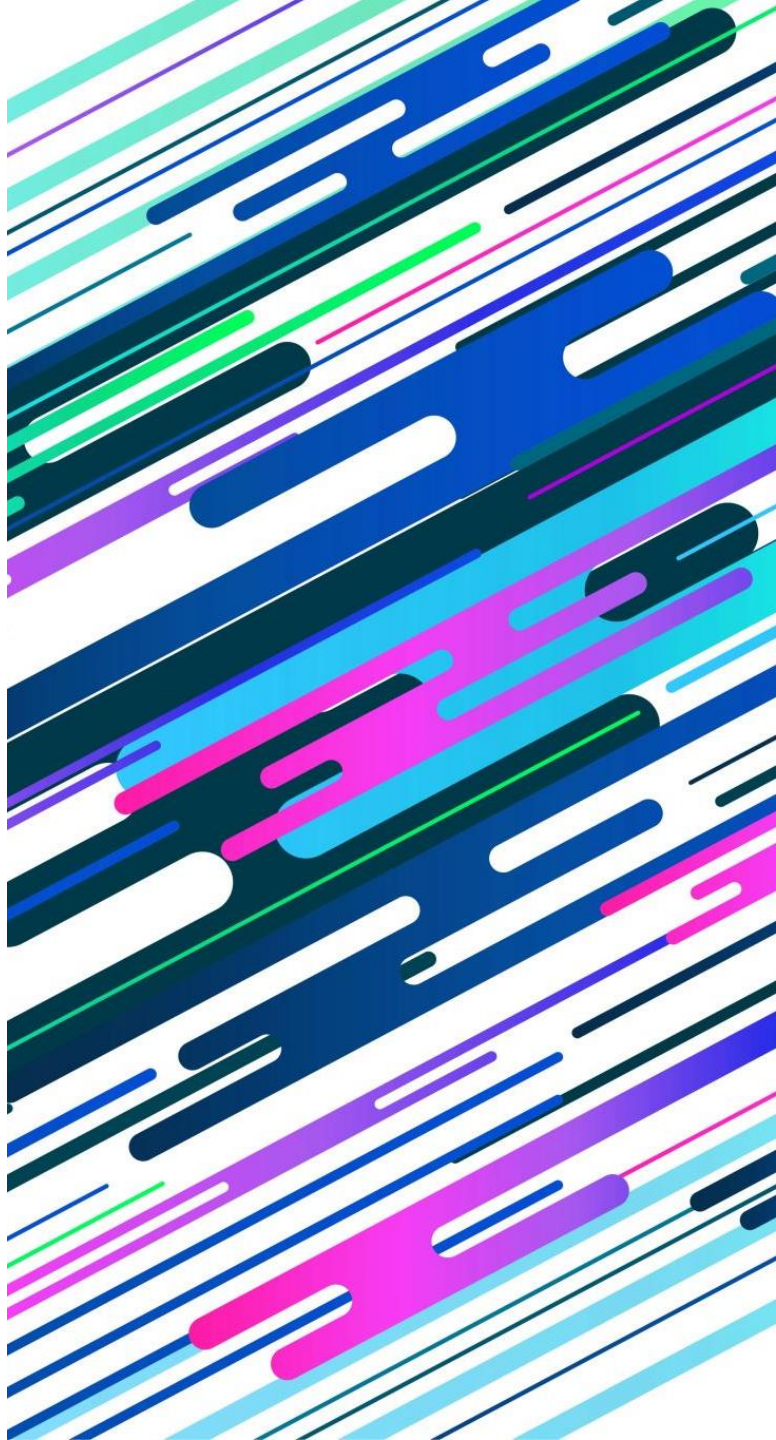
LOGISTIC REGRESSION

Thomas Tiam-Lee, PhD

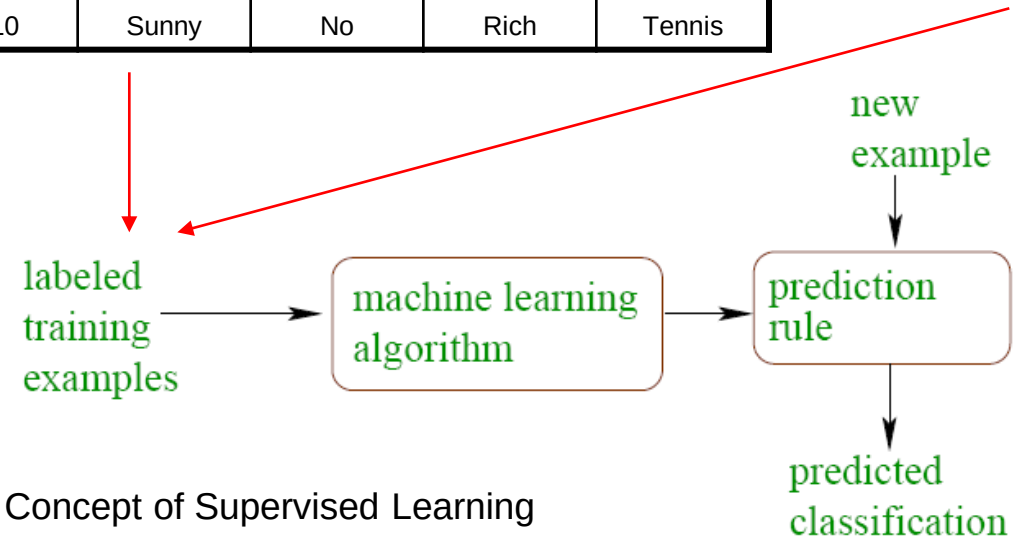
Norshuhani Zamin, PhD



Supervised Machine Learning

Weekend	Weather	Parents Visiting	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

Student	Dean's List Jan Semester?	Male?	Work Hard?	Play a Lot?	Dean's List July Semester?
