Machine Learning!!

Day 3: Allison and Kyle

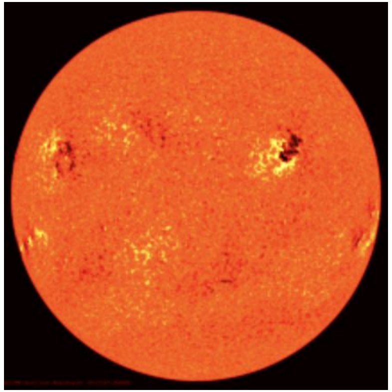
Which cats are real?

https://thiscatdoesnotexist.com/

About Me - Allison

I am a rising Senior at CU Boulder studying Applied Mathematics and Computer Science.

I am using Machine Learning to predict solar flares from the Sun.



Overview

- Intro and Motivation
    - What is Machine Learning? Why do we care about it?
    - Where do we see ML in our everyday lives?
    - Why is it useful?
- ML Basics
    - What are features?
    - Machine learning models
    - DATA! (and data splitting)
- ML Projects
    - What is the workflow of an ML project

What is Machine Learning?

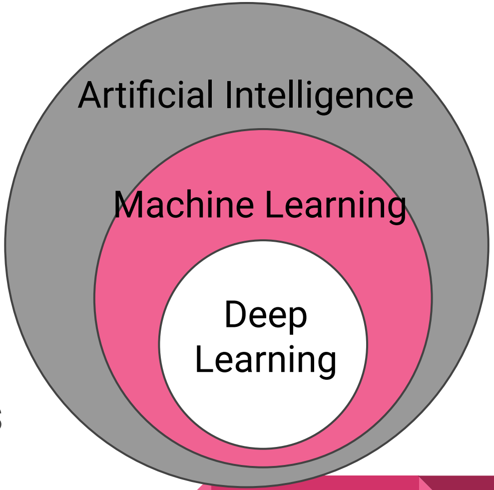
- A subset of computer science that uses data to improve predictions without explicit instruction
  - Like we can learn from experience, so can computers!

Other "buzzwords":

Artificial Intelligence: Computers that mimic intelligence

Sense, reason, adapt, react

Deep Learning: Complex algorithms and neural networks



Why do we care?

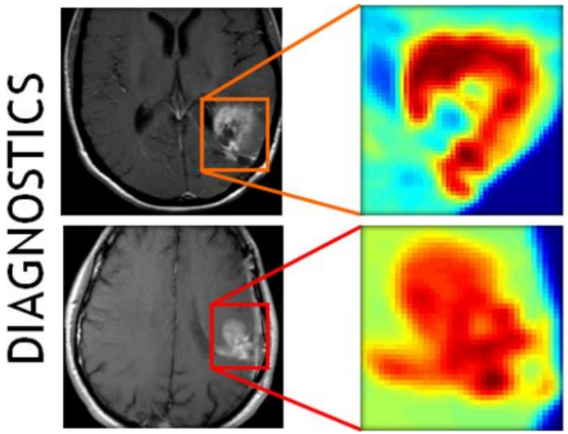
- Computers are tools we can use to solve problems
- Machine learning allows us to process information much faster than humans could alone
- We generate a lot of data!
  - Applications:
    - Health care
    - Social media
    - Retail
    - Manufacturing
    - Security
    - Transportation
    - Real estate
    - Gaming

Difficulties:

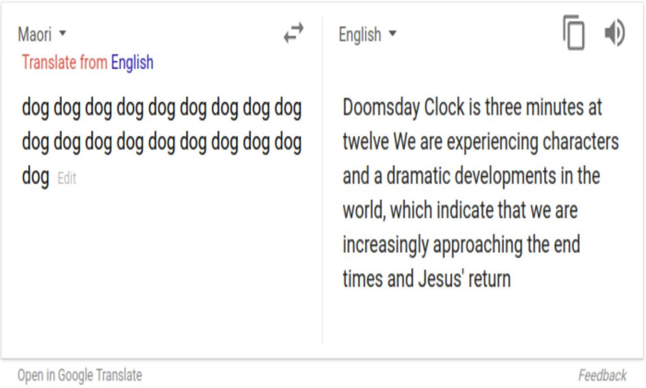
- Understanding how a machine arrived at a result

