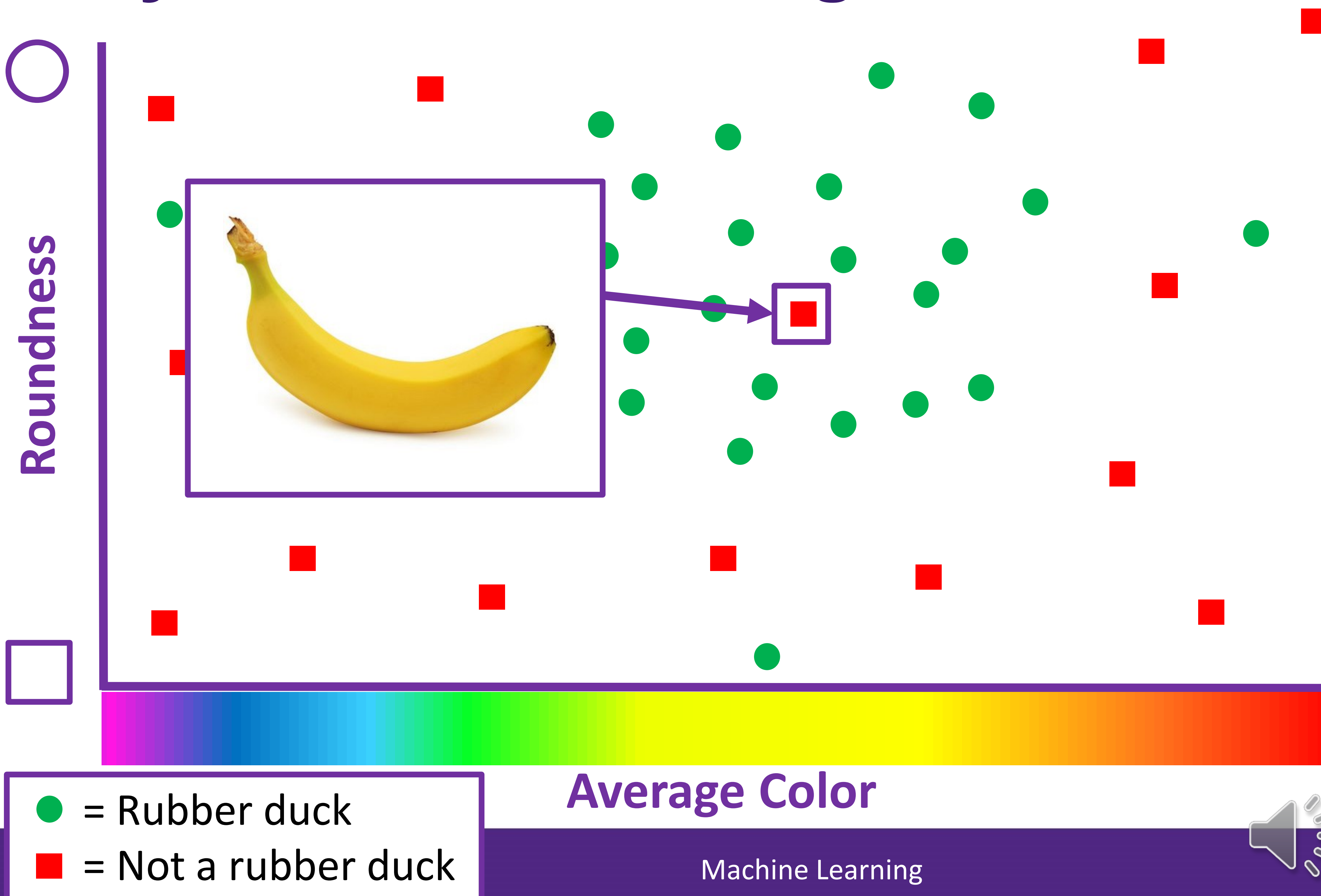


= Rubber duck
= Not a rubber duck

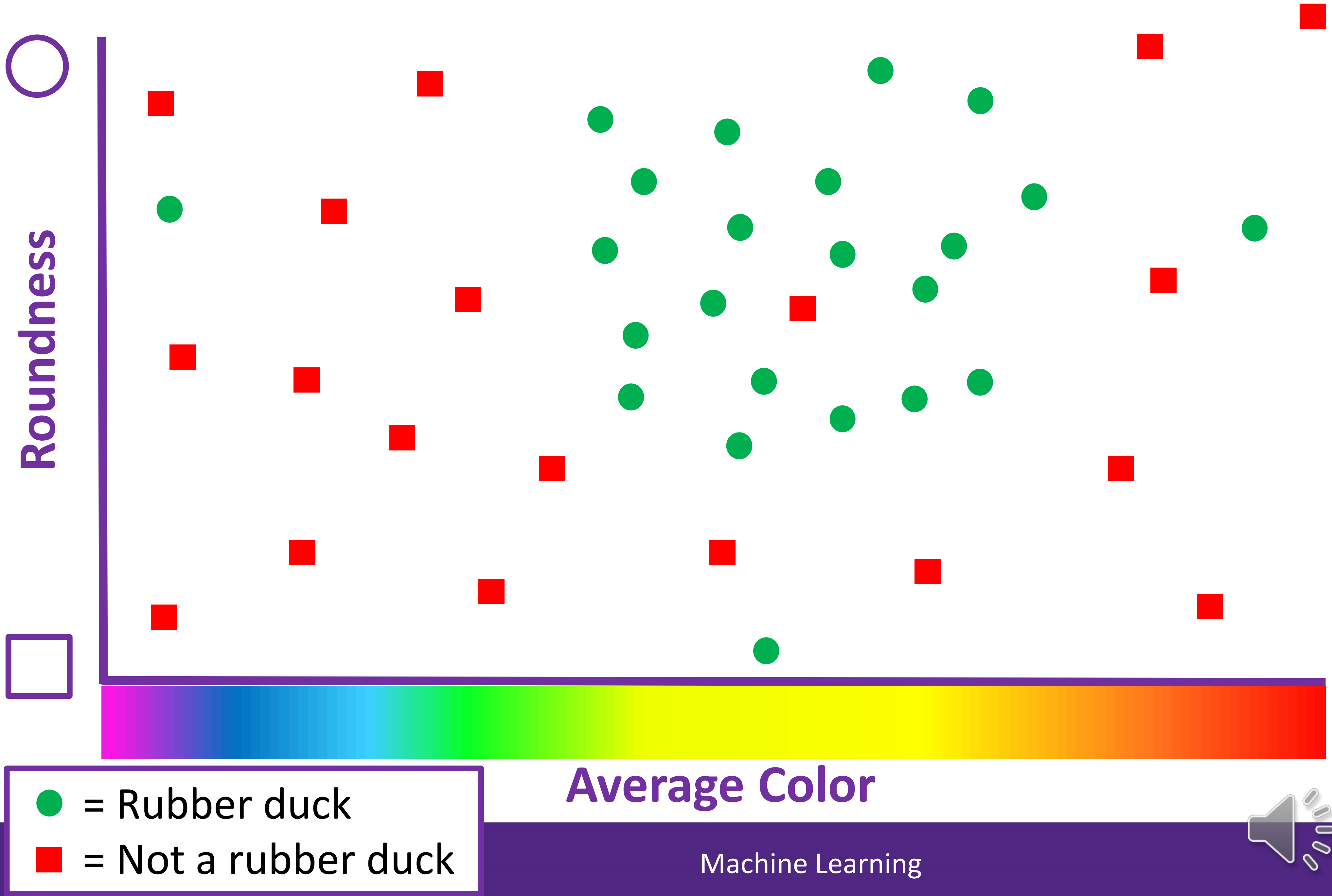
Toy Problem: Training Set



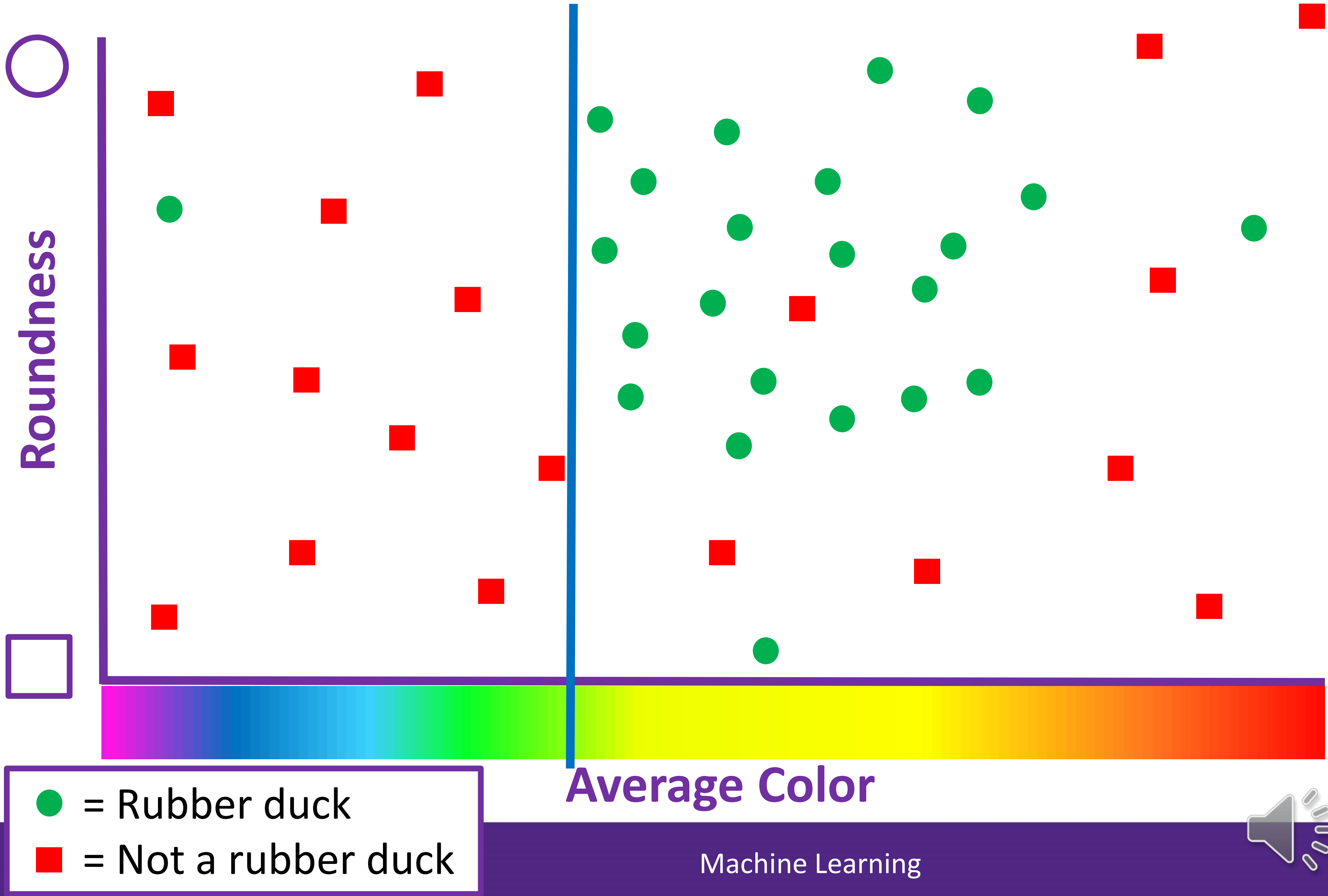
Toy Problem: Training Set



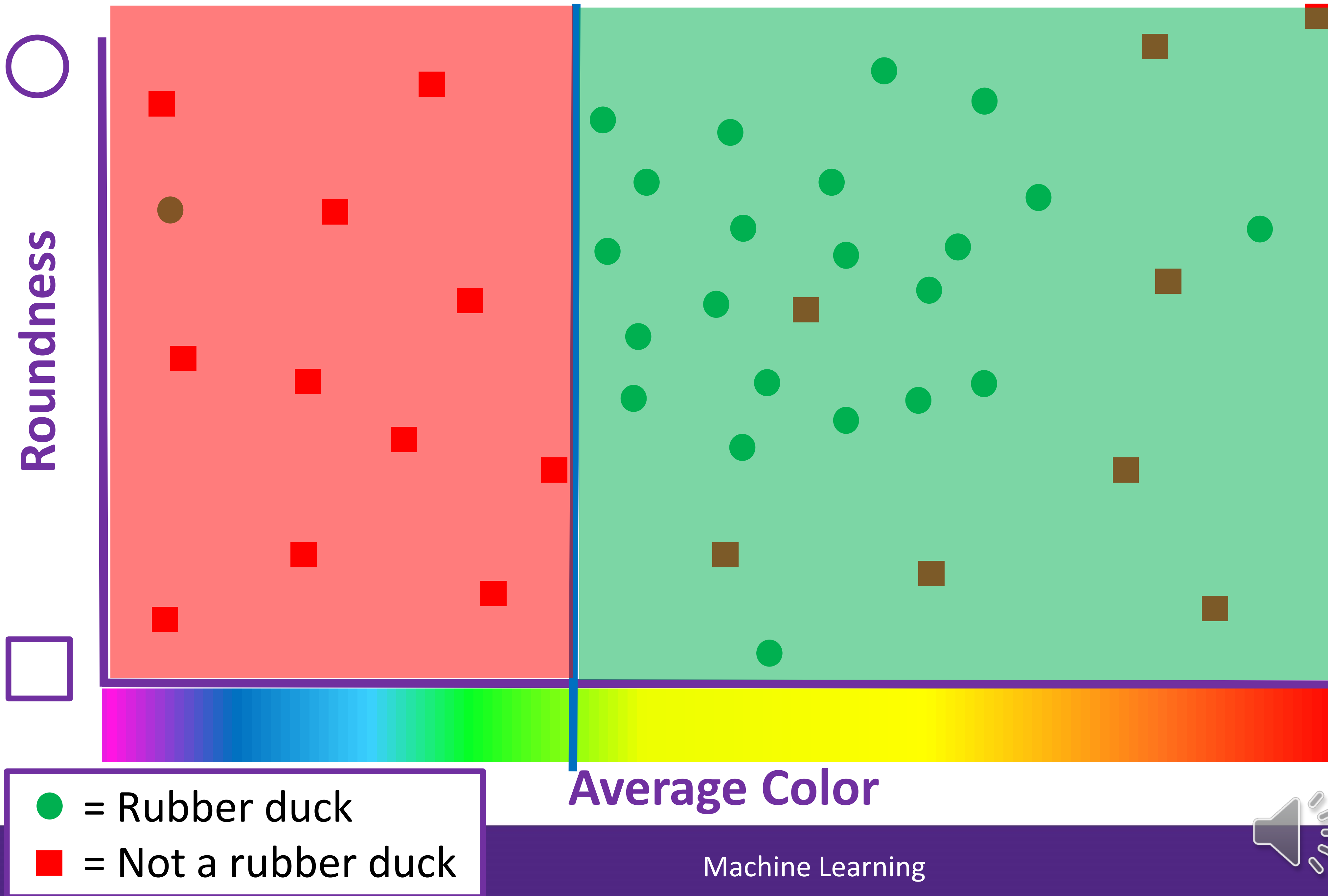
Toy Problem: Linear Classifier