Anwar	Yes	Yes	No	Yes	Yes
Ameer	Yes	Yes	Yes	No	Yes
Lilian	No	No	Yes	No	Yes
Raveen	No	Yes	No	Yes	No
Fatimah	Yes	No	Yes	Yes	Yes
Eddy	Yes	Yes	Yes	No	No



W11 Rainy Yes Rich ???

John Yes Yes No No ???

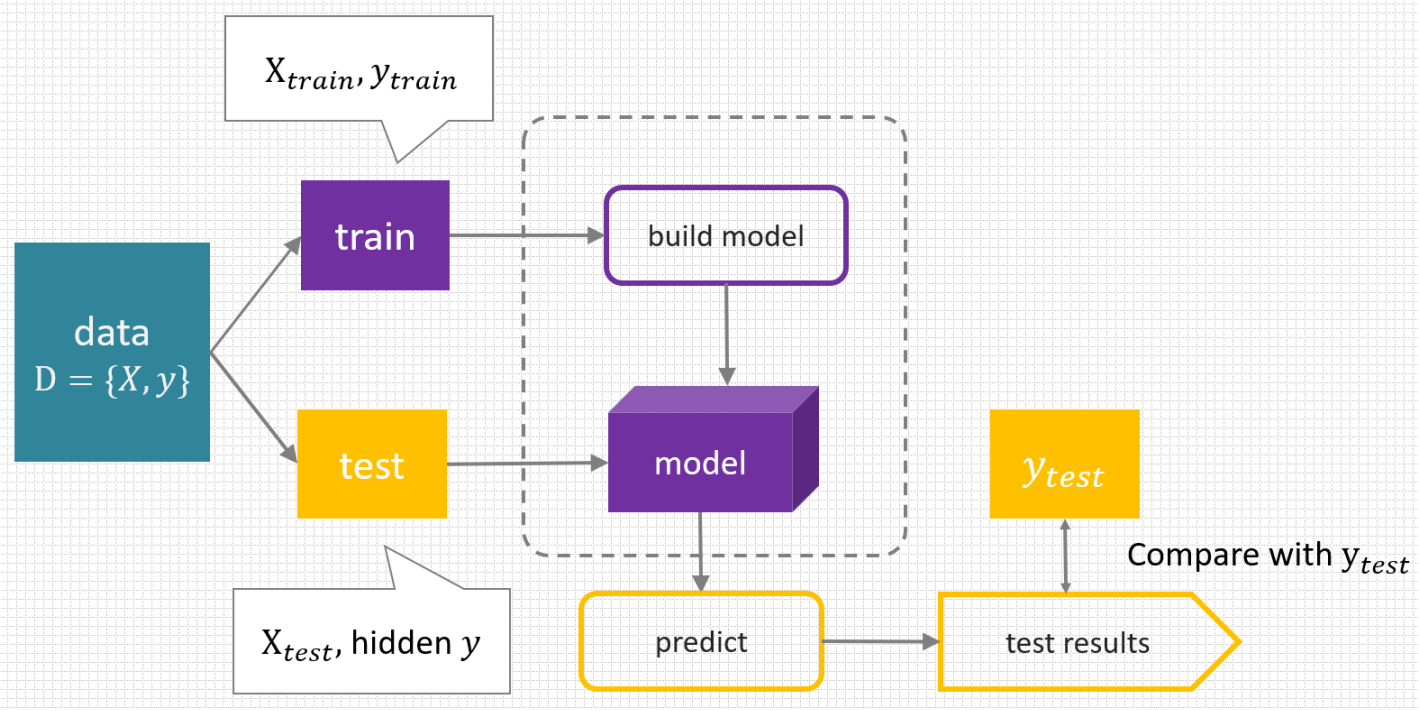
Decision: Cinema,Tennis, Shopping or Stay in? **Multi-class**