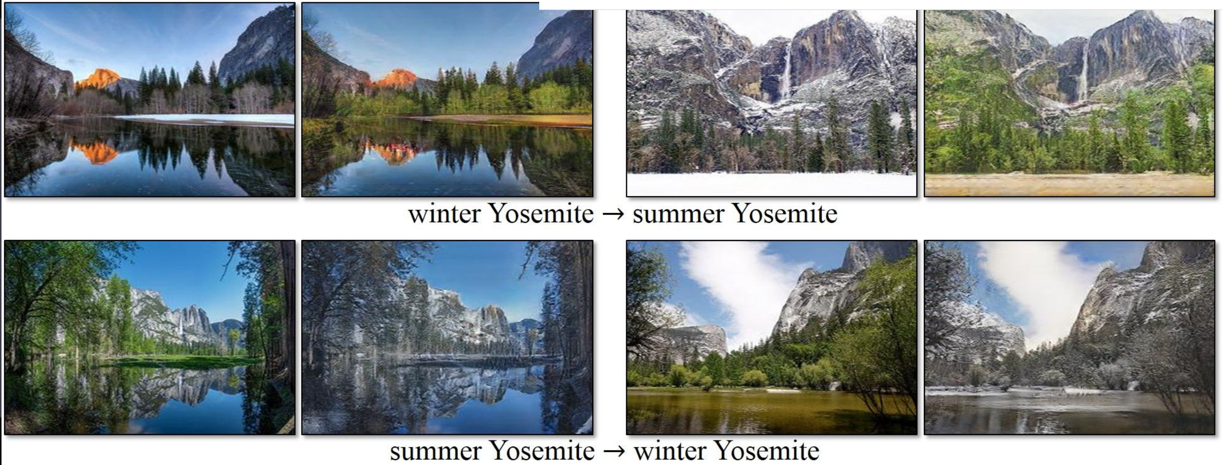
Source: Intel

Where do we see ML in our everyday lives?



More ML Examples:





What types of problems are good for ML?

- Examining patterns in data
    - Lots of samples/data points
- Solving problems that would be difficult to solve manually or using traditional programming: copying human behavior/decision making

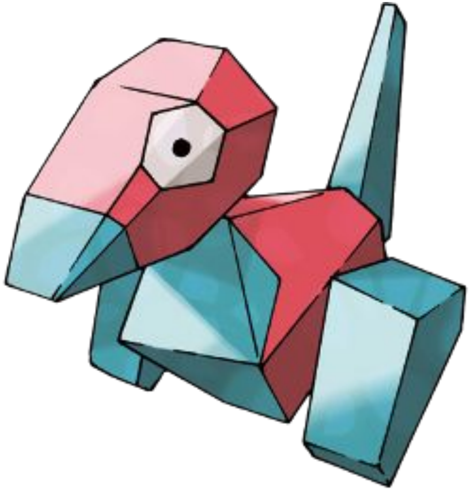
Why has it become so popular?

- Data
- Computational power
- Models have been developed that are easy and accessible to use

Features

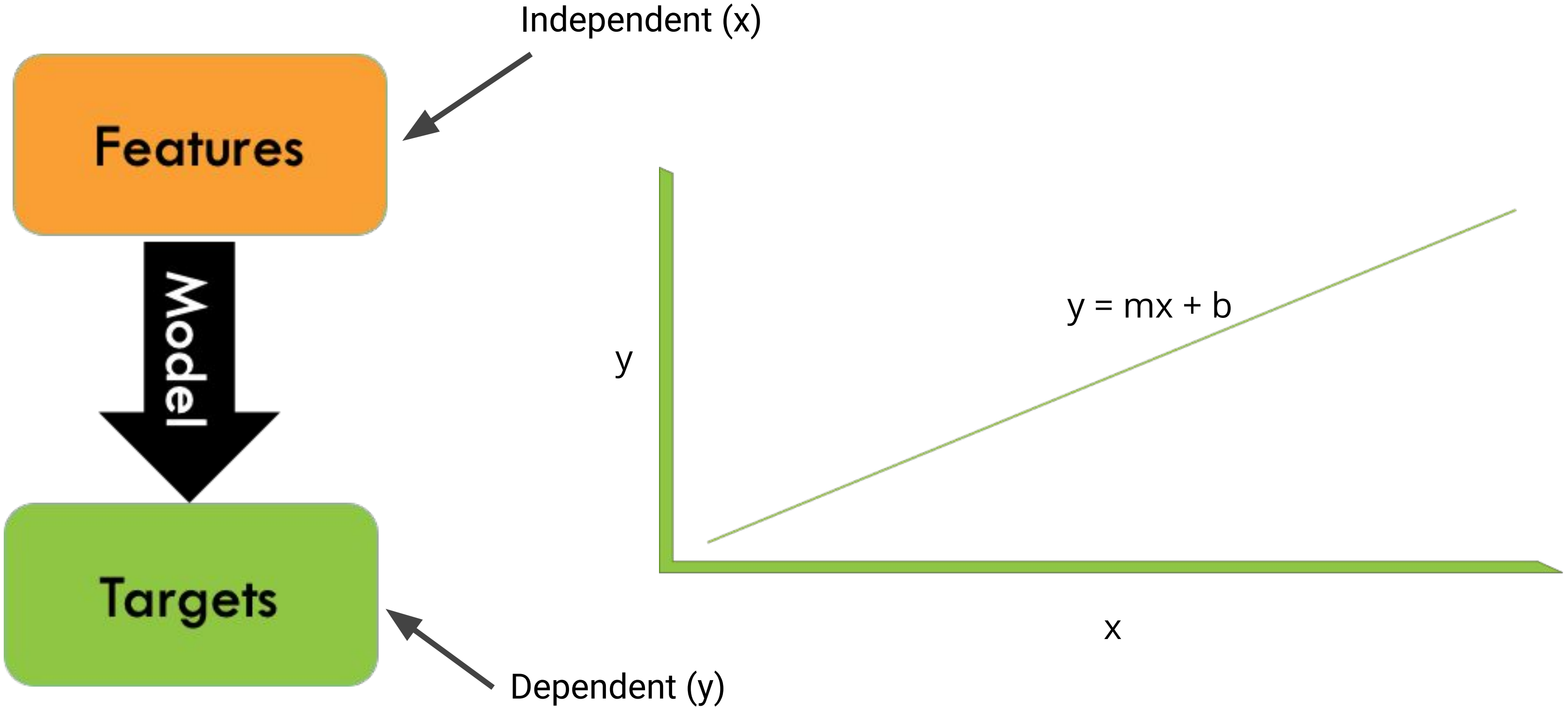
How do we get information from data?