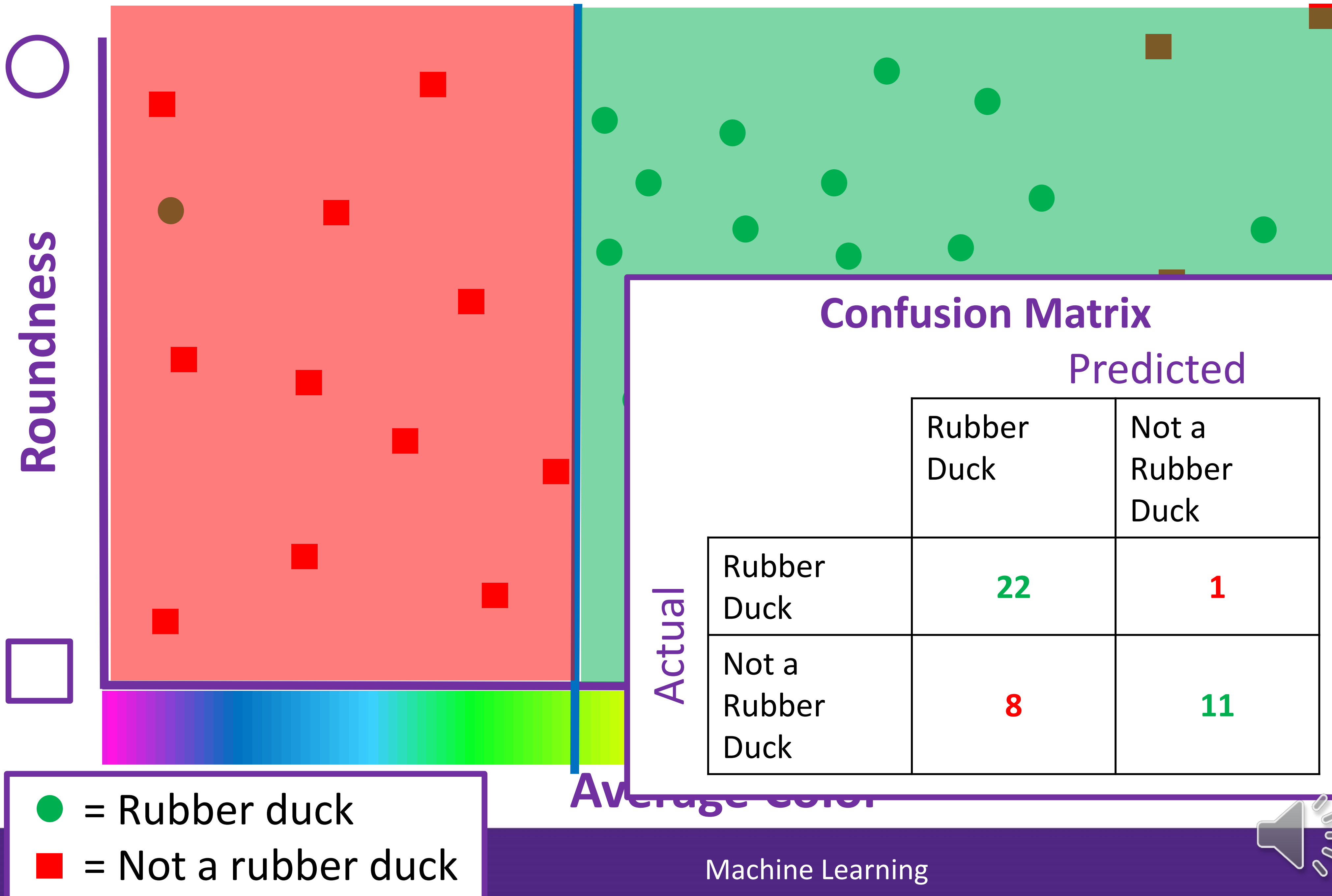
Toy Problem: Linear Classifier



Toy Problem: Linear Classifier



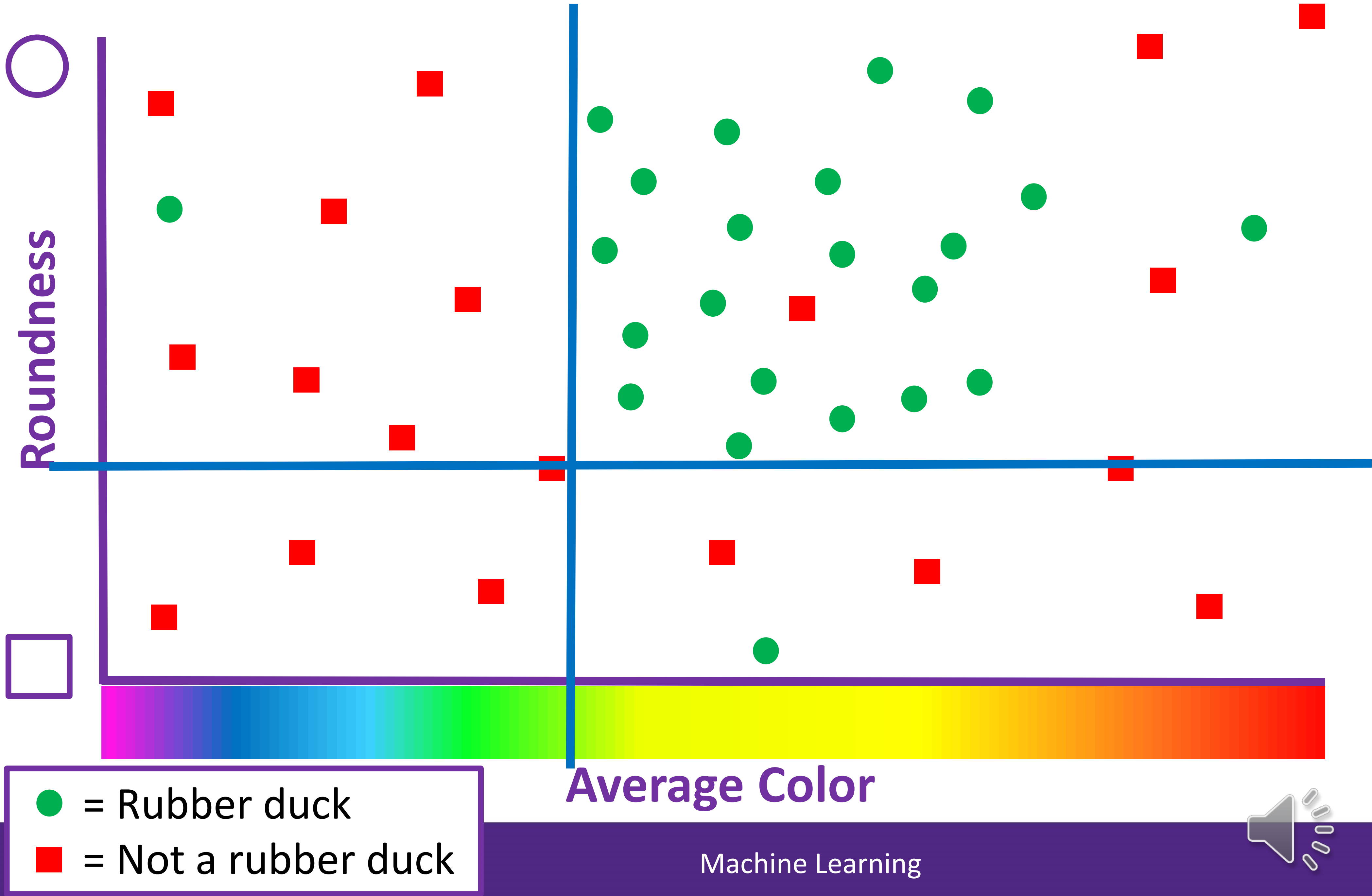
Toy Problem: Linear Classifier



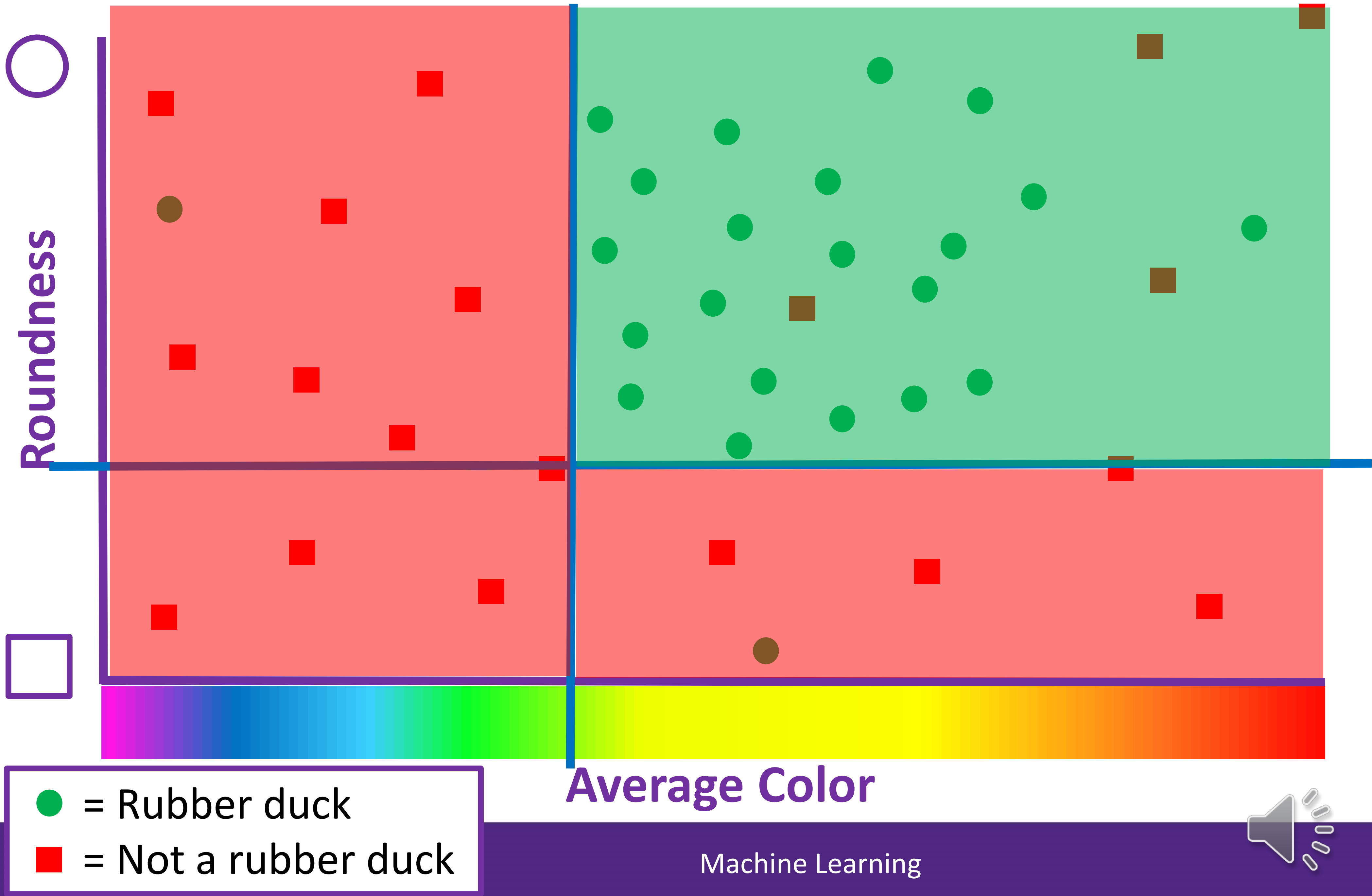
Confusion Matrix

Actual \ Predicted	Rubber Duck	Not a Rubber Duck
Rubber Duck	22	1
Not a Rubber Duck	8	11