Obtain Dean's List for July semester? Yes or No **Binary**

Concept of Supervised Learning

Supervised ML Outputs

- **Binary Classification**
  - Given the input x, returns “yes” or “no”
  - Eg: Predict if an email is spam or not



- Divide the data into two groups: training set and test set.
- Use the training set as input to the learning algorithm and build the model.
- Use the test set to make predictions using the model.
- Evaluate the performance by checking the correctness of the predictions.

Basic Machine Learning Pipeline

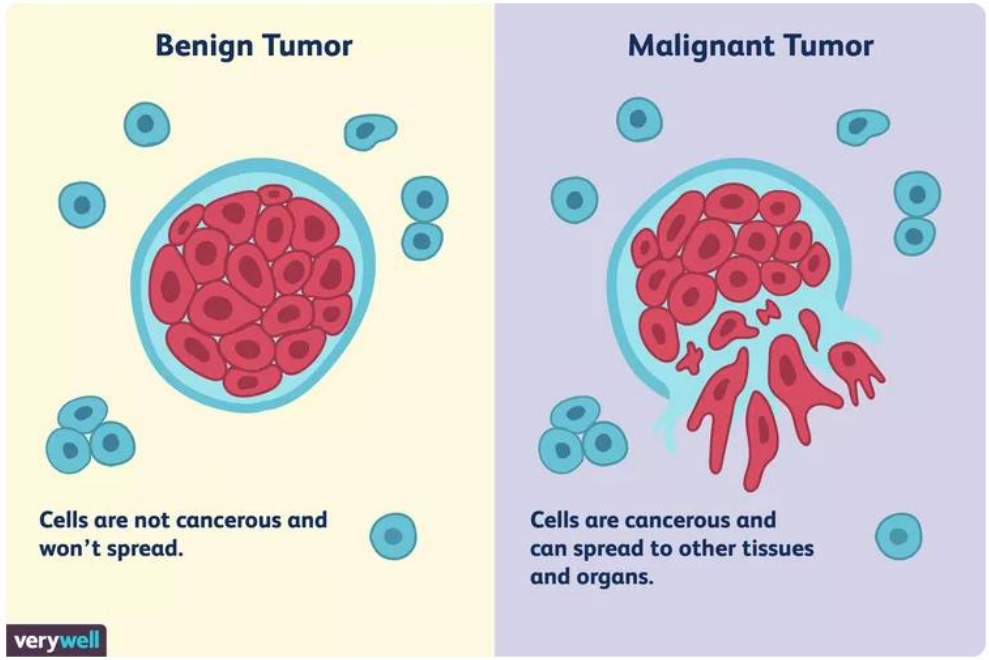
- Divide the data into two groups: training set and test set.
- Use the training set as input to the learning algorithm and build the model.
- Use the test set to make predictions using the model.
- Evaluate the performance by checking the correctness of the predictions.

Logistic Regression

- A **supervised learning algorithm**
  - Contains a target variable (label) that we want to predict

- A model designed for **binary classification**
  - The label is a yes/no categorical variable