HP



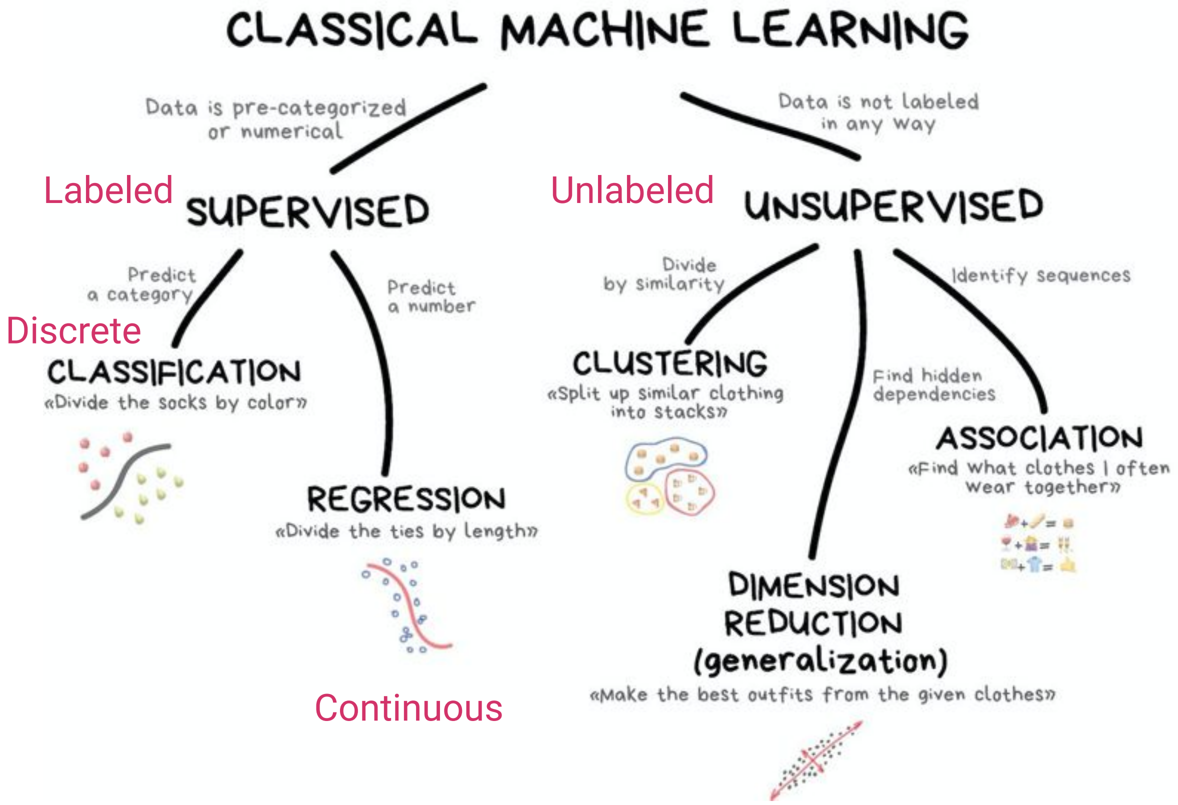
Source: Wendy Carande

ML Basics



What are examples of features Facebook might use to decide what content to show you?