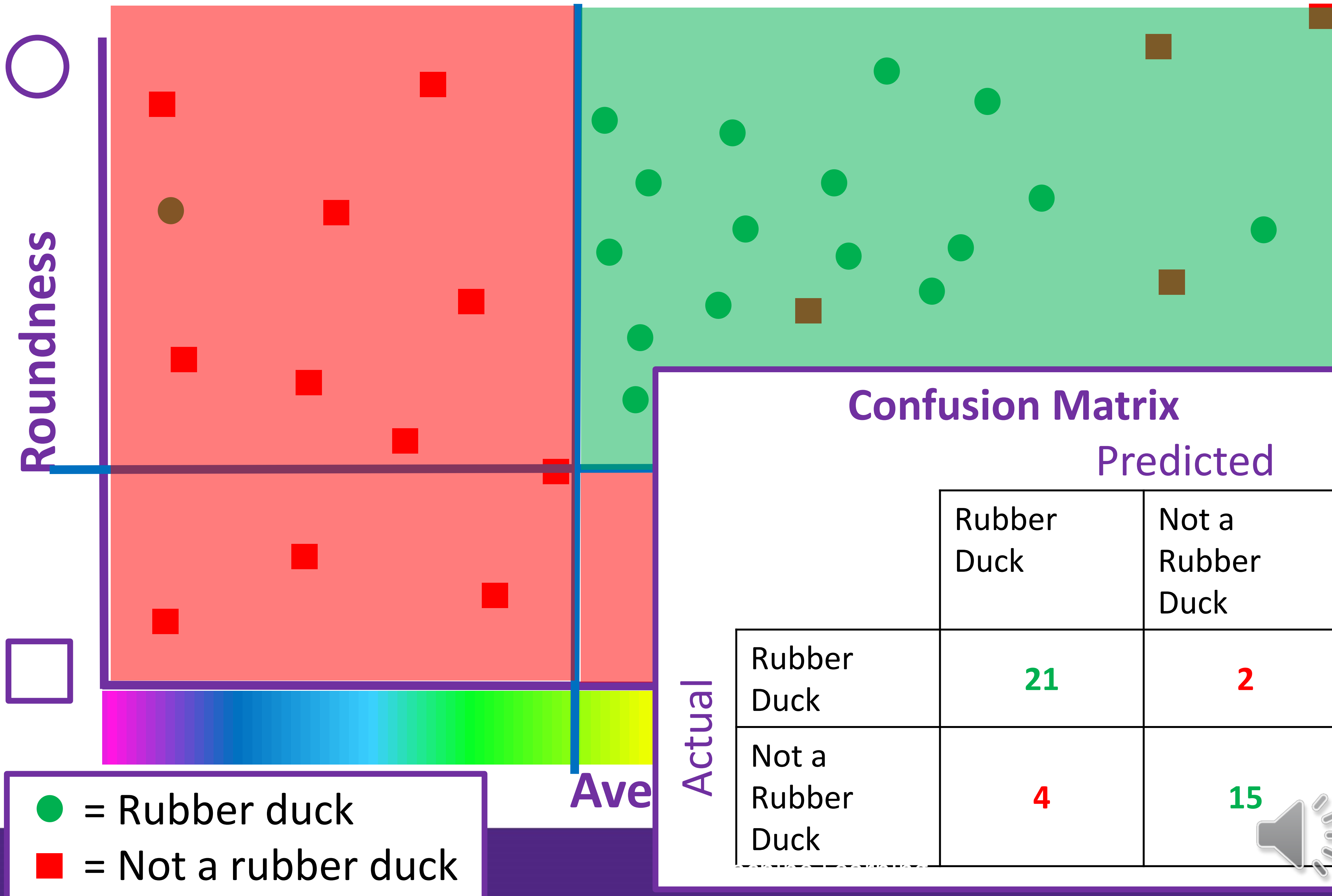
Toy Problem: Linear Classifier



Toy Problem: Linear Classifier



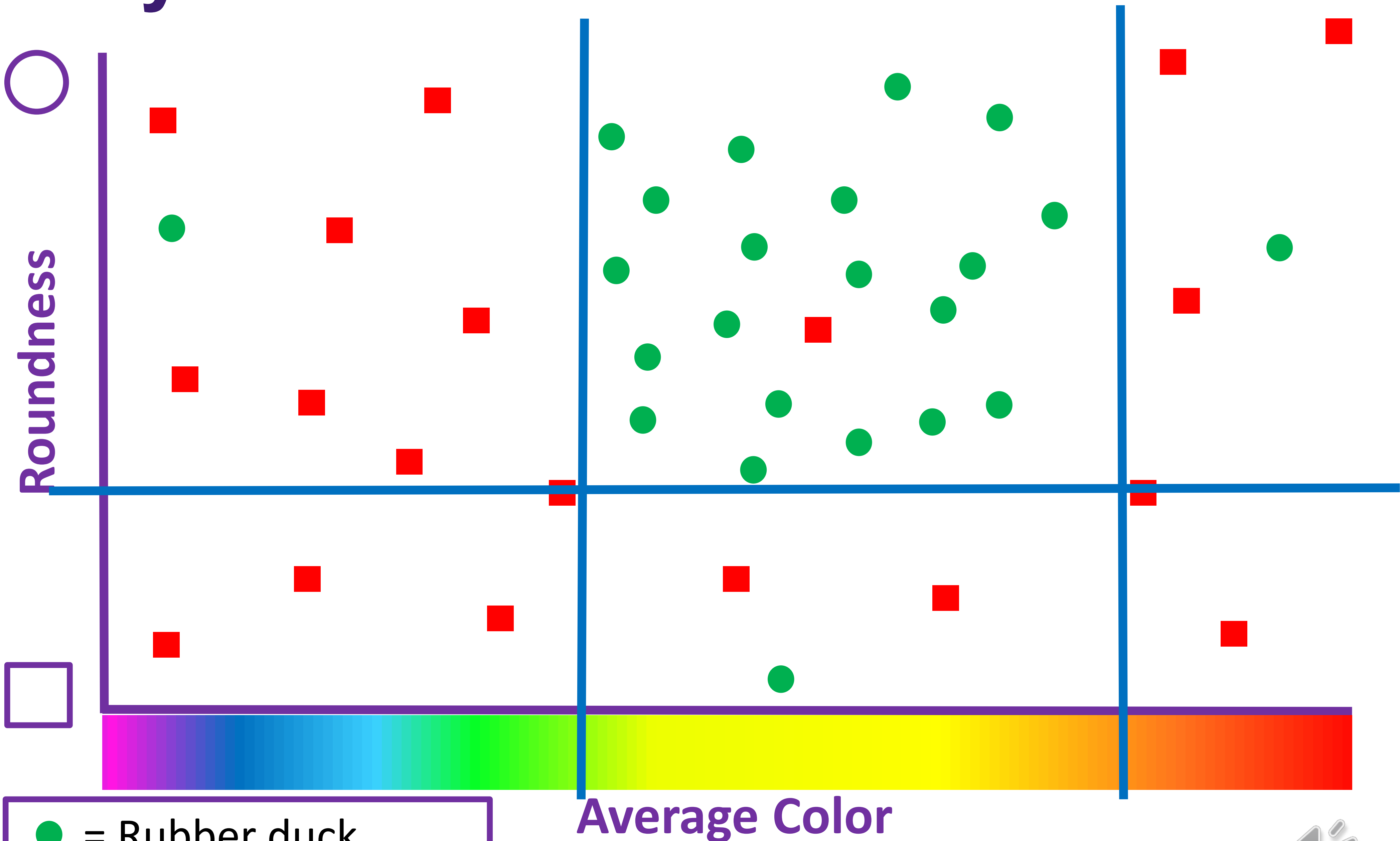
Toy Problem: Linear Classifier



Confusion Matrix

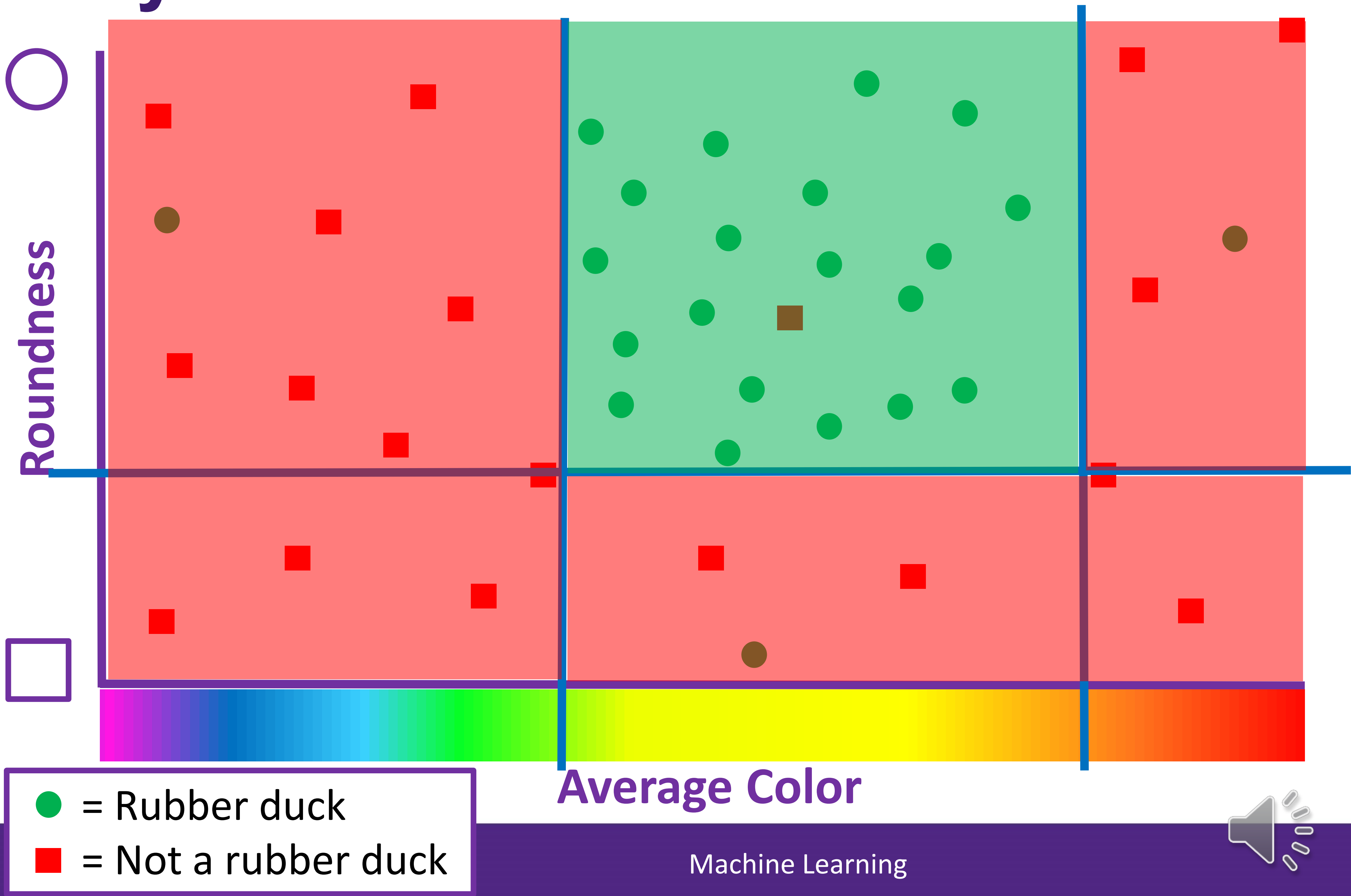
		Predicted	
		Rubber Duck	Not a Rubber Duck
Actual	Rubber Duck	21	2
	Not a Rubber Duck	4	15

Toy Problem: Linear Classifier

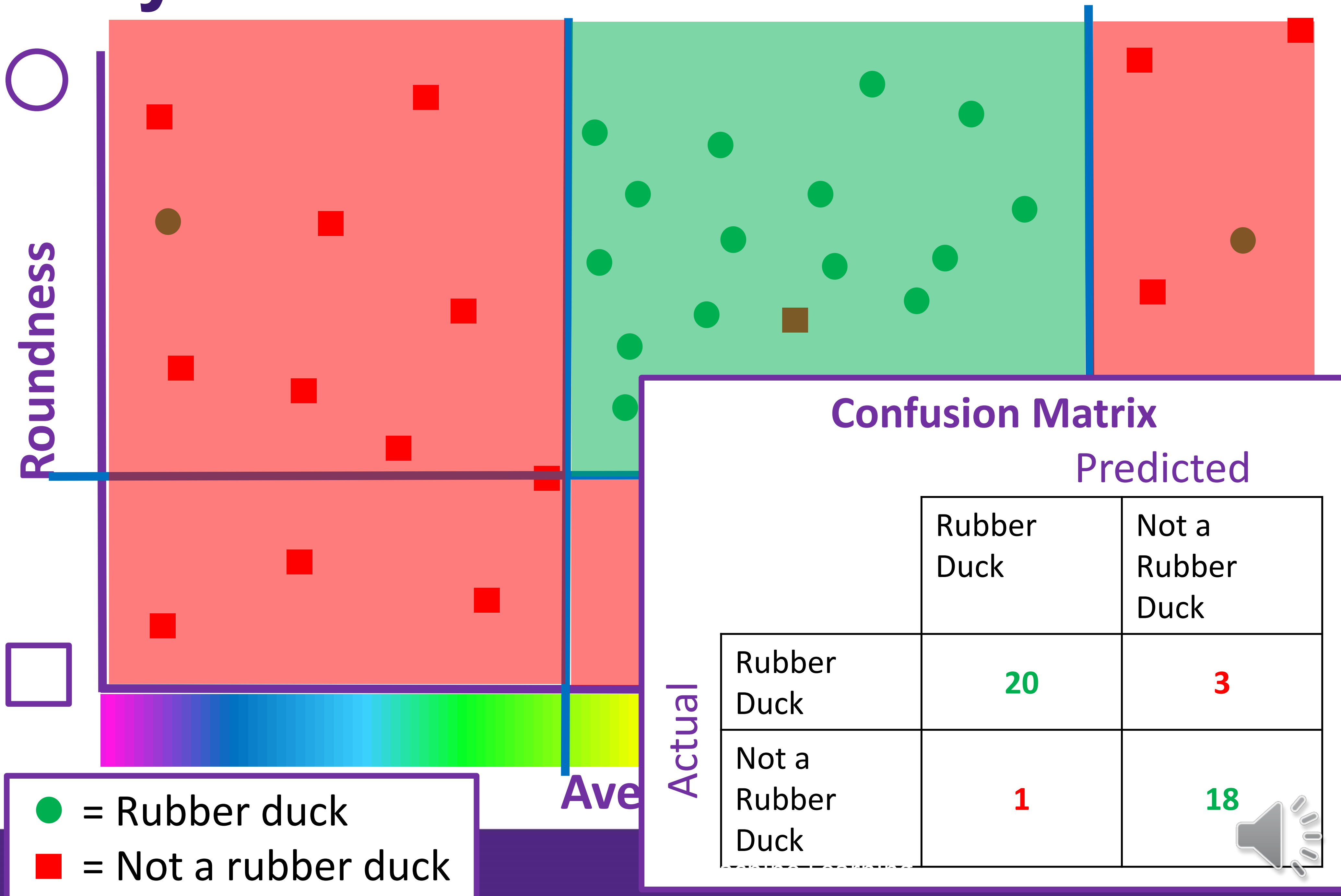


● = Rubber duck
■ = Not a rubber duck

Toy Problem: Linear Classifier



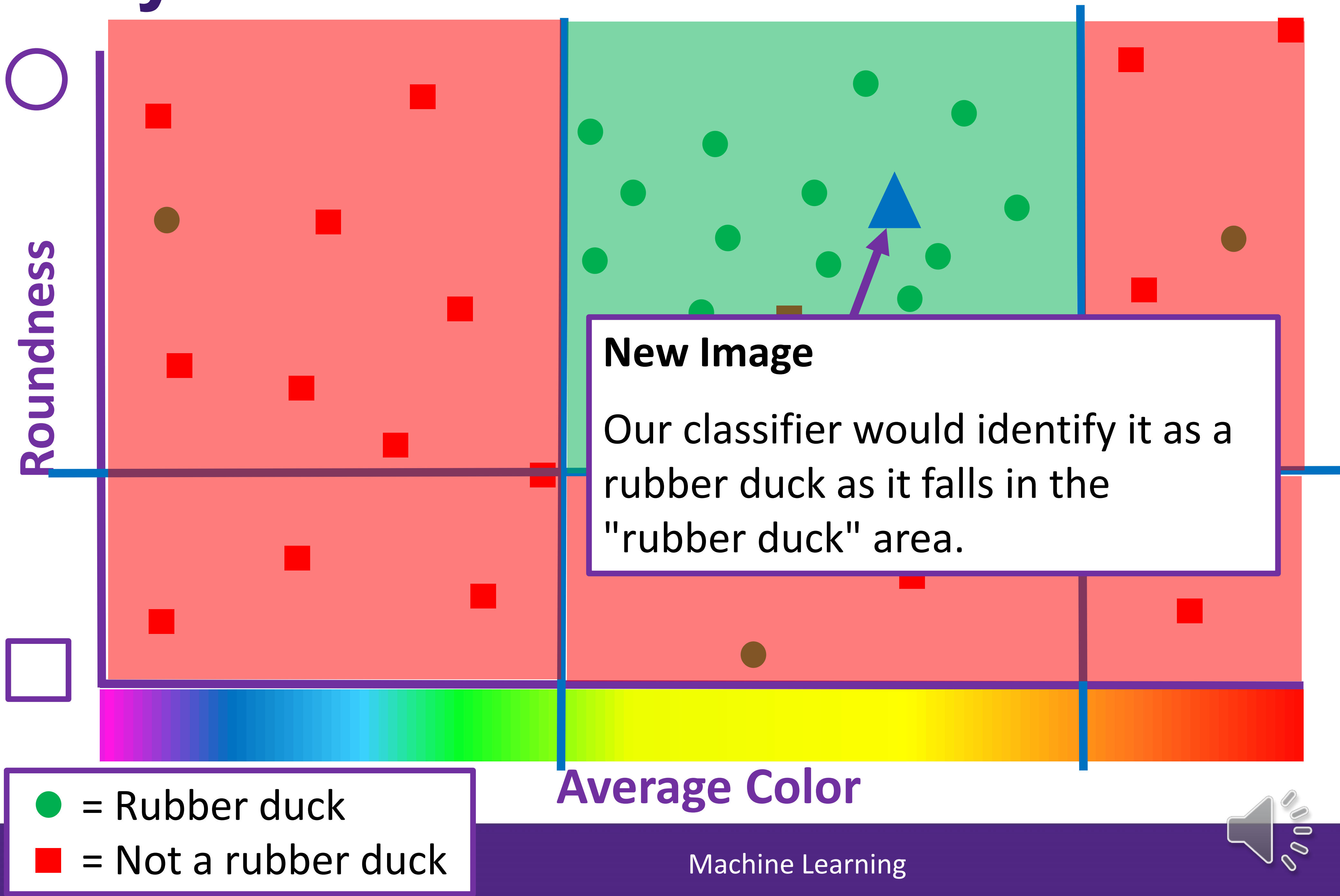
Toy Problem: Linear Classifier



Confusion Matrix

		Predicted	
		Rubber Duck	Not a Rubber Duck
Actual	Rubber Duck	20	3
	Not a Rubber Duck	1	18

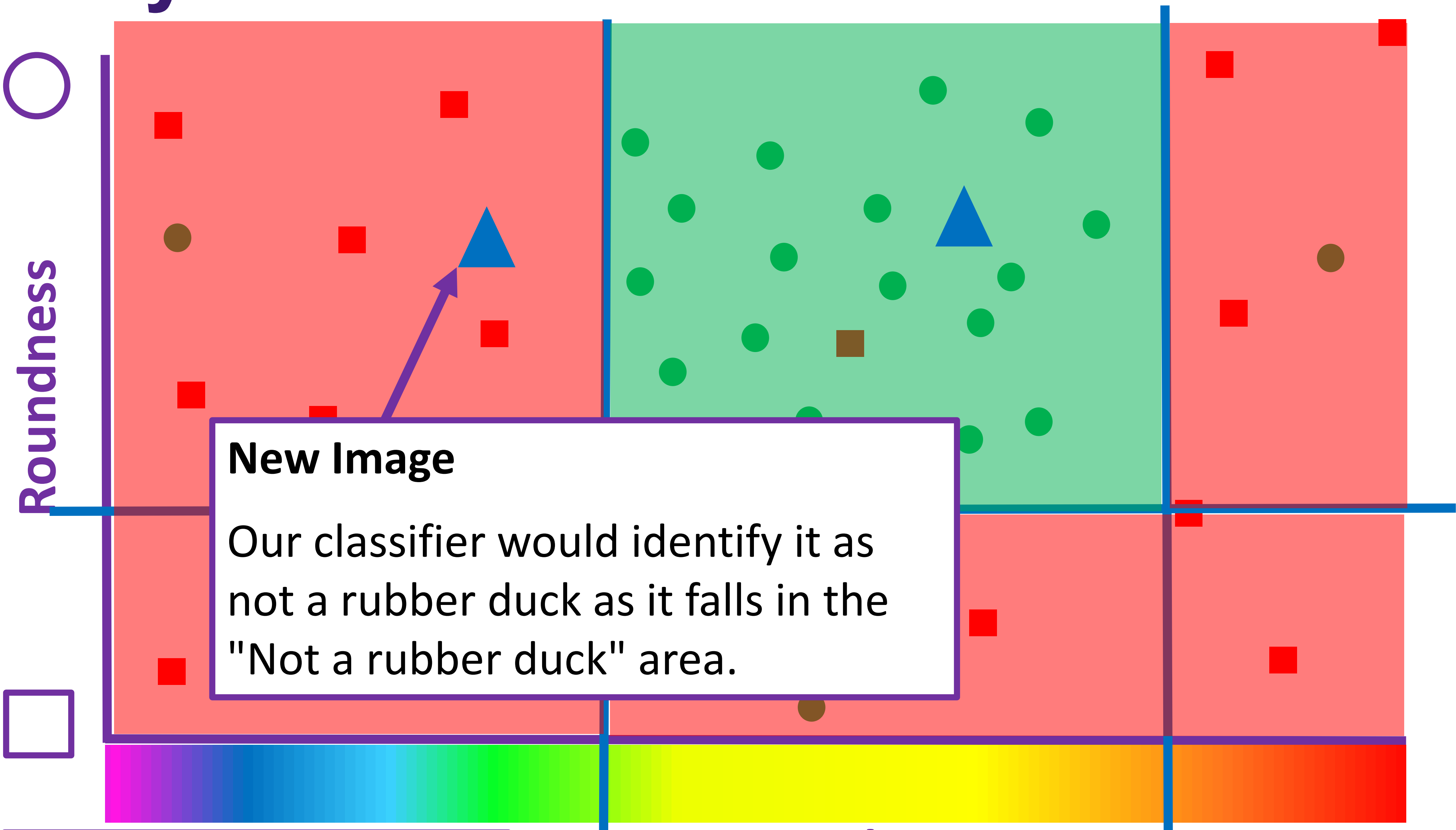
Toy Problem: Linear Classifier



New Image

Our classifier would identify it as a rubber duck as it falls in the "rubber duck" area.