- **Example:** *predict whether a tumor is malignant/benign given its size*

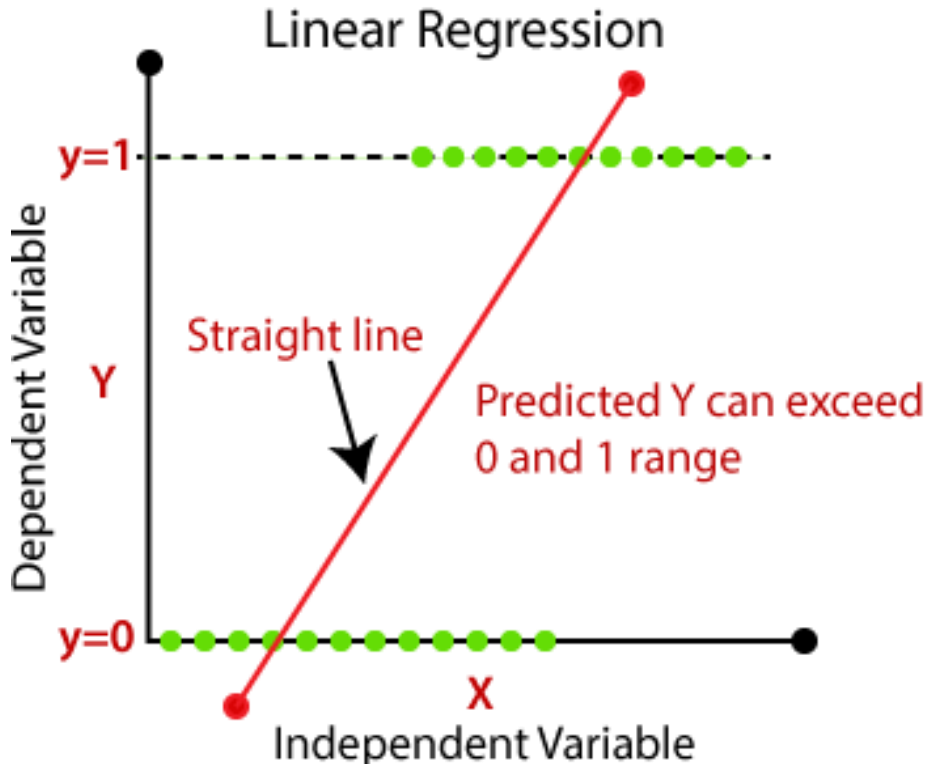


Logistic Regression

- Based on a similar idea to linear regression
- Model outputs a numerical value
- The numerical value is mapped to a categorical output (yes/no)

Problem of Linear Regression for Classification

- Predicted value can exceed the 0 – 1 range (does not make sense for binary classification)
- Lacks flexibility to handle certain configurations



Source: Javatpoint

Modeling: Logistic Regression

- Solution: feed the linear regression output to a **sigmoid function**

- Properties of sigmoid function:
  - Always returns a value between 0 and 1 (exclusive)
  - S-shape

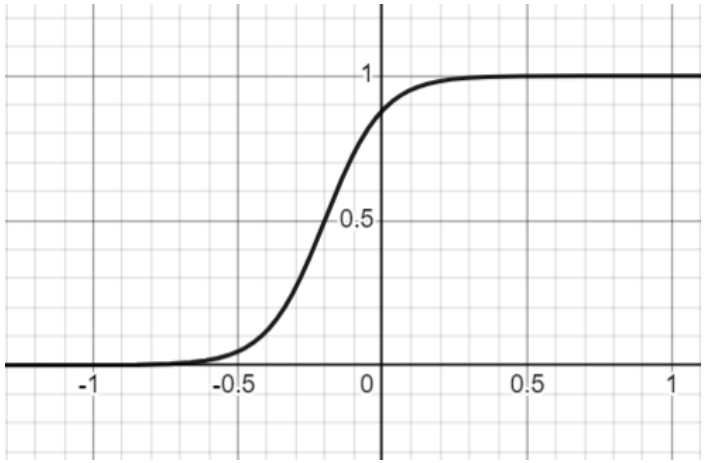
Sigmoid Function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Log. Reg. Model