Machine Learning

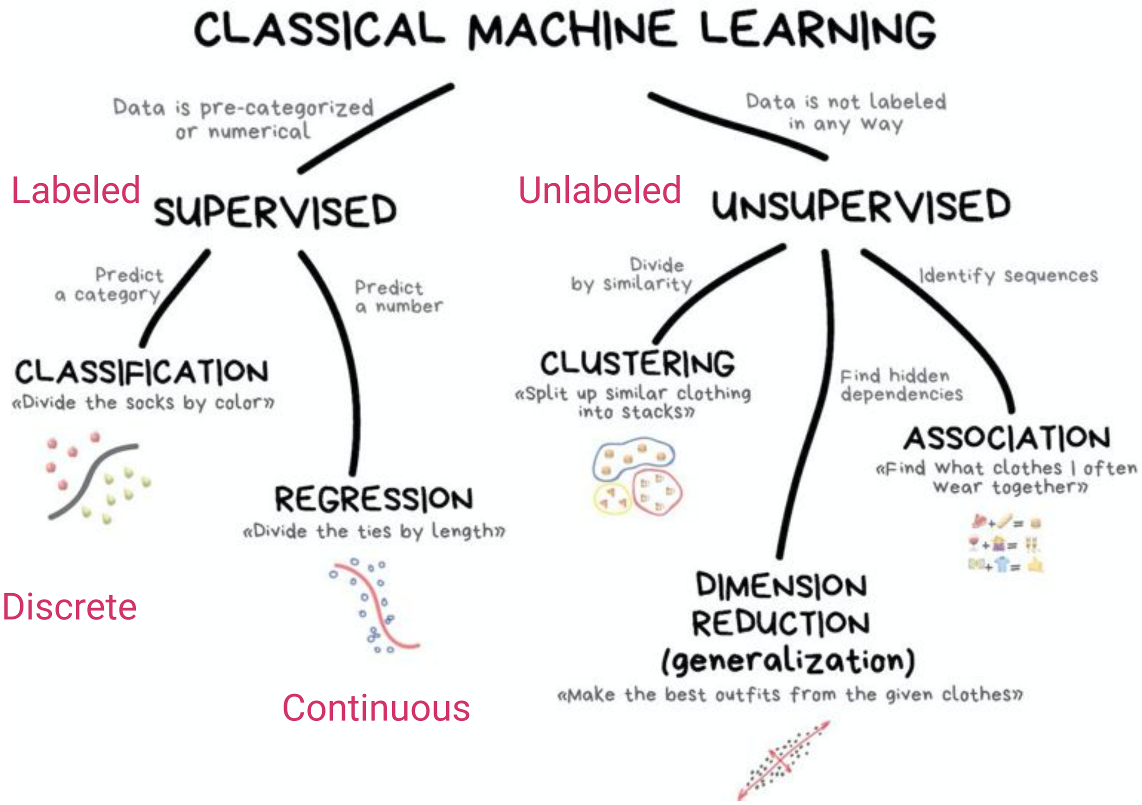


You have a bunch of labeled images of bananas and toasters and you want to know whether the image is a banana or a toaster.

Supervised or unsupervised?

Classification or Regression?