Toy Problem: Linear Classifier

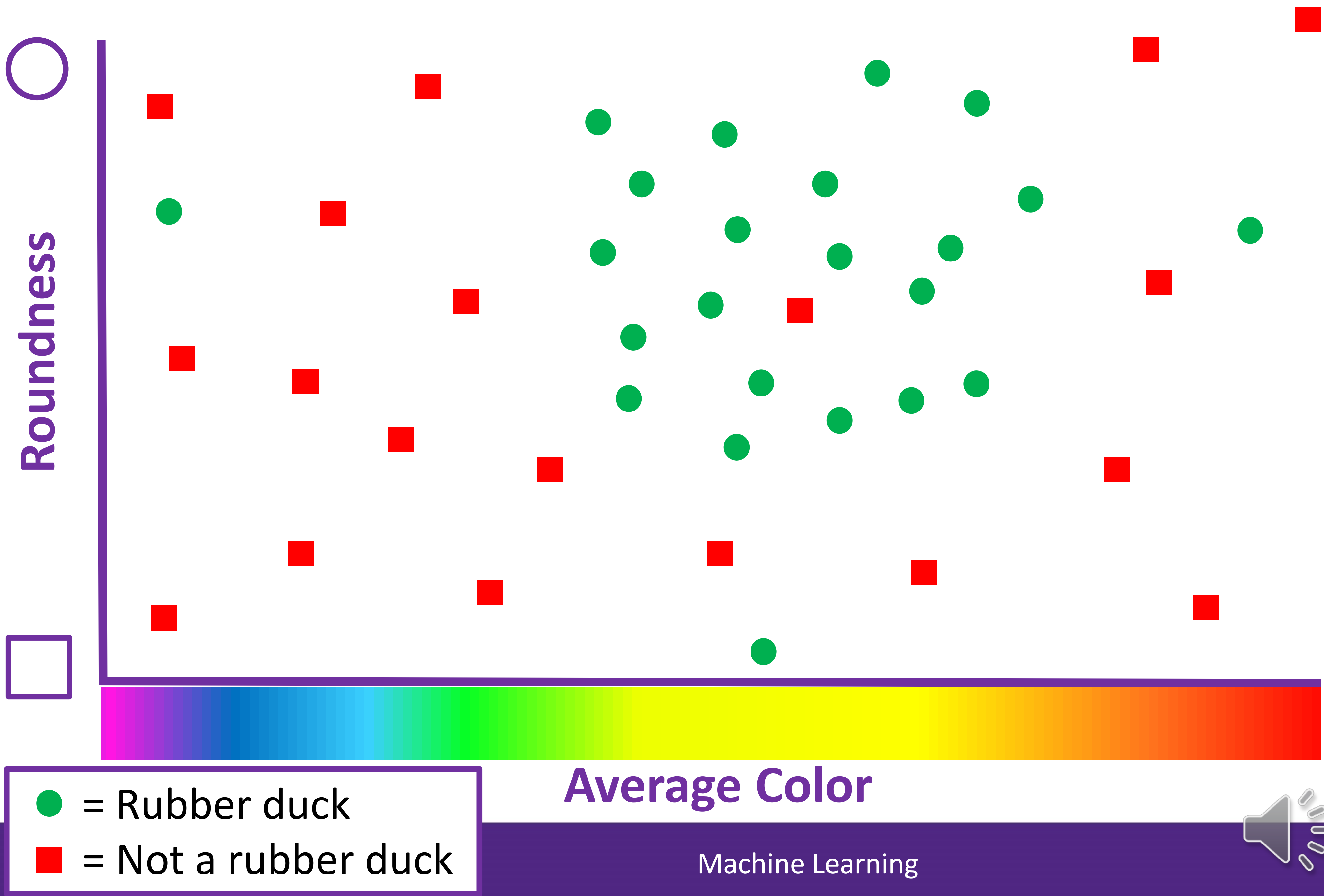


New Image

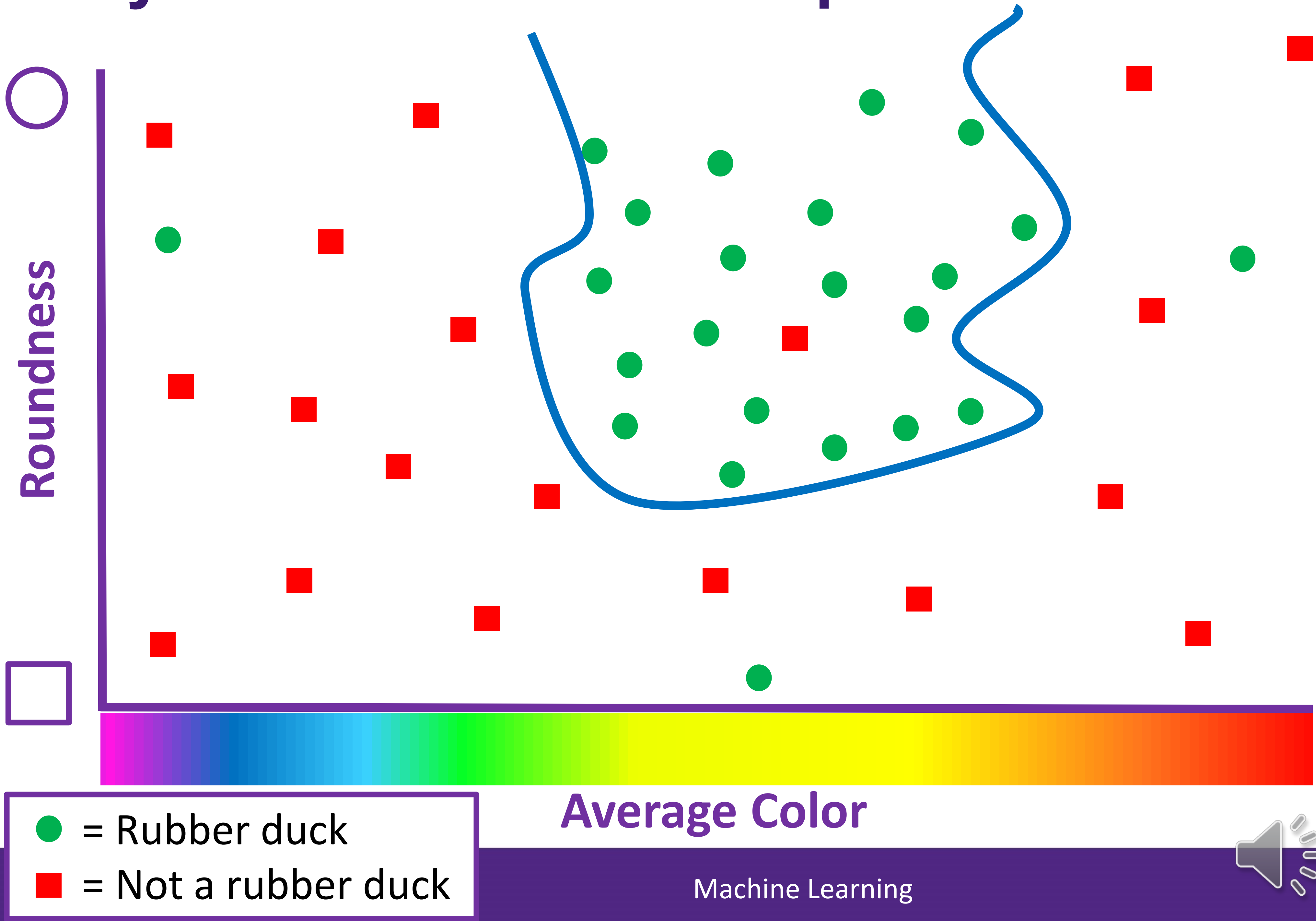
Our classifier would identify it as not a rubber duck as it falls in the "Not a rubber duck" area.