$$y = \frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + w_0)}}$$

Original linear regression model



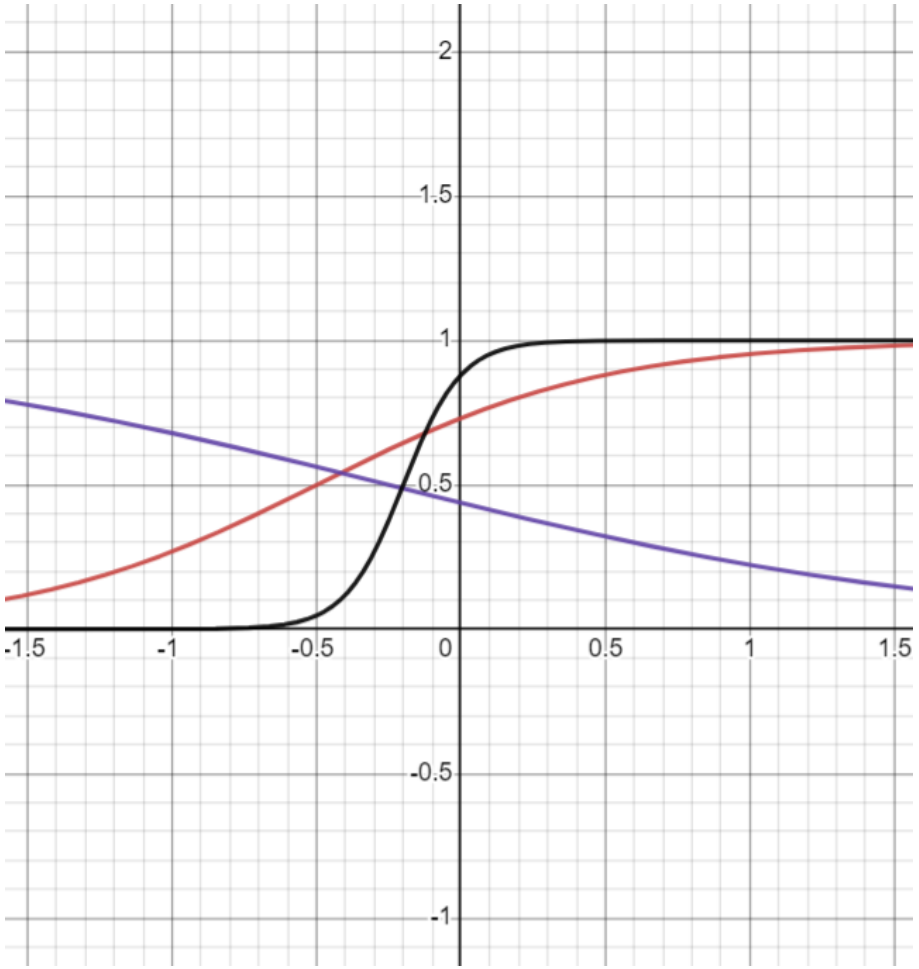
Modeling: Logistic Regression

- **Key Idea:** By **changing the parameters** of the model, we can **change how the model behaves** (i.e., the orientation of the curve)

$w_1 = 2$
$w_0 = 1$

$w_1 = -1$
$w_0 = -0.25$

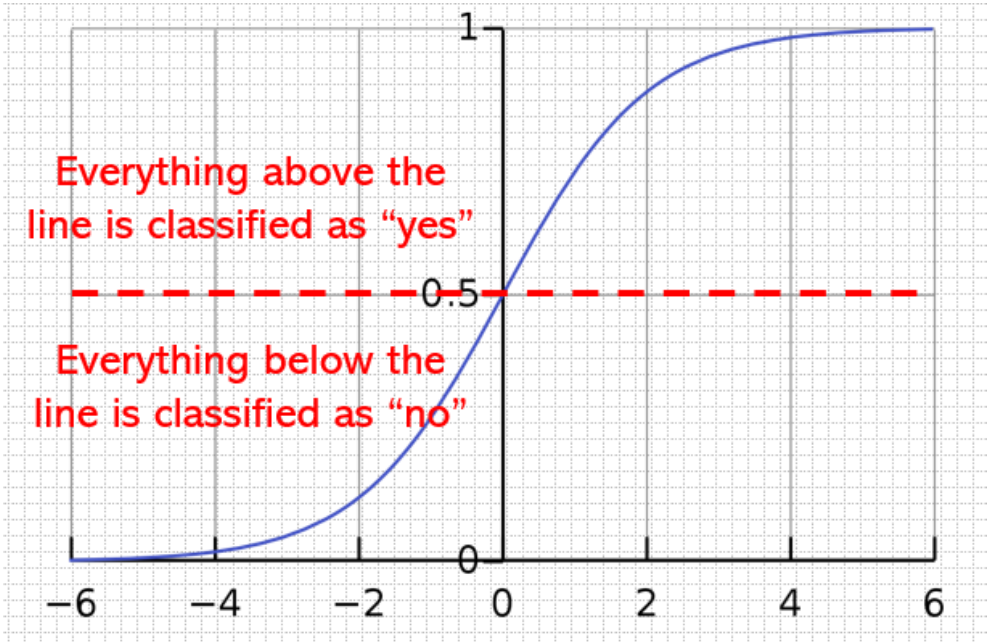
$w_1 = 10$
$w_0 = 2$



Interpretation of Output

- We interpret the regression output as the **confidence of the model at the data point is classified as “yes” or 1**.