Machine Learning



Labeled

Unlabeled

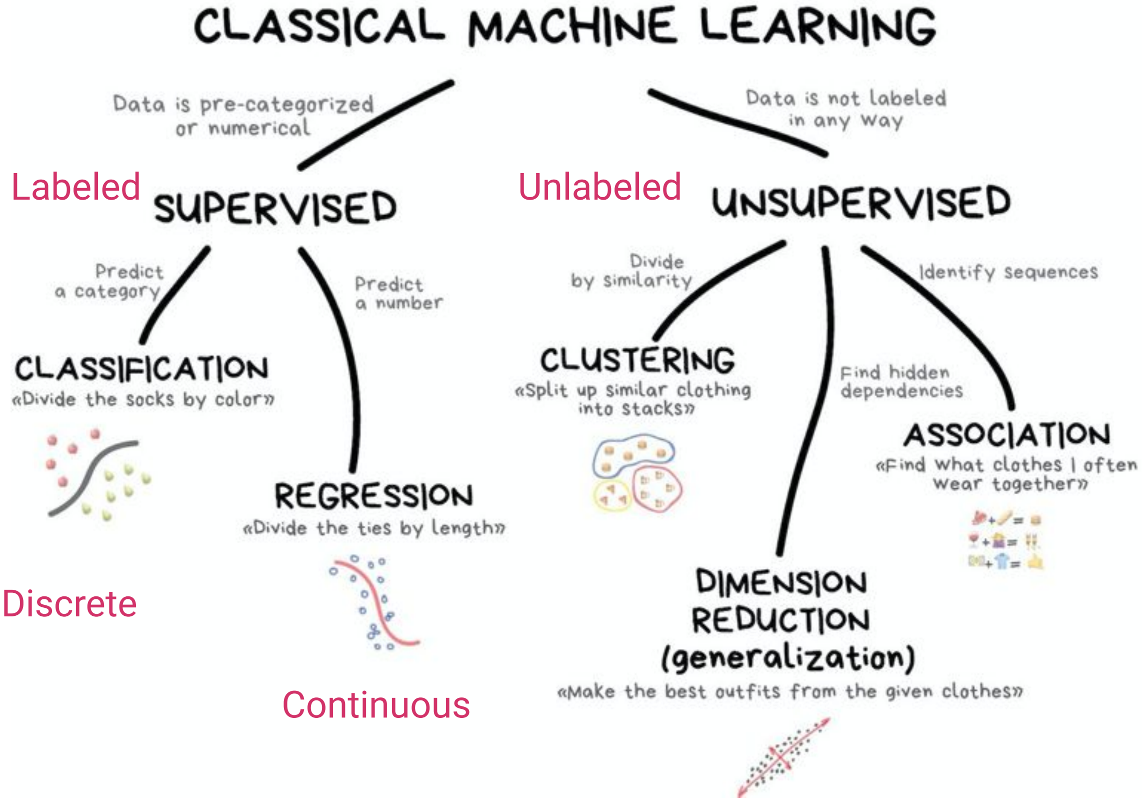
Discrete

Continuous

You have images of bananas apples, and oranges, and you want to group them to explore patterns.

Supervised or unsupervised?