● = Rubber duck
■ = Not a rubber duck

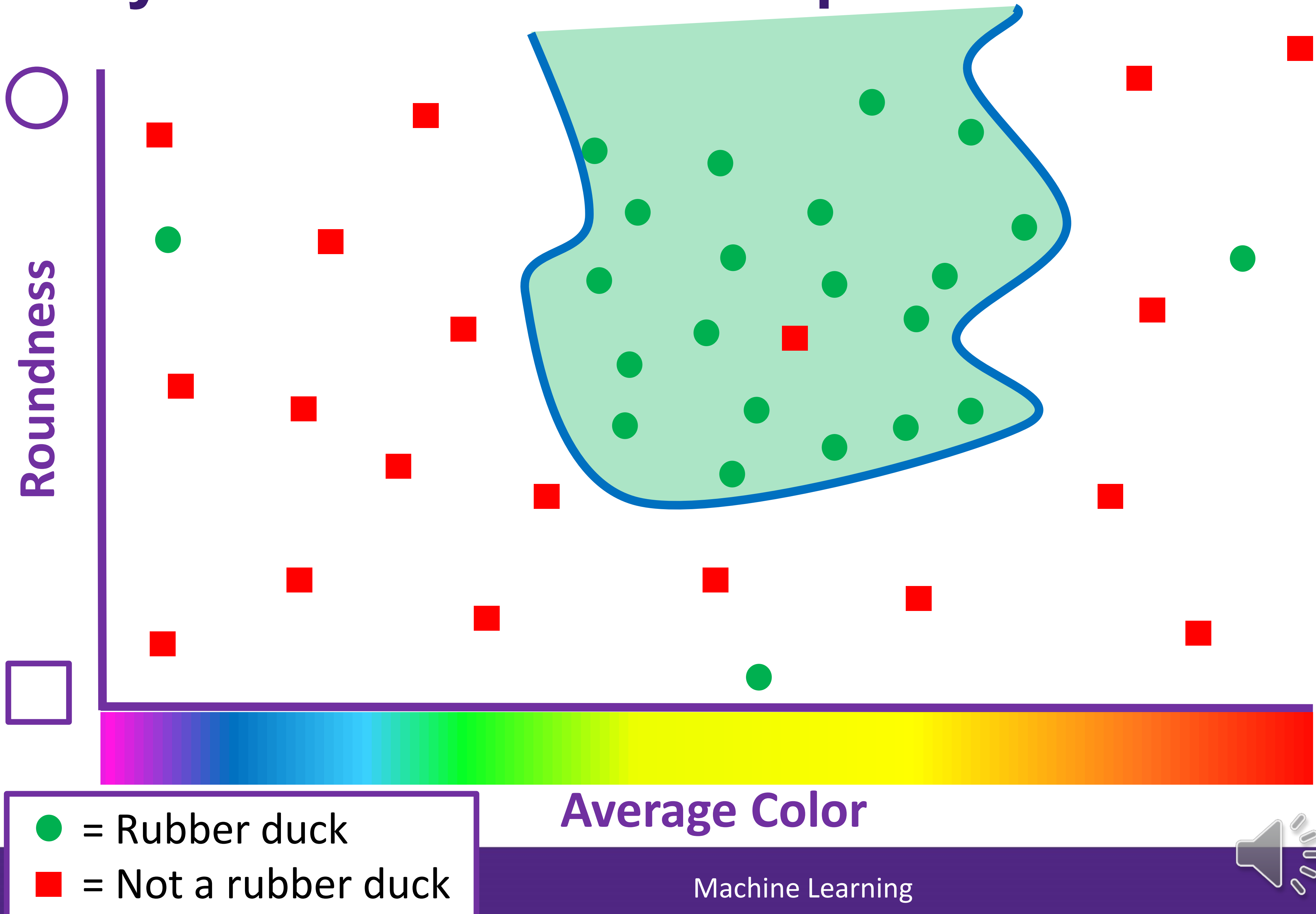
Toy Problem: More Complex Classifiers



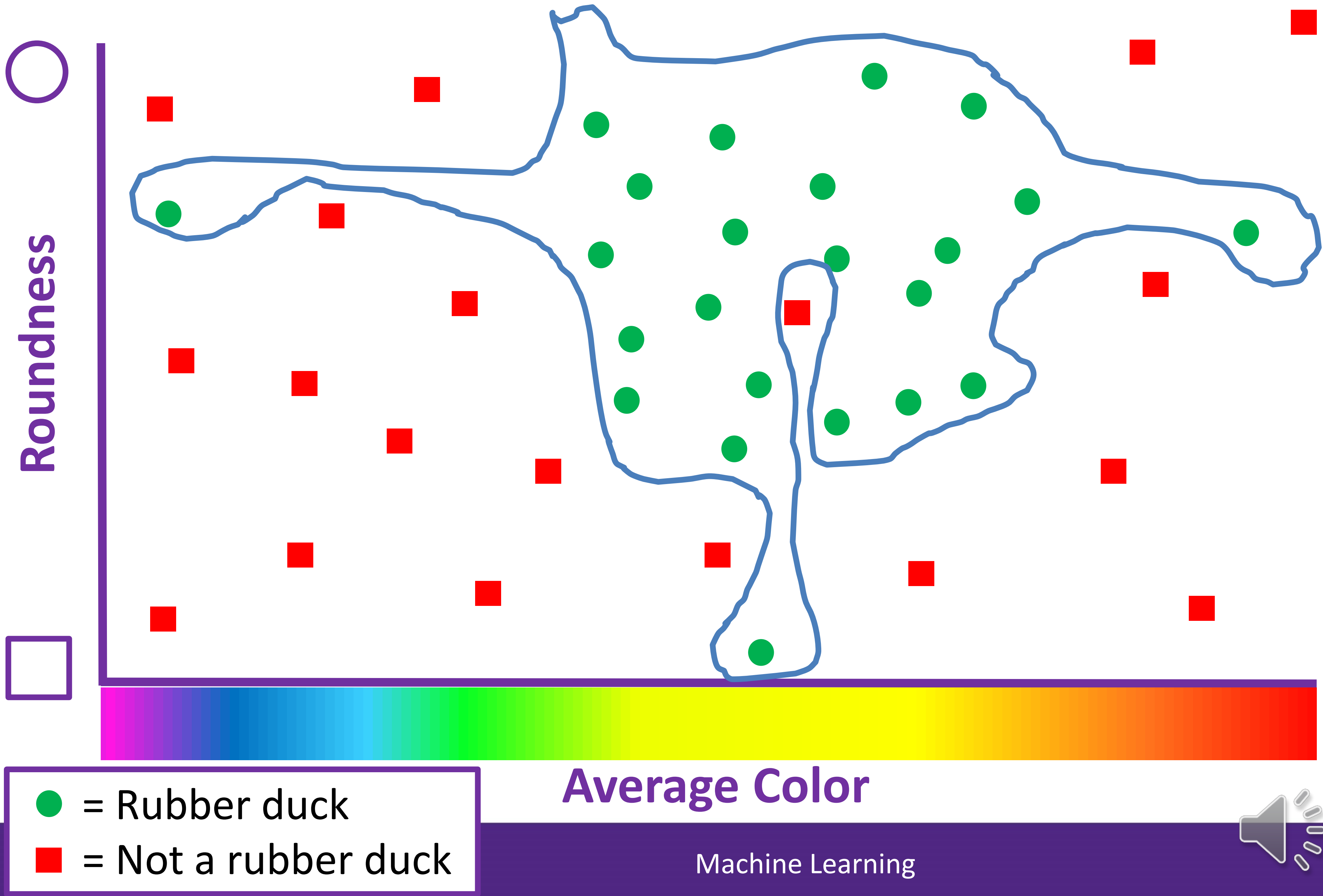
Toy Problem: More Complex Classifiers



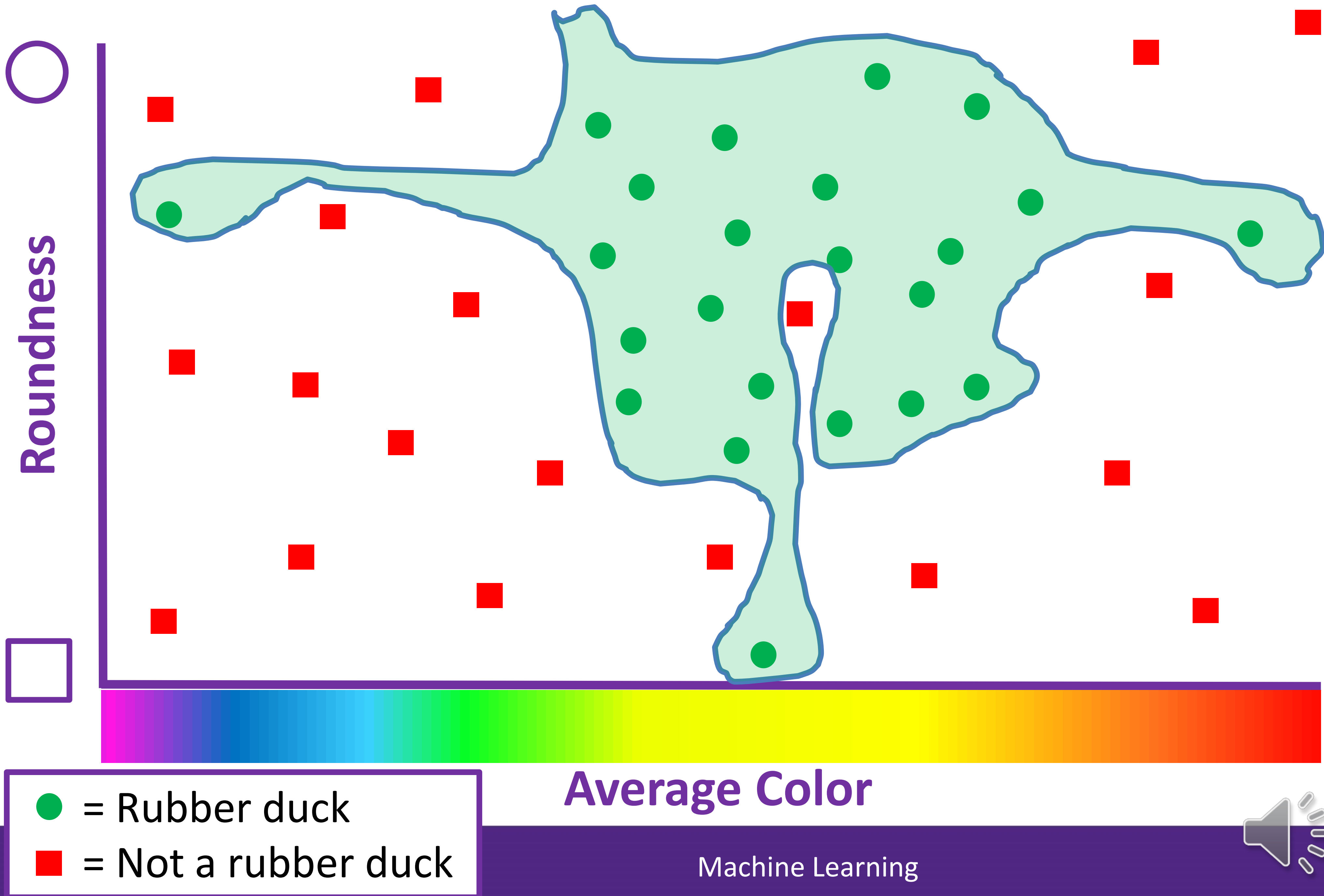
Toy Problem: More Complex Classifiers



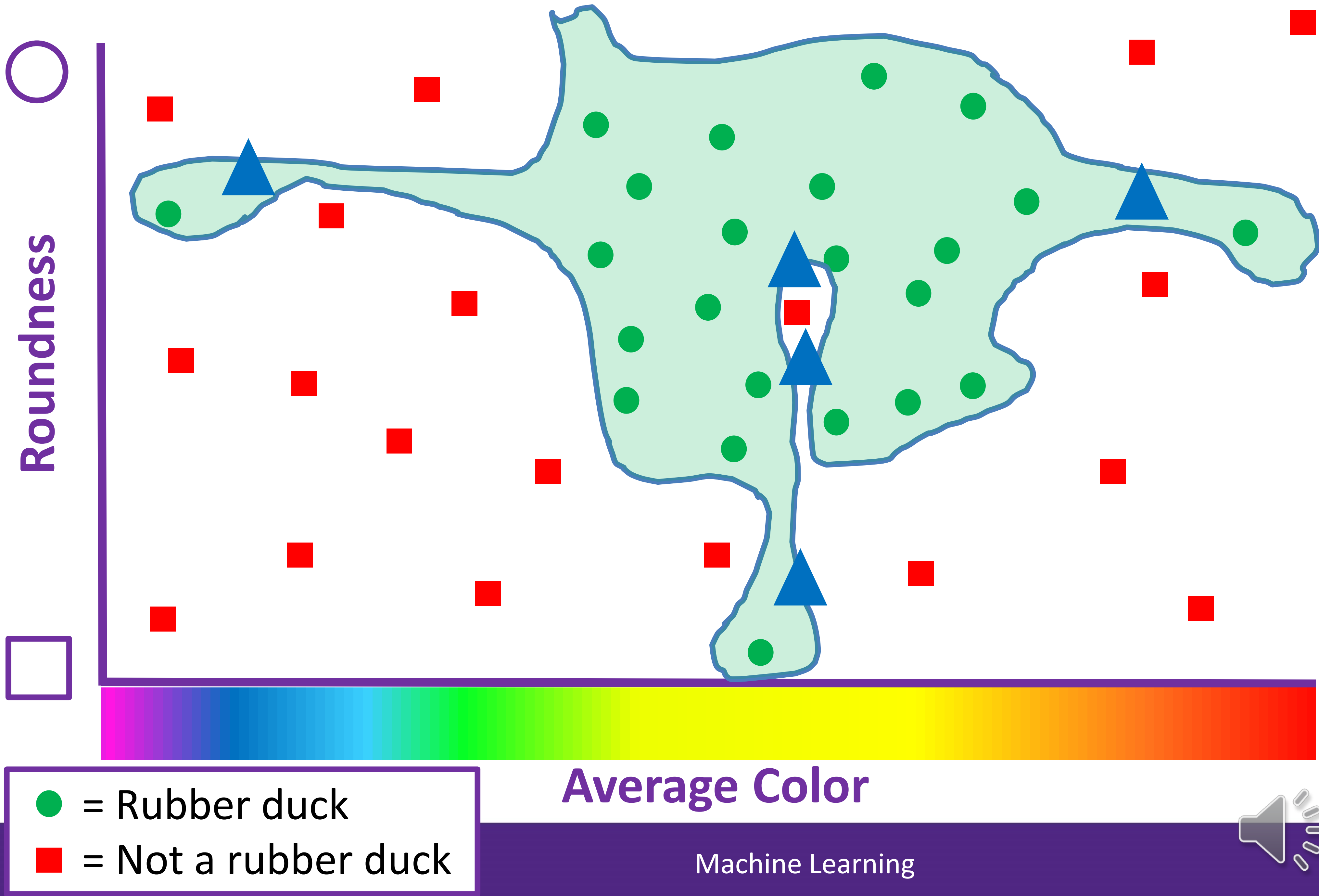
Toy Problem: Overfitting



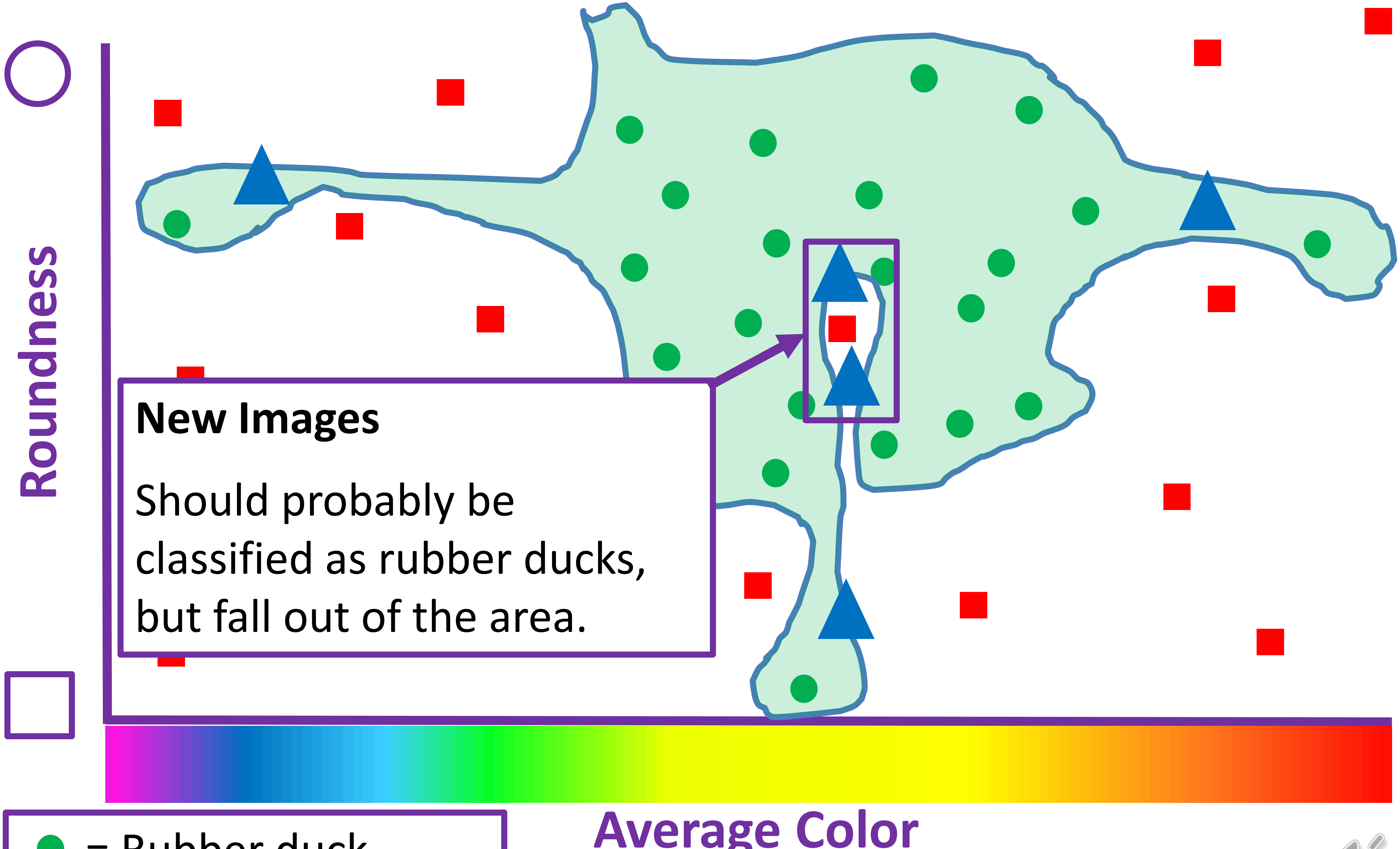
Toy Problem: Overfitting



Toy Problem: Overfitting

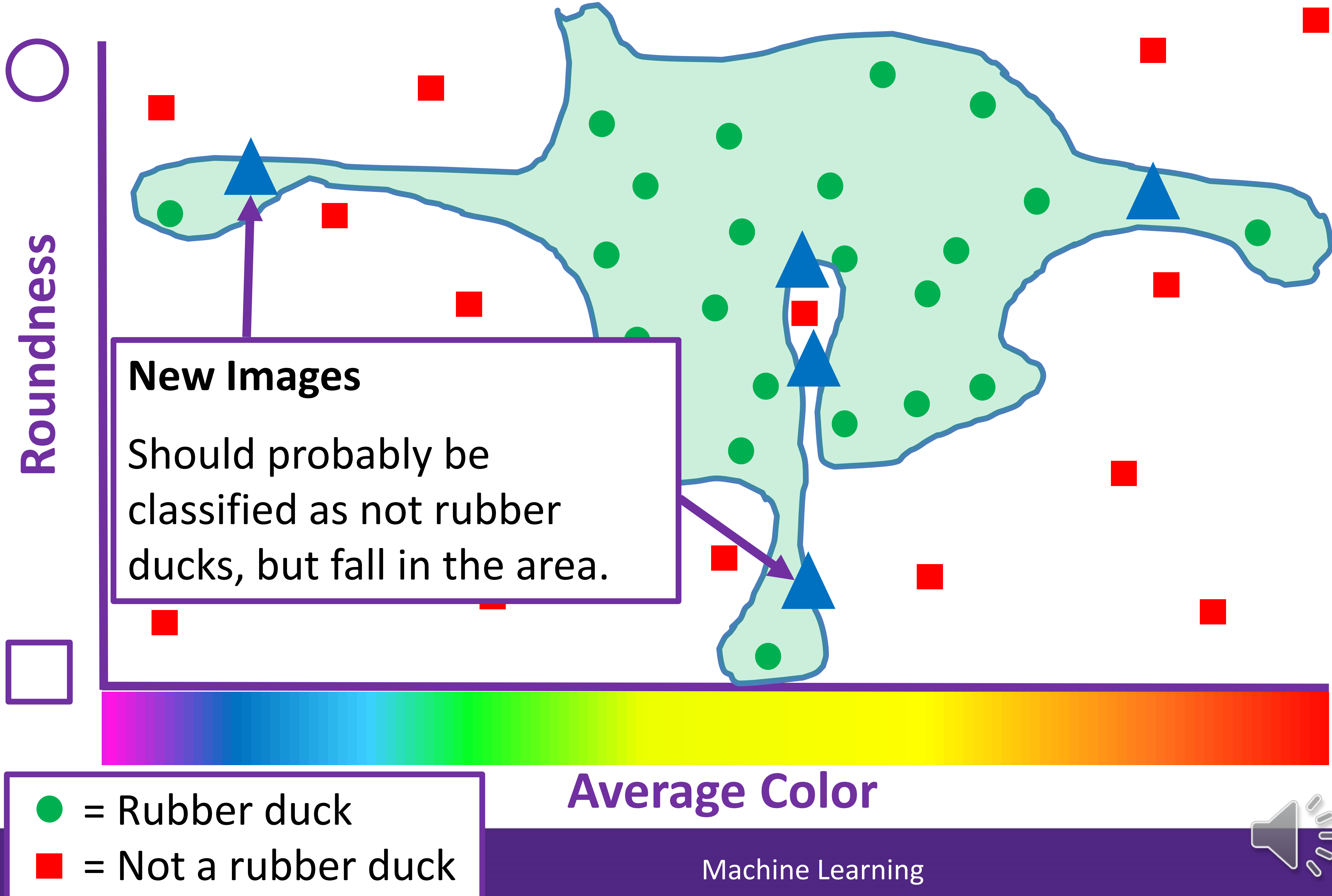


Toy Problem: Overfitting



● = Rubber duck
■ = Not a rubber duck

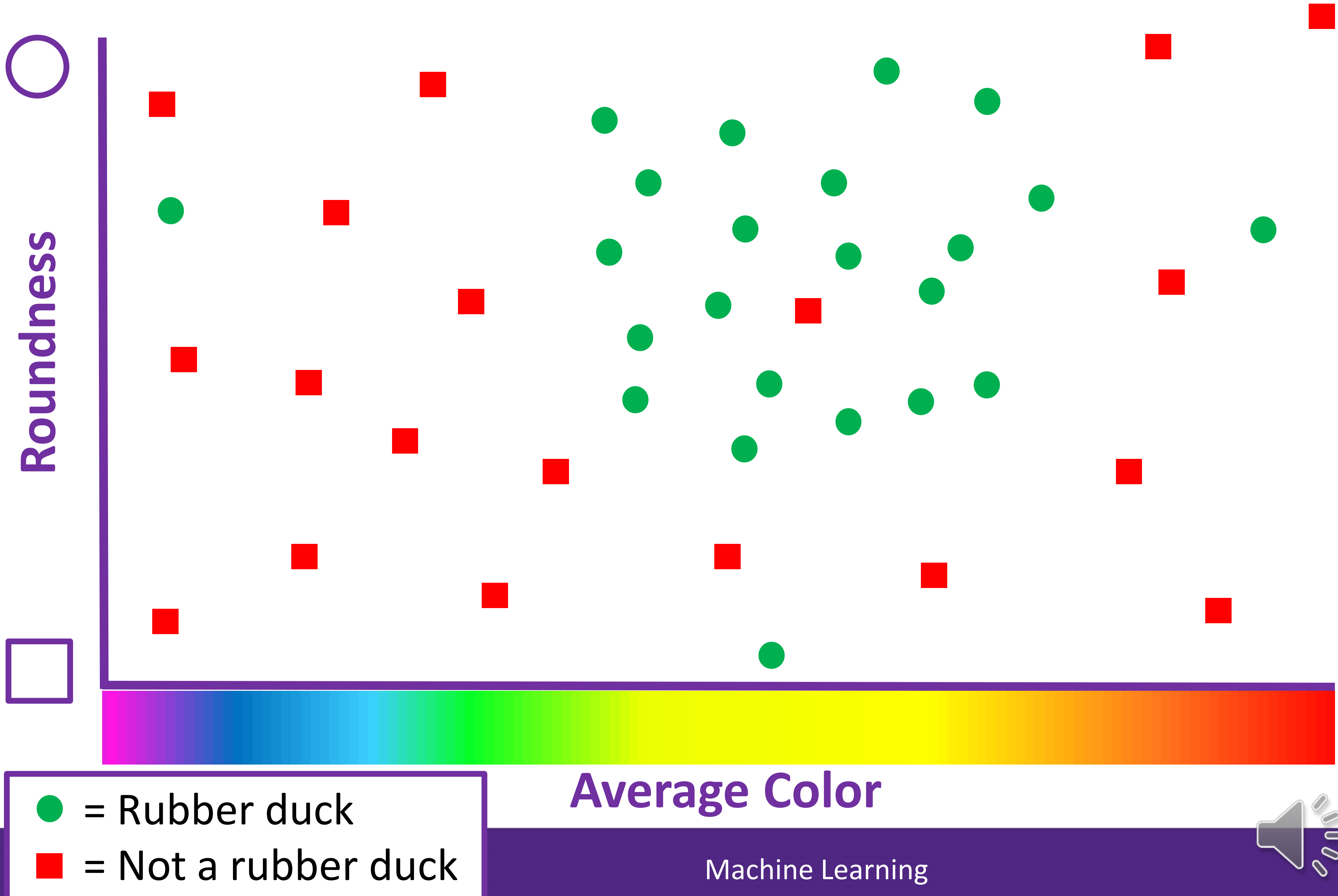
Toy Problem: Overfitting



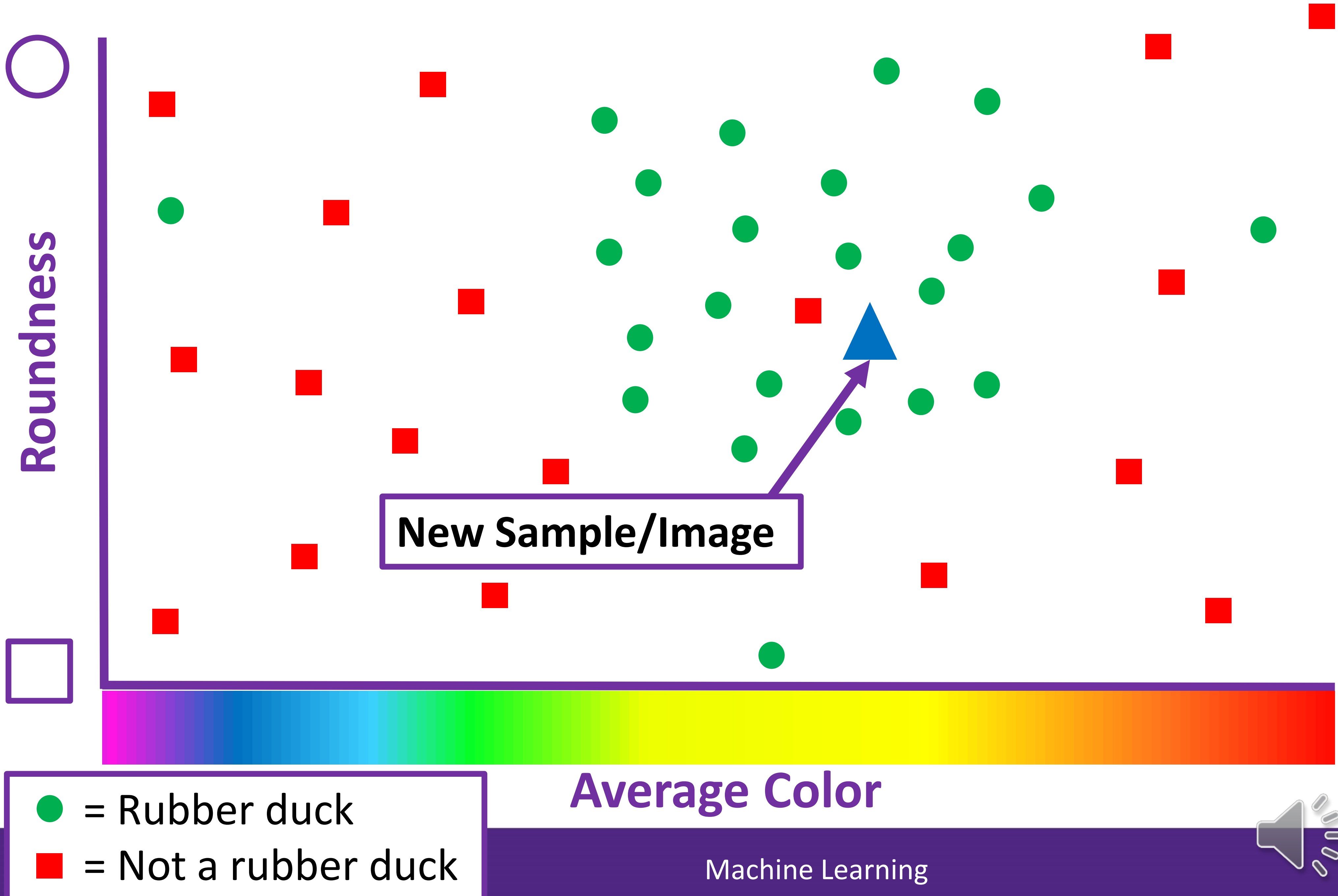
k-Nearest Neighbours (k-NN)

- Another classification method that works when there is a notion of distance in the feature space.
- Bases classification on the k closest neighbors in the training samples to a new sample in the feature space.