- Mapping to class:
  - Set a threshold (i.e., 0.5)
    - If output ≥ threshold, classify as “yes”
    - If output ≤ threshold, classify as “no"



Logistic Regression Loss Function

$$l(\theta) = -\sum y_i \log(\sigma(x;\theta)) + (1 - y_i)\big(\log\big(1 - \sigma(x;\theta)\big)\big)$$

Logistic Regression Loss Function

$$l(\theta) = -\sum y_i \log(\sigma(x;\theta)) + (1 - y_i)\big(\log\big(1 - \sigma(x;\theta)\big)\big)$$

used when the real label is 1 (true)

used when the real label is 0 (false)

Logistic Regression Loss Function

$$l(\theta) = -\sum y_i \log(\sigma(x;\theta)) + (1 - y_i)\big(\log\big(1 - \sigma(x;\theta)\big)\big)$$

used when the real label is 1 (true)

used when the real label is 0 (false)

- **Key idea:** scores that are closer to the real labels should have less penalty.
- Log function is applied to ensure the resulting loss function is convex.
  https://study.com/academy/lesson/convex-definition-shape-function.html

Goal of Learning Algorithm

$$l(\theta) = -\sum y_i \log(\sigma(x;\theta)) + (1 - y_i)\big(\log\big(1 - \sigma(x;\theta)\big)\big)$$

- How do we find the set of parameters θ that will **minimize the loss function**?

Gradient Descent Learning Algorithm

procedure Gradient Descent(θ):
while not converged do:

$$\theta_i := \theta_{i-1} - \alpha \frac{\partial y}{\partial x}$$

return θ

θ is the slope

α is the learning rate, determines how large the update will be.

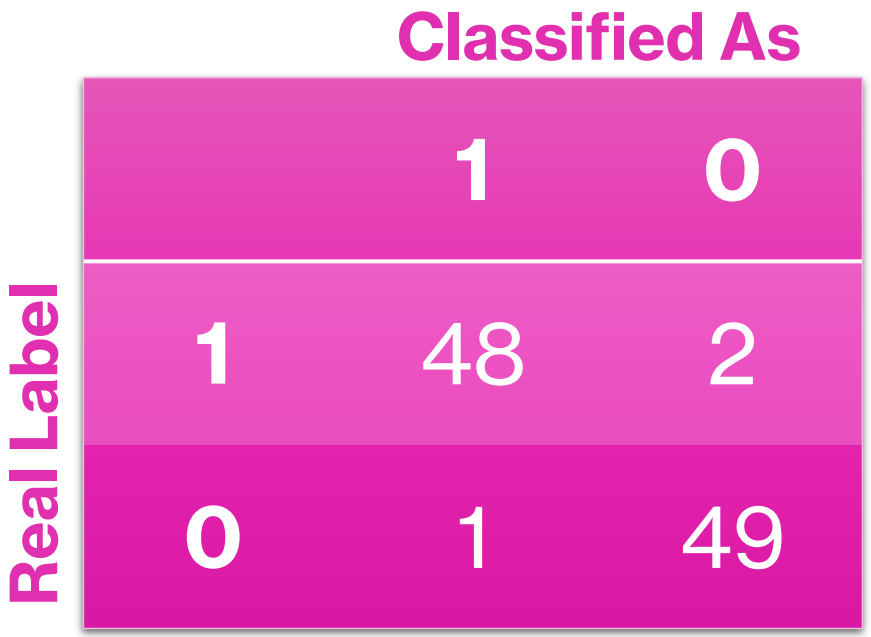
$\frac{\partial y}{\partial x}$ is the gradient of the loss