Machine Learning



Labeled

Unlabeled

Discrete

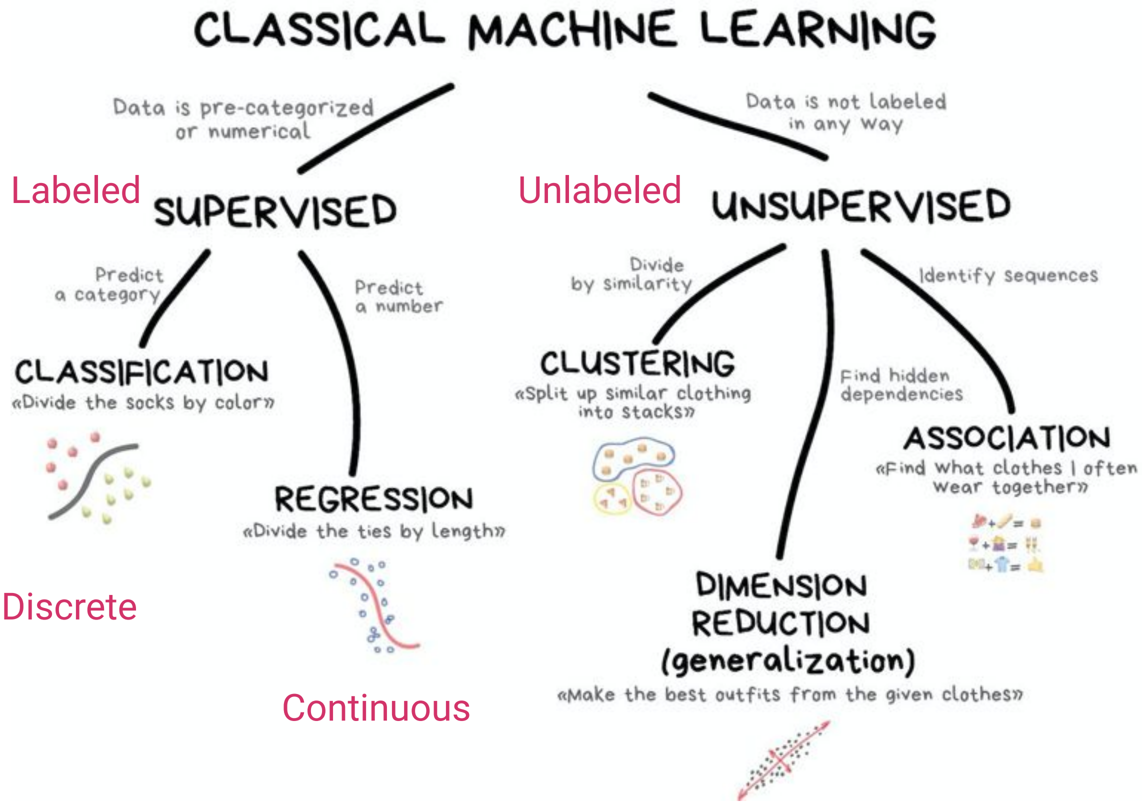
Continuous

Based on location, size, number of floors, what is the value of a given house?

Classification or Regression?

https://alison.rbind.io/post/2019-12-23-learning-to-teach-machines-to-learn/

Machine Learning



Labeled

Unlabeled

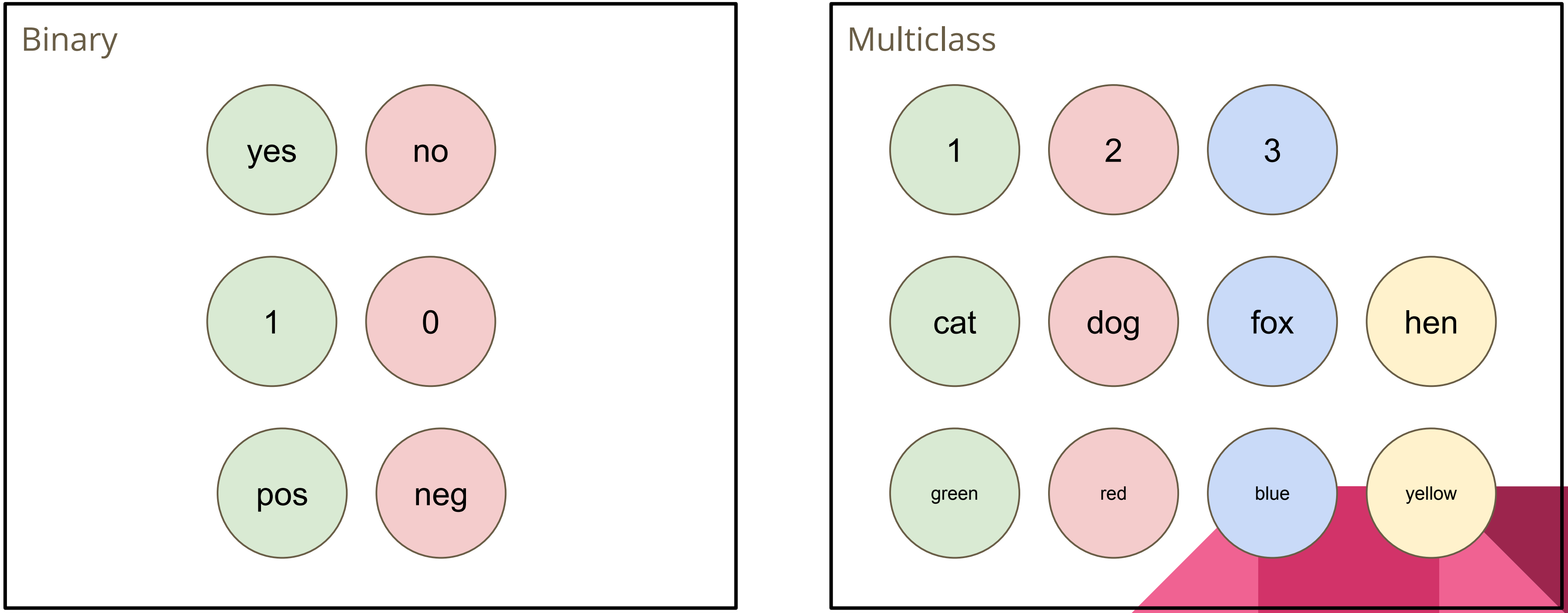
Discrete

Continuous

Given height, weight, and shoe size, you want to determine the position of a football player.

Classification or Regression?