- Computationally simple and efficient.
  - Easy to implement
  - Fast results
  - No training time
- Lazy learning
  - Can be powerful in some cases but has limitations
  - Irrelevant features degrade results
  - Classes with more samples dominate

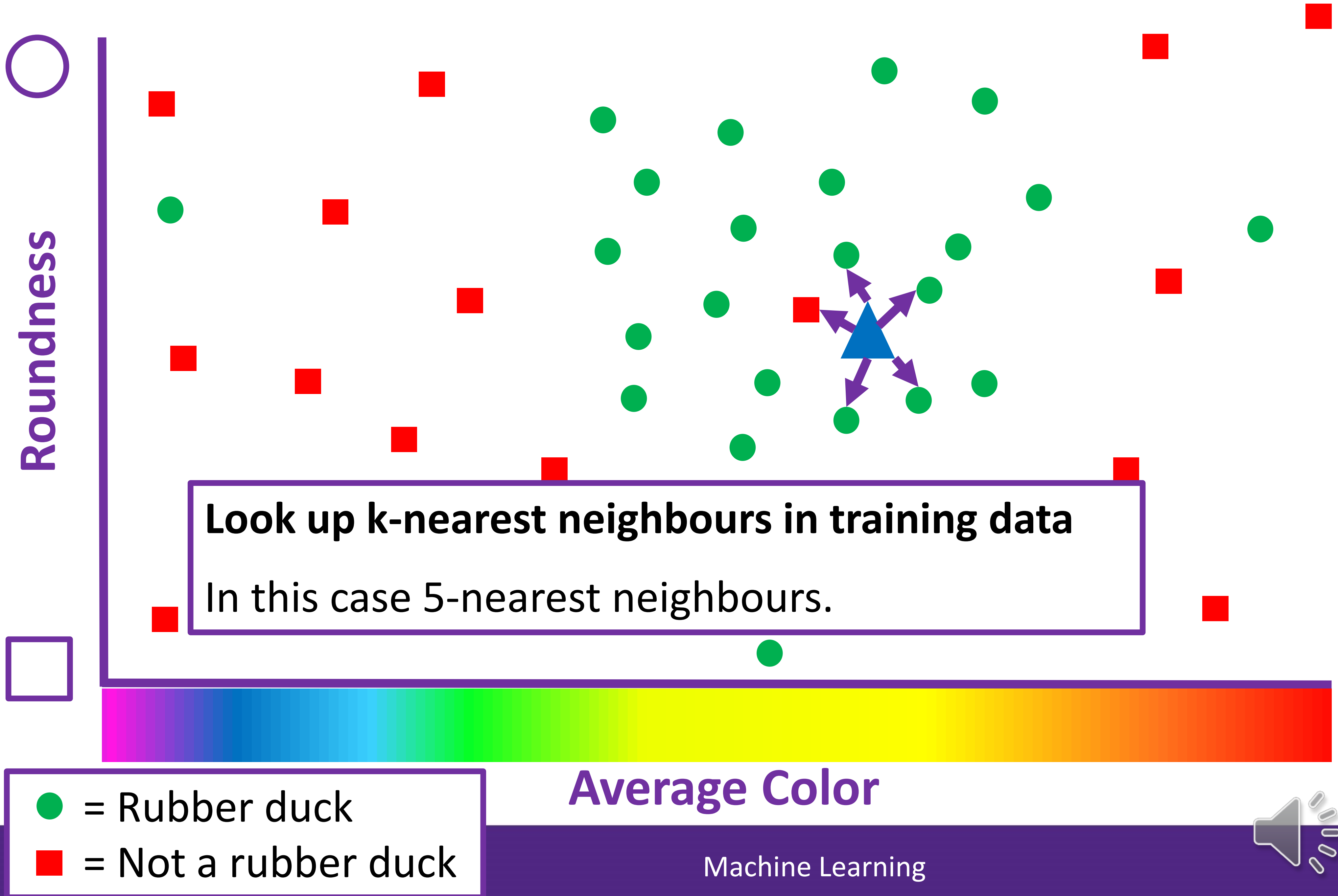
k-Nearest Neighbours (k-NN)



k-Nearest Neighbours (k-NN)



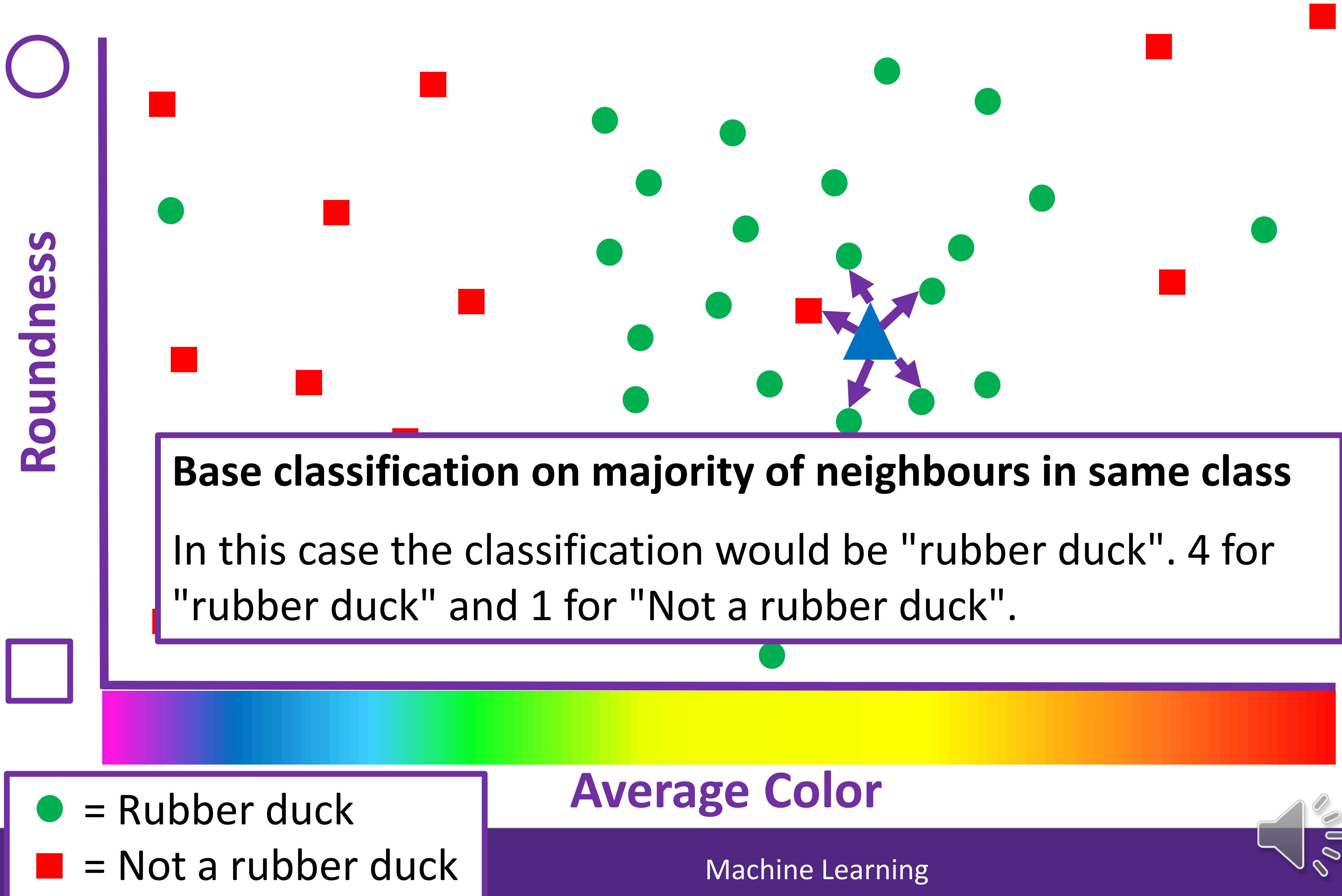
k-Nearest Neighbours (k-NN)



Look up k-nearest neighbours in training data

In this case 5-nearest neighbours.

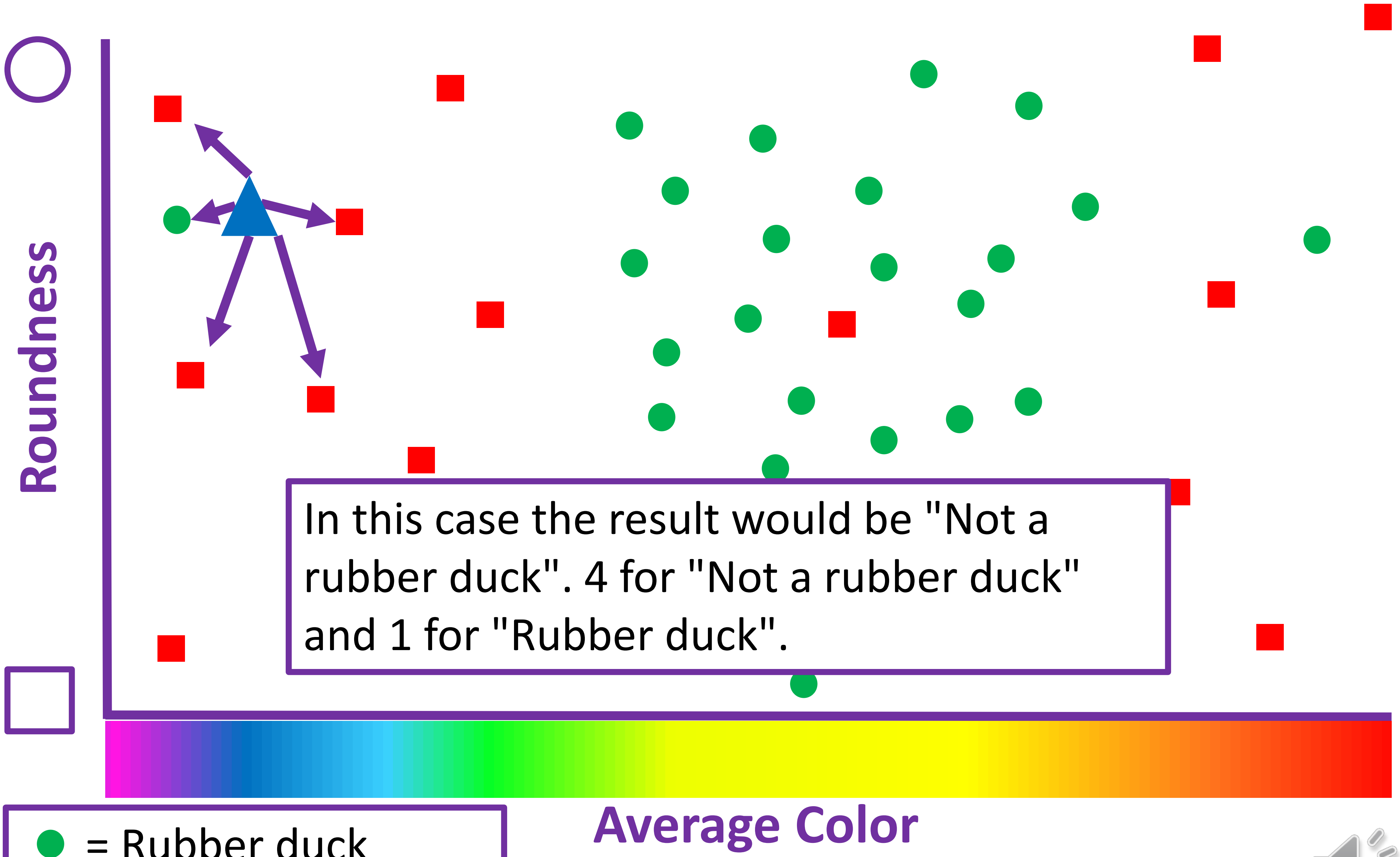
k-Nearest Neighbours (k-NN)



Base classification on majority of neighbours in same class

In this case the classification would be "rubber duck". 4 for "rubber duck" and 1 for "Not a rubber duck".

k-Nearest Neighbours (k-NN)



In this case the result would be "Not a rubber duck". 4 for "Not a rubber duck" and 1 for "Rubber duck".