Evaluating Classification Results

- Confusion matrix
- Accuracy
- Precision
- Recall
- F1-Score

Confusion Matrix

- Shows us the statistics of the prediction

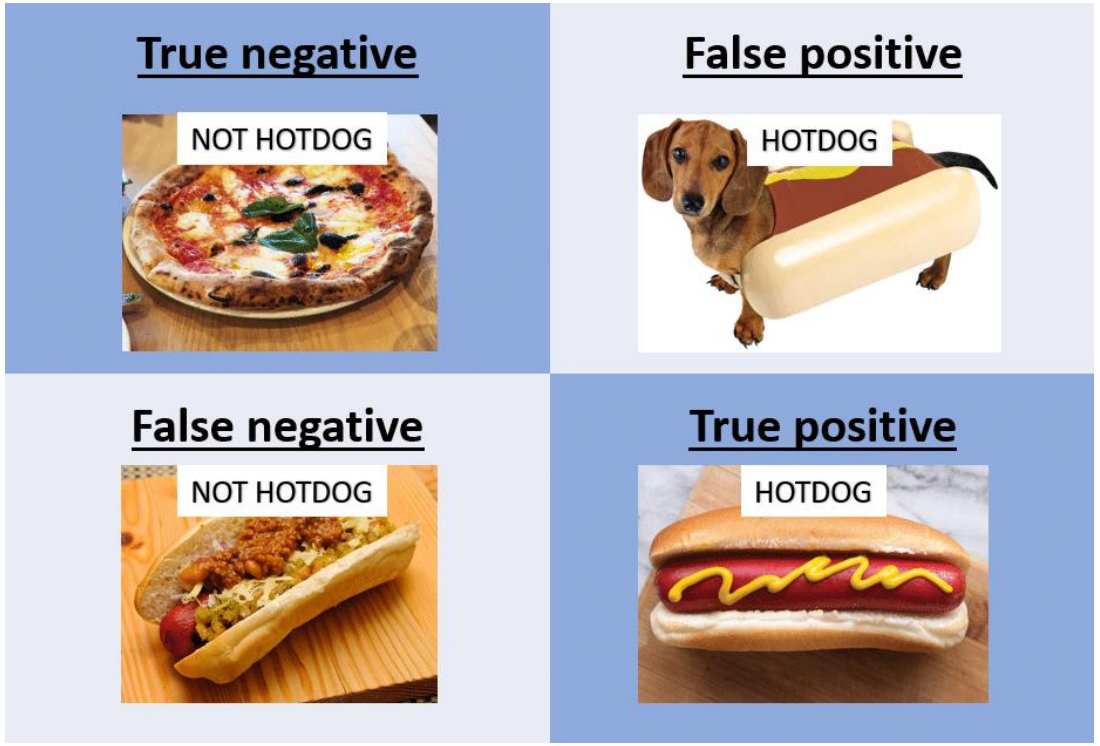


Confusion Matrix

- Shows us the statistics of the prediction

Real Label \ Classified As	1	0
1	True positive	False positive
0	False negative	True negative

The Analogy..



Accuracy

- Number of correctly classified instances over all instances.

$$\frac{TP + TN}{TP + TN + FP + FN}$$

		Classified As: 1	Classified As: 0
Real Label	1	**True positive**	False positive
Real Label	0	False negative	**True negative**

Precision

- Out of all the instances predicted as positive, how many are really positive?

$$\frac{TP}{TP + FP}$$

		Classified As	
		1	**0**
Real Label	**1**	**True positive**	False positive
	0	False negative	True negative

Recall

- Out of all the instances that are actually positive, how many did you predict as positive?

$$\frac{TP}{TP + FN}$$

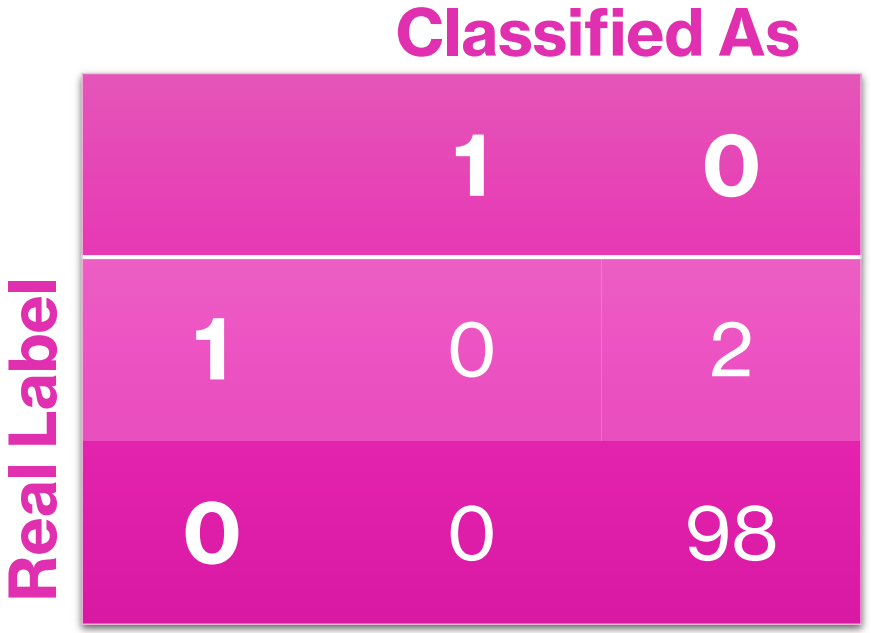
		Classified As: 1	Classified As: 0
Real Label	1	**True positive**	False positive
Real Label	0	False negative	True negative

Metrics

- **Accuracy:** Number of correctly classified instances over all instances.
  - $\frac{TP+TN}{TP+TN+FP+FN}$
- **Precision:** Out of all things you predicted positive, how many are really positive?
  - $\frac{TP}{TP+FP}$
- **Recall:** Out of all the things that are really positive, how many did you correctly predict?
  - $\frac{TP}{TP+FN}$

Why is Accuracy Not Enough?