Binary vs Multiclass Classification



Source: Wendy Carande

Machine Learning Projects

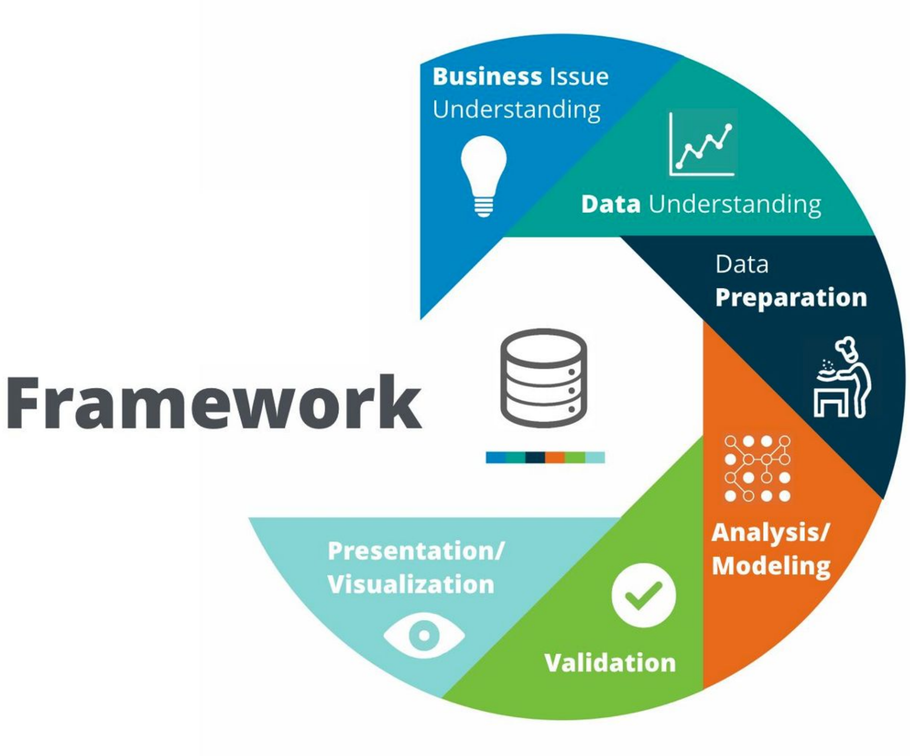
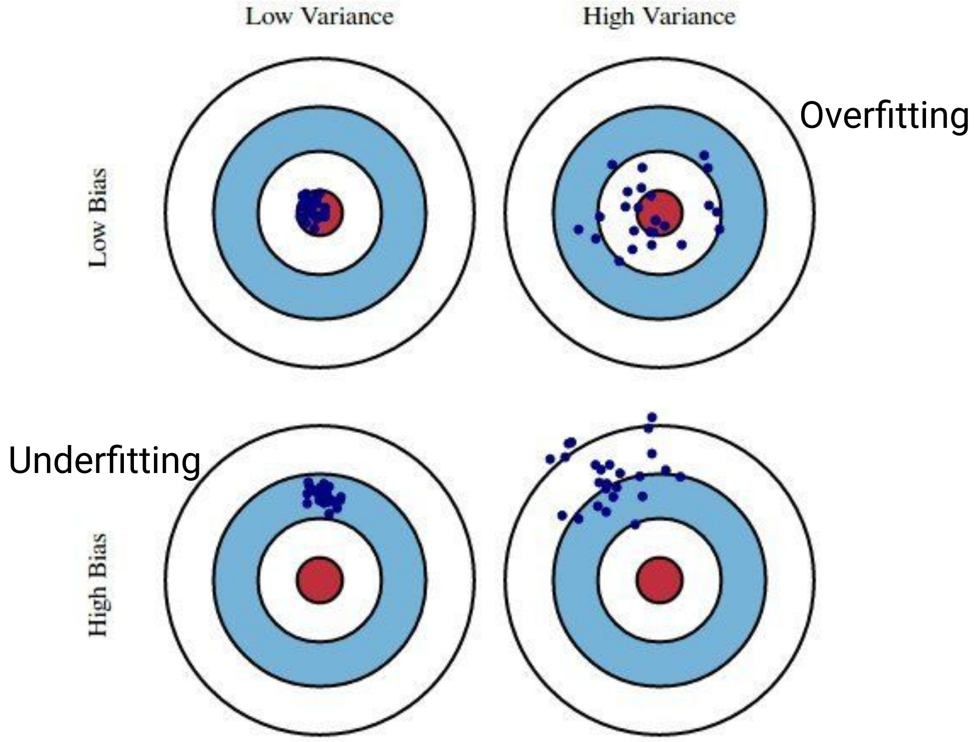