● = Rubber duck
■ = Not a rubber duck

Decision Tree Learning

- Widely used method for getting a general idea from specific examples.
- Robust to noisy data.
- **Method:**
  - Find the unused feature that gives the most information and consider that feature first.
  - Repeat until done.
- **Concepts:**
  - Information theory
  - Entropy
  - Information gain

Decision Tree Learning

- Widely used method for getting a general idea from specific examples.
- Robust to noisy data.
- **Method:**
  - Find the unused feature that gives the most information and consider that feature first.
  - Repeat until done.
- **Concepts:**
  - Information theory
  - Entropy
  - Information gain

Building a Decision Tree

- Will use the concepts of **entropy** and **information gain**.

- **Entropy** is:
  - A measure of how mixed a collection is.
  - A measure of impurity.

- **Information gain** measures how well a choice separates examples.
  - Measures change in entropy.

Decision Tree Learning: Example

(from "Machine Learning" by Tom Mitchell, McGraw Hill 1997 and notes by J. Forgette)

- Given this information:
  - Outlook is sunny
  - Temperature is hot
  - Humidity is high
  - Wind is strong
- Should we play tennis?
  - Yes
  - No

Decision Tree Learning: Example

(from "Machine Learning" by Tom Mitchell, McGraw Hill 1997 and notes by J. Forgette)

- Given this information:
  - Outlook is sunny
  - Temperature is hot
  - Humidity is high
  - Wind is strong
- Should we play tennis?
  - Yes
  - No

Features (aka Attributes)

- Outlook
- Temperature
- Humidity
- Wind

Decision Tree Learning: Example

(from "Machine Learning" by Tom Mitchell, McGraw Hill 1997 and notes by J. Forgette)

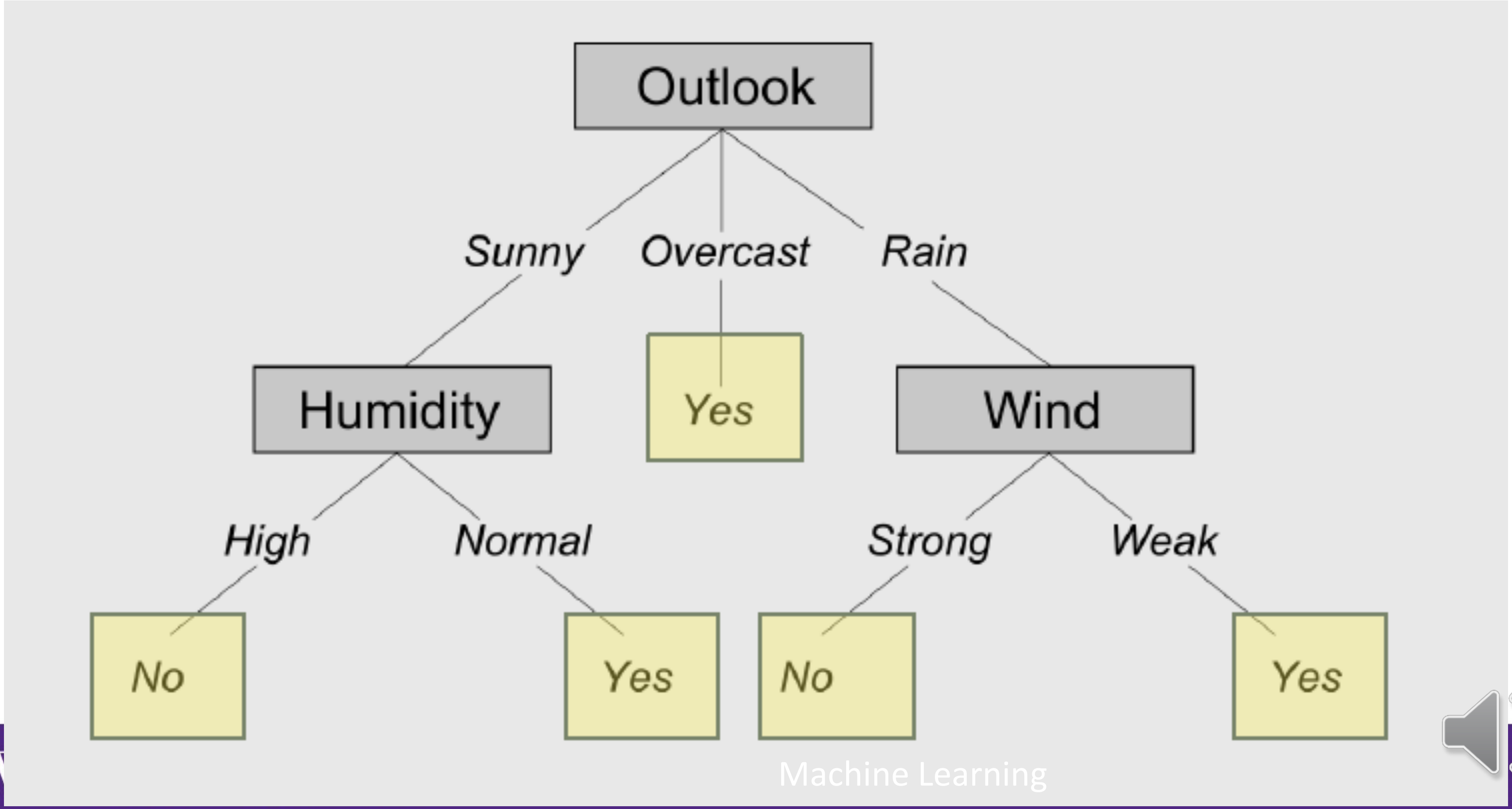
- Given this information:
  - Outlook is sunny
  - Temperature is hot
  - Humidity is high
  - Wind is strong

- Should we play tennis?
  - Yes
  - No

Classes
Yes or No

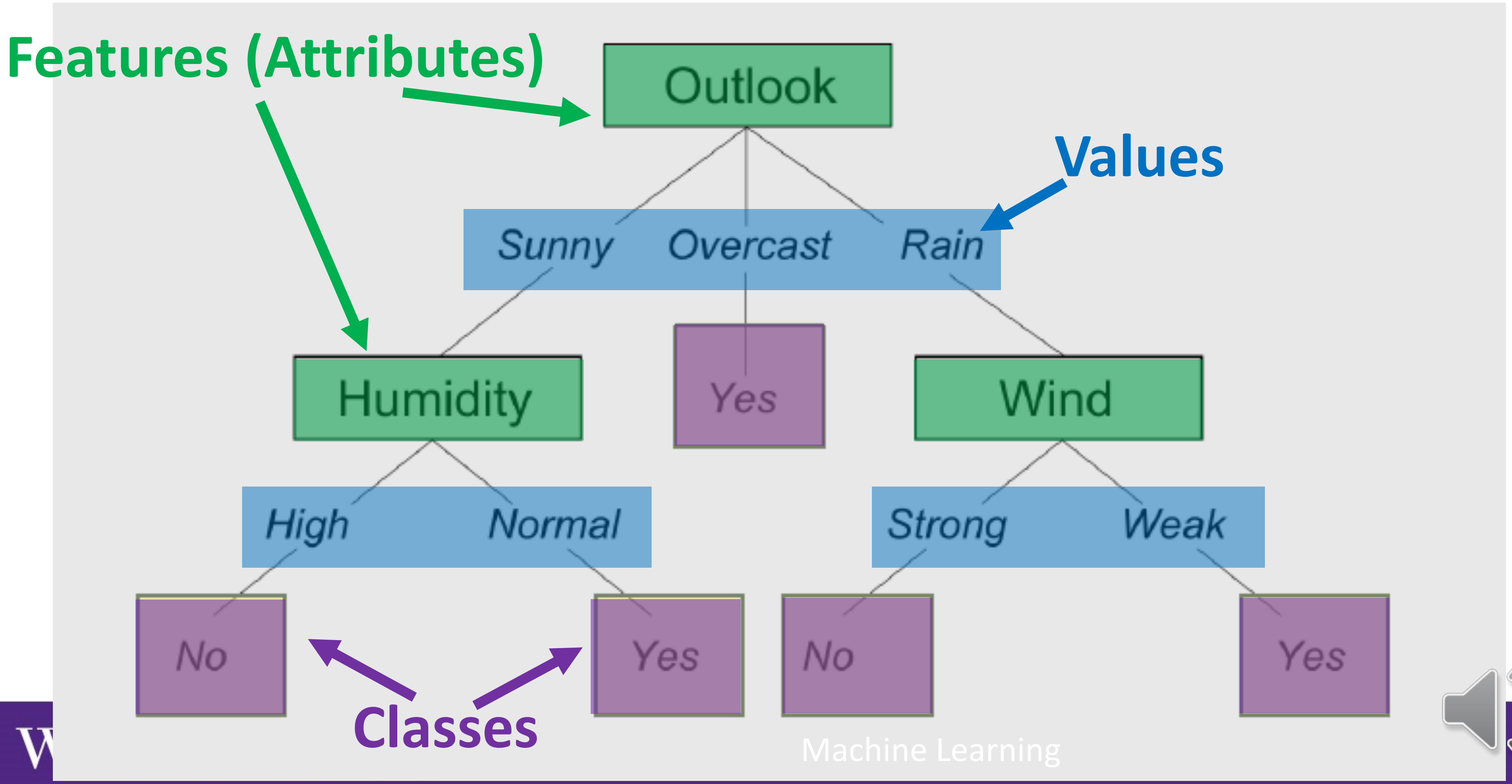
Decision Tree Learning: Example

Goal: Create a decision tree that classifies a set of weather related features (Outlook, Temp, Wind, Humidity) into two classes, play tennis (yes) or do not play tennis (no).



Decision Tree Learning: Example

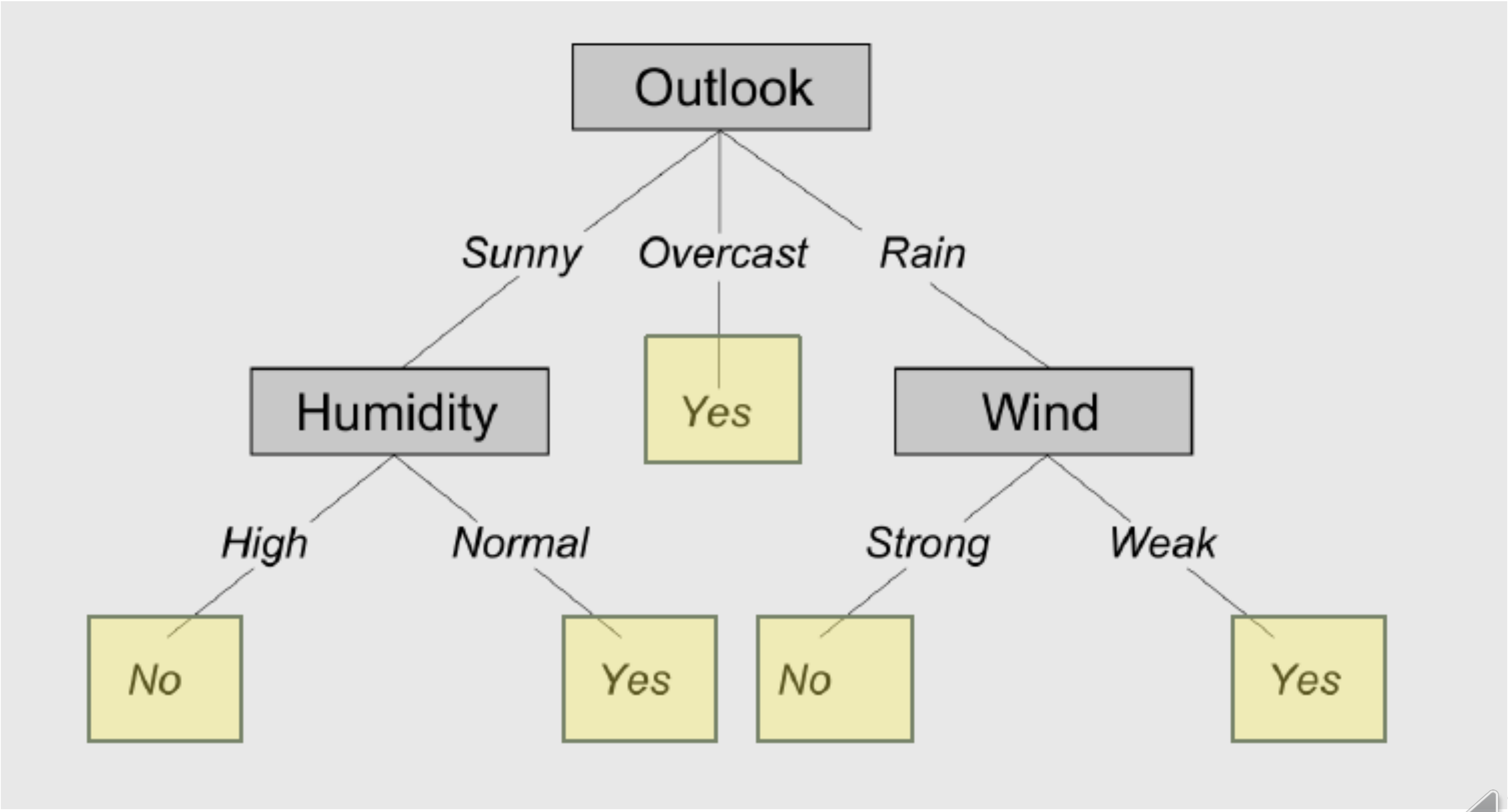
Goal: Create a decision tree that classifies a set of weather related features (Outlook, Temp, Wind, Humidity) into two classes, play tennis (yes) or do not play tennis (no).



Decision Tree Learning: Example

Example Input 1:

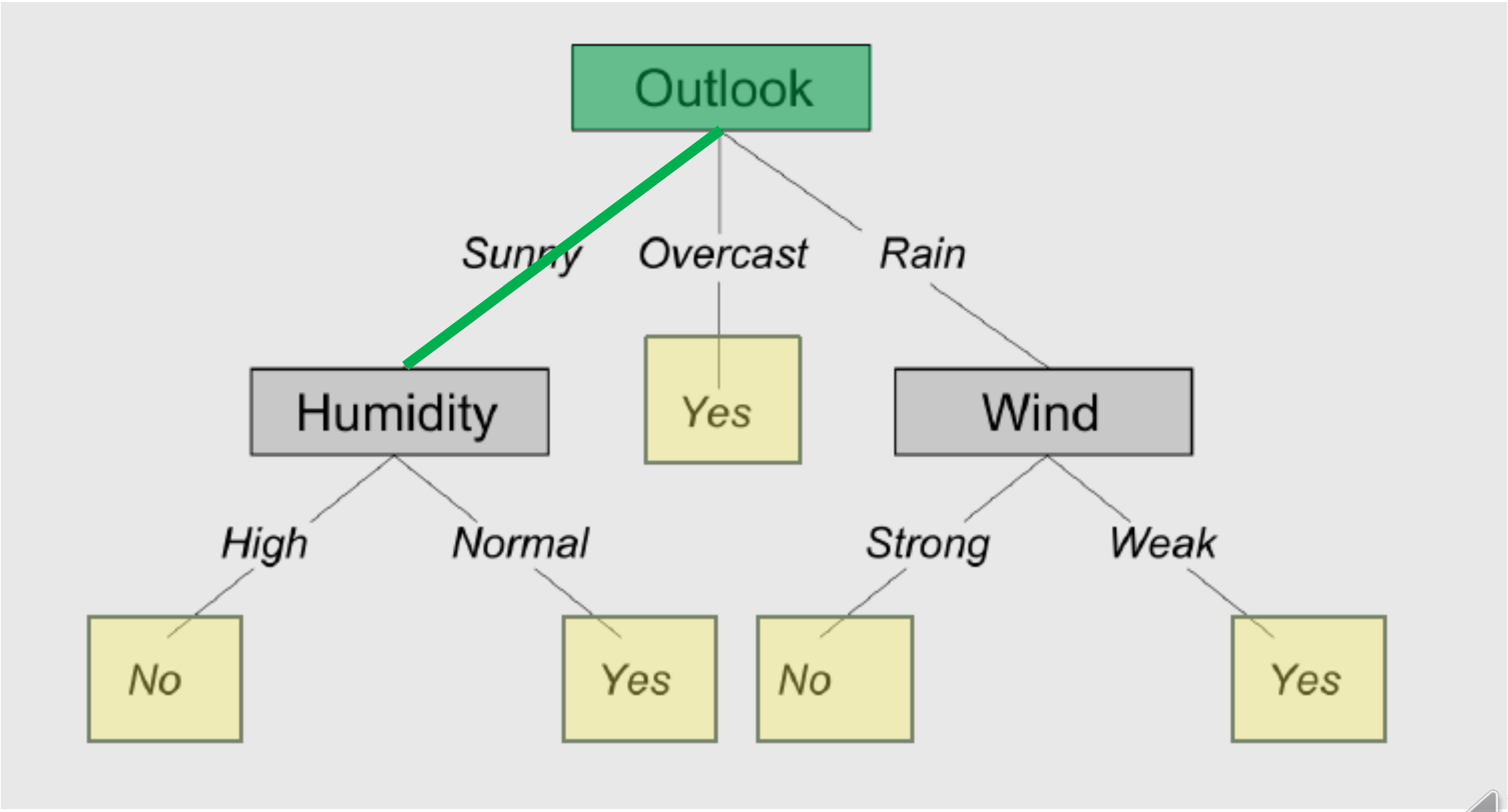
Outlook = Sunny, **Humidity** = Normal, **Wind** = Strong, **Temp** = Hot