- Very high accuracy (98%), but very stupid model (just predicts negative every time)
- When the classes are imbalanced, must consider other metrics other than the accuracy.



F1-Score

- Harmonic mean of the precision and recall (summarizes the two metrics).

$$2 \times \frac{precision \times recall}{precision + recall}$$

- When there is a large class imbalance, F-Score is more helpful than accuracy.
- The interpretation of F1-Score →

F1 score	Interpretation
> 0.9	Very good
0.8 - 0.9	Good
0.5 - 0.8	OK
< 0.5	Not good

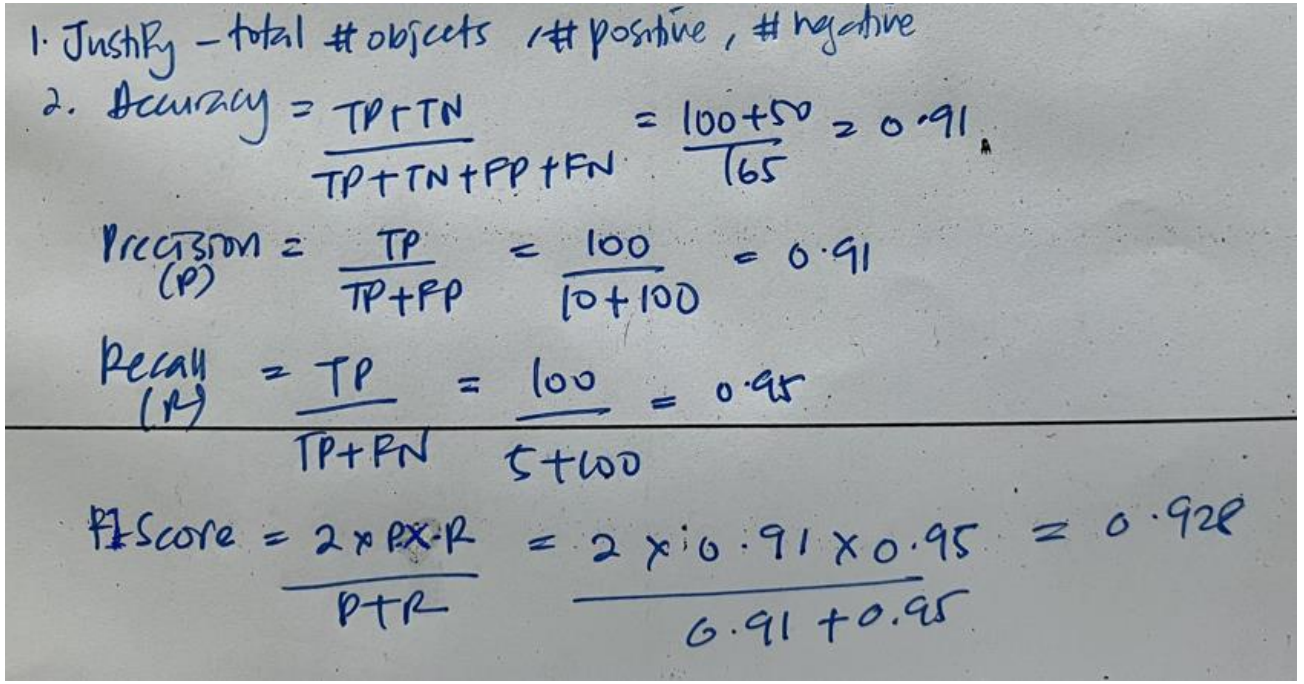
Try this:

You are given the following Confusion Matrix:

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

1. What can you learn from the matrix?
2. Calculate the Accuracy, Precision, Recall and F1 Score. Make an interpretation on its overall performance.

n=165	Predicted: NO	Predicted: YES
Actual: NO	50 TN	10 FP
Actual: YES	5 FN	100 TP



Extension to Multi-Class Classification

- When dealing with multiple classes, we can generally:
  - Use algorithms / models designed to handle multiple classes (decision trees, random forest, Naïve Bayes, etc.)
- Treat them as separate binary classification problems
  - For example, to classify between dog, cat, and mouse, create 3 models:
  - Model 1: Is it a cat or not?
  - Model 2: Is it a dog or not?
  - Model 3: Is it a mouse or not?

Acknowledgments

- Previous STINTSY slides by the following instructors:
  - Courtney Ngo
  - Arren Antioquia