Image source: Kaggle

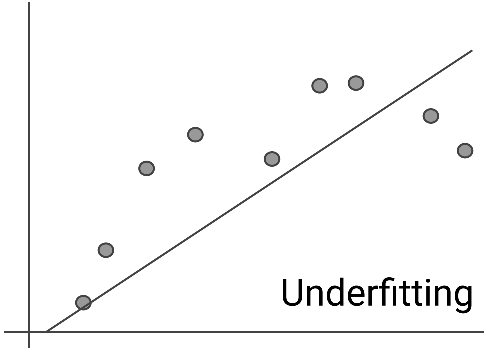
Bias-Variance Tradeoff



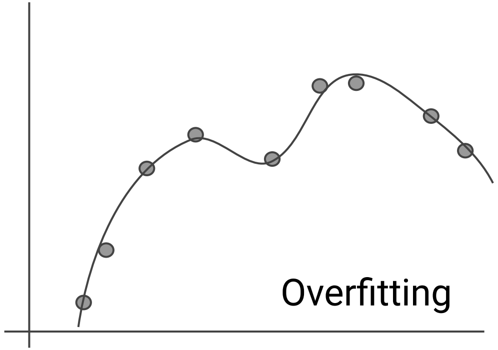
Bias: average prediction of model vs correct value

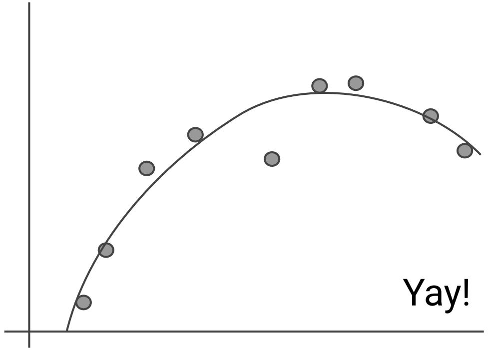
Variance: the spread of our data

Bias-Variance Tradeoff

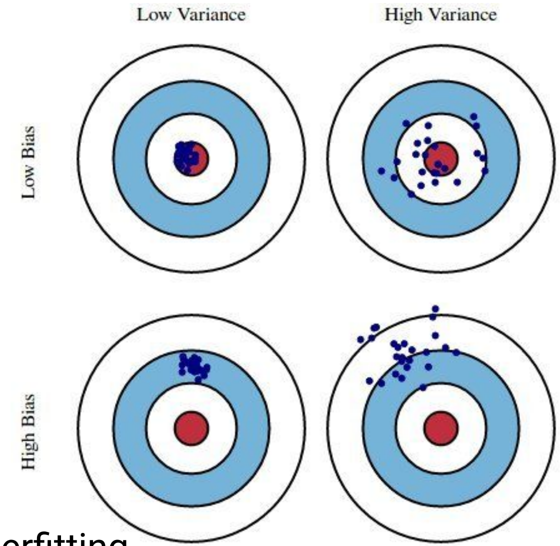


Underfitting: cannot capture underlying pattern of the data. Usually results from not enough data or fitting linear model to non-linear data
Overfitting: the model captures pattern and noise





Overfitting



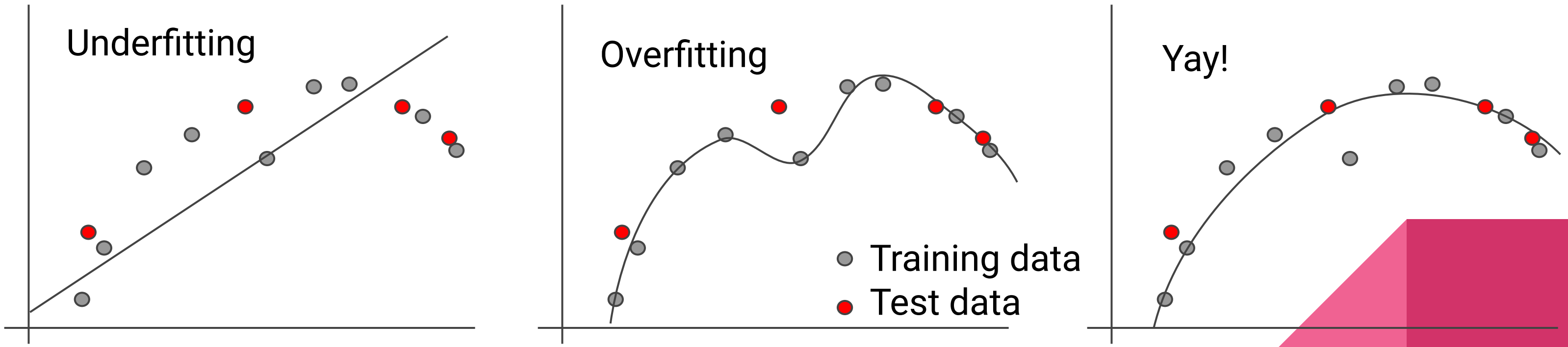
Underfitting

Data Splitting