Decision Tree Learning: Example

Example Input 1:

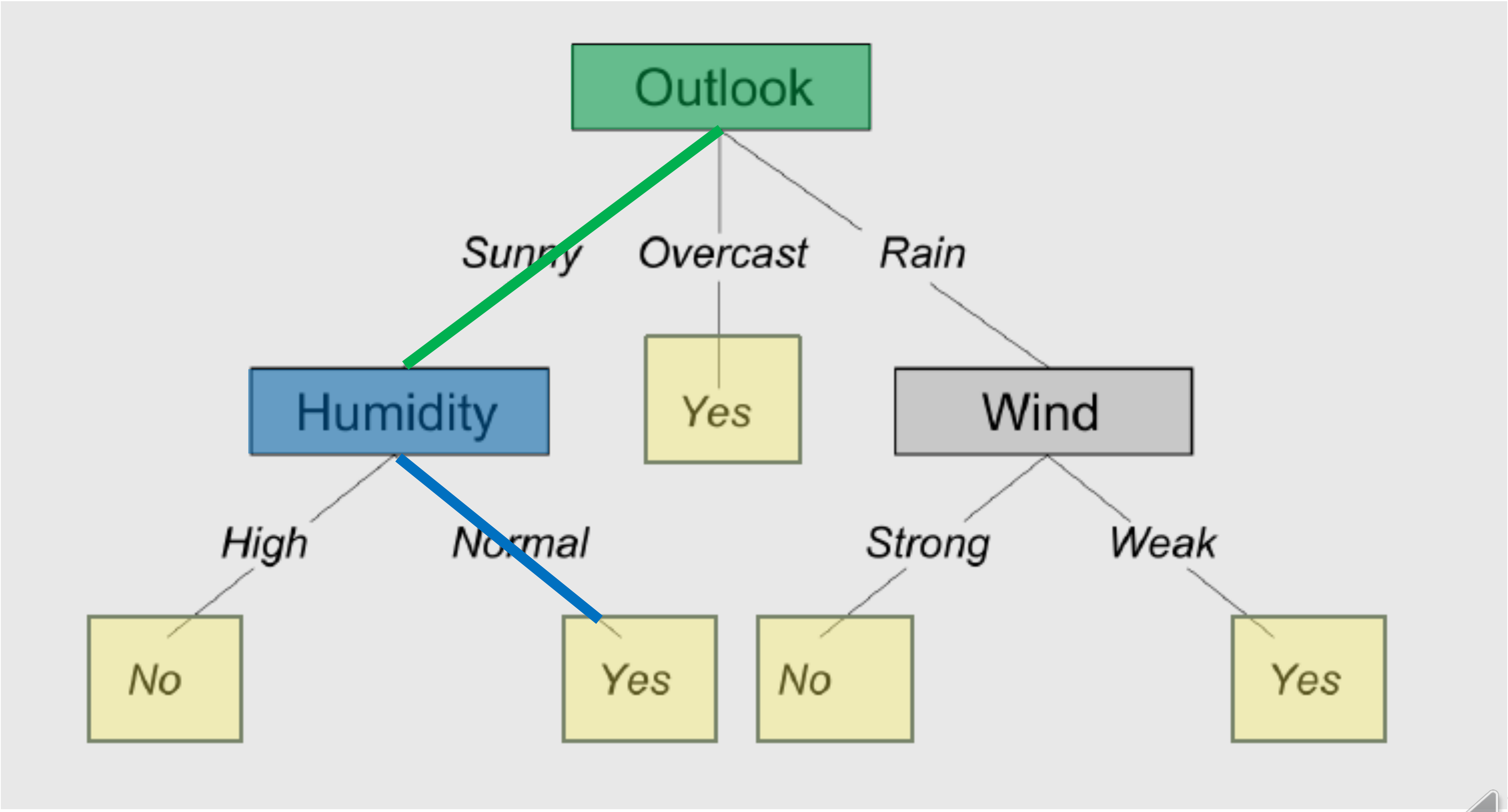
Outlook = Sunny, **Humidity** = Normal, **Wind** = Strong, **Temp** = Hot



Decision Tree Learning: Example

Example Input 1:

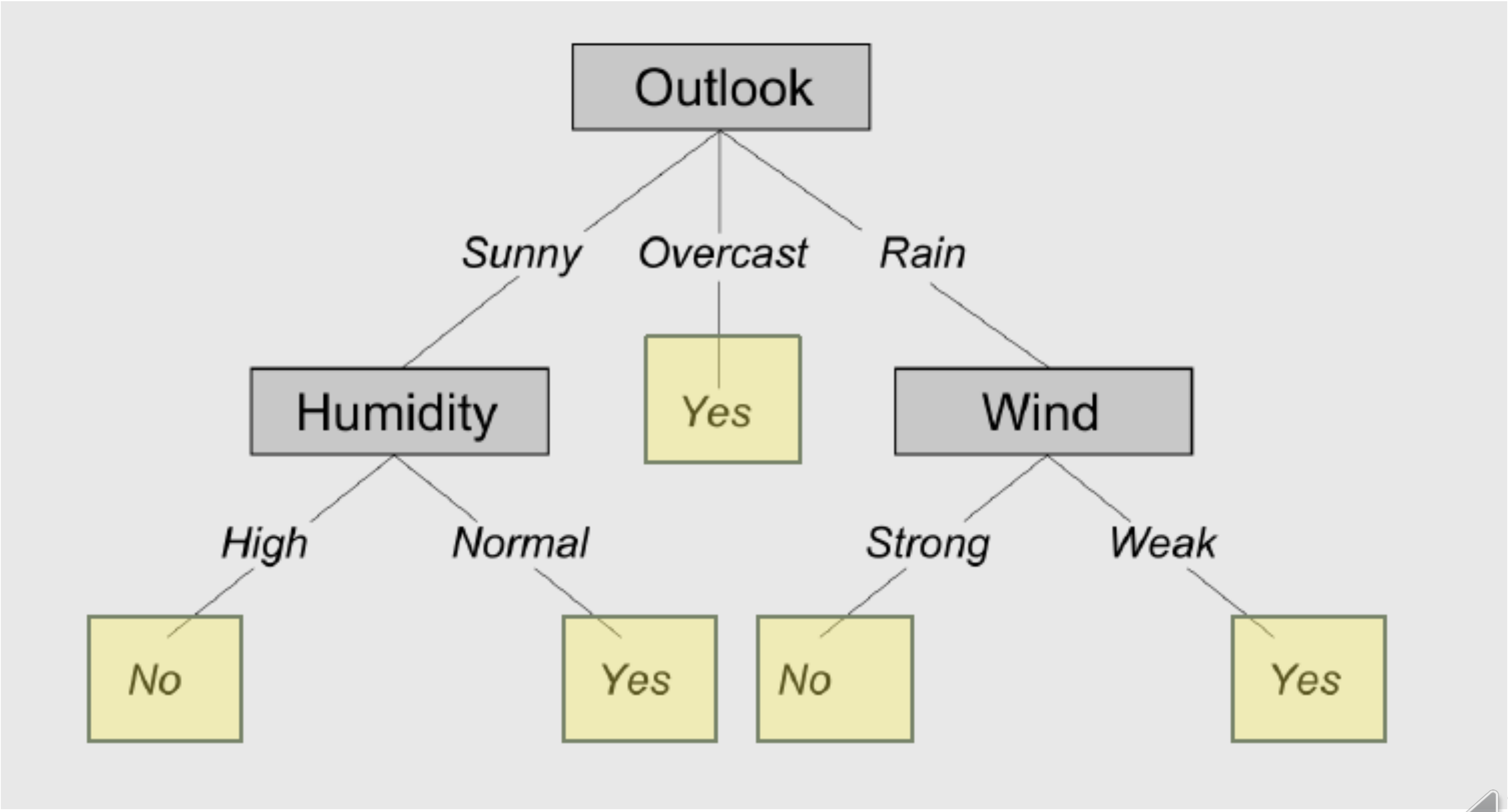
Outlook = Sunny, **Humidity** = Normal, **Wind** = Strong, **Temp** = Hot



Decision Tree Learning: Example

Example Input 2:

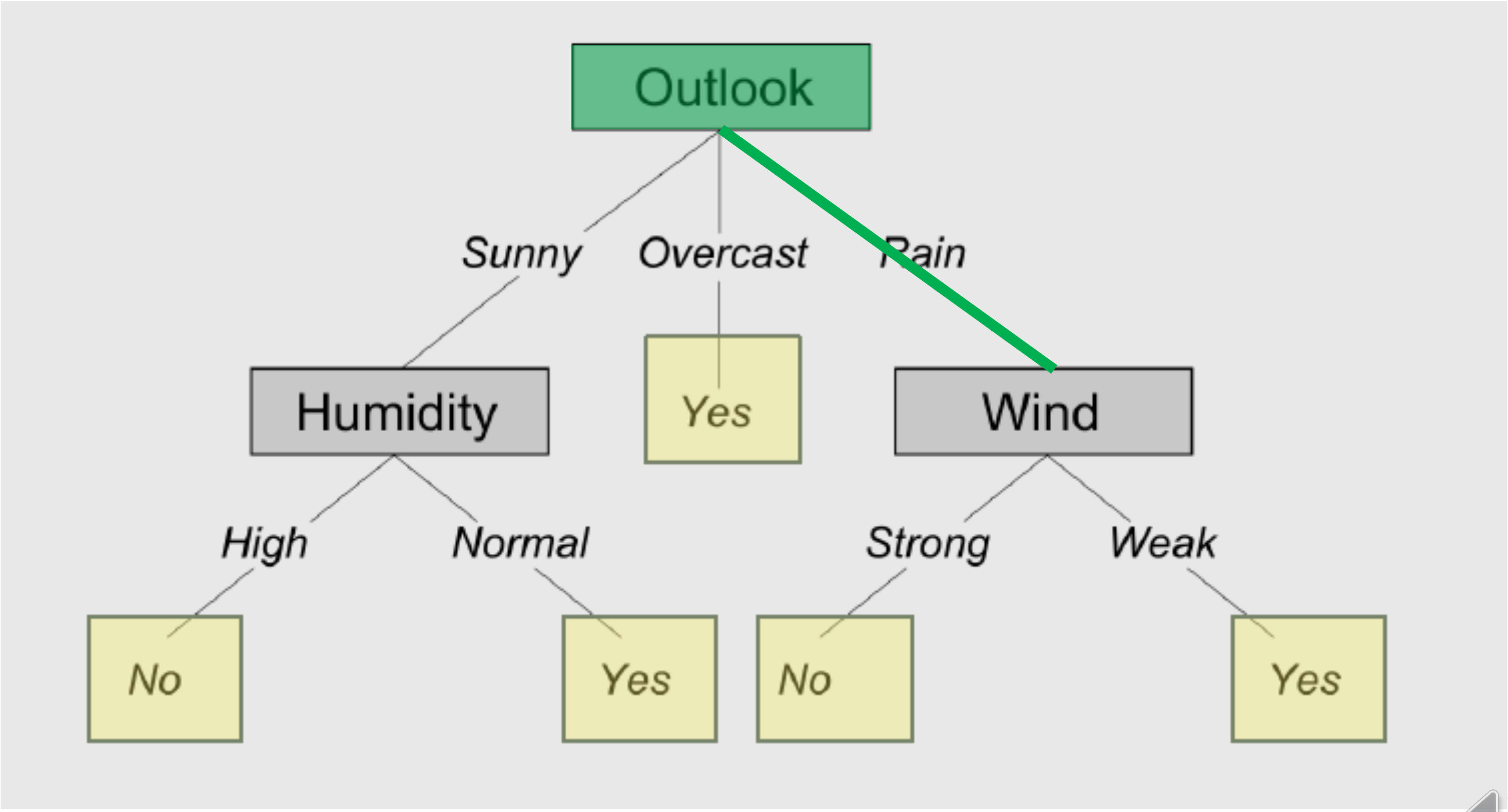
Outlook = Rain, **Humidity** = High, **Wind** = Strong, **Temp** = Cold



Decision Tree Learning: Example

Example Input 2:

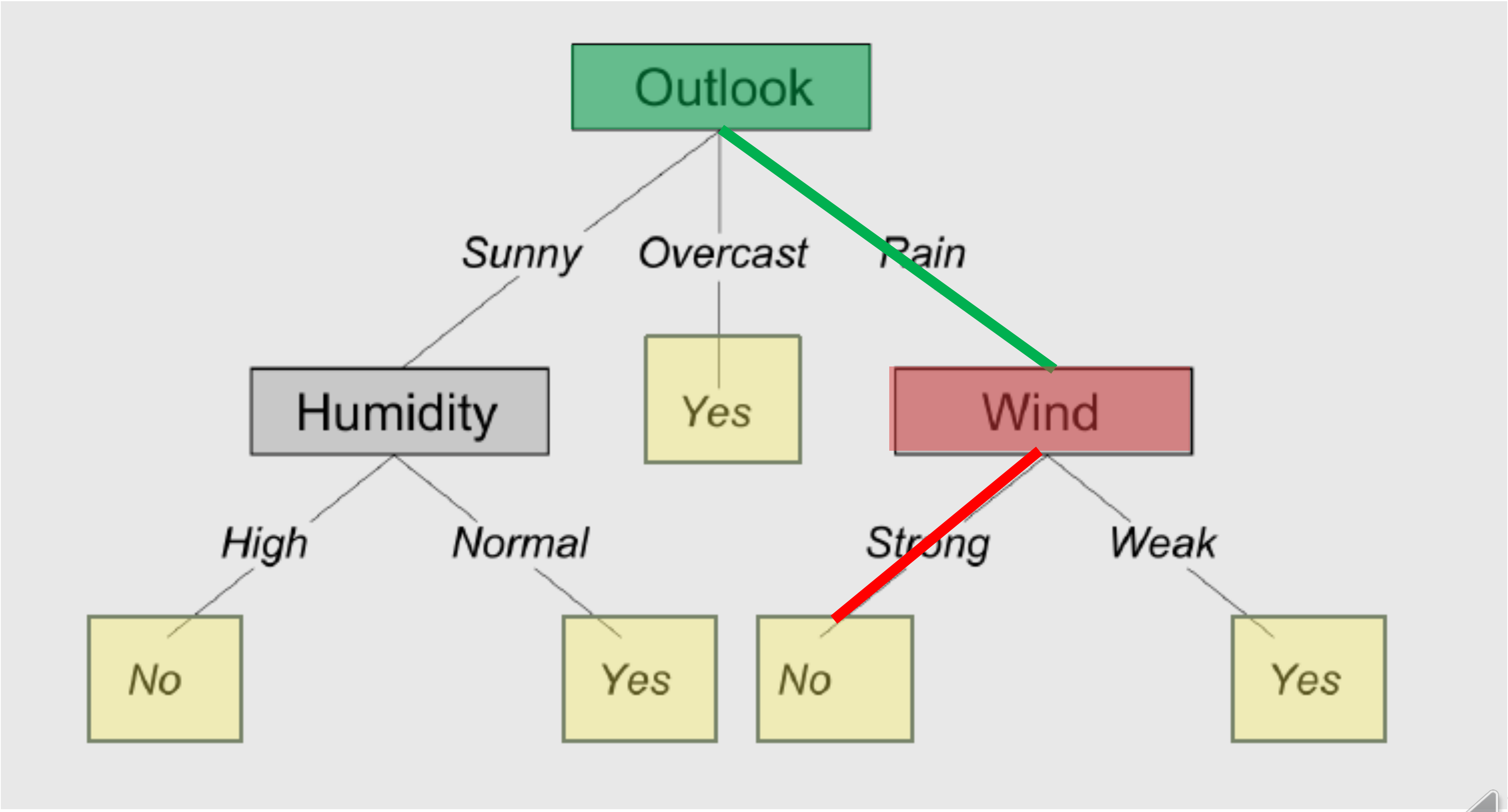
Outlook = Rain, **Humidity** = High, **Wind** = Strong, **Temp** = Cold



Decision Tree Learning: Example

Example Input 2:

Outlook = Rain, **Humidity** = High, **Wind** = Strong, **Temp** = Cold



Decision Trees

- Tree is constructed from training examples.
- Once tree is constructed, use it to classify new instances (that were not in the examples).
- The effectiveness of the tree is determined by the correctness of the classification of new instances.

Other Methods/Classifiers

- Many other methods/classifiers exist
  - Support Vector Machines (SVM)
  - Neural Networks
  - Cluster analysis
  - Genetic algorithms
  - Etc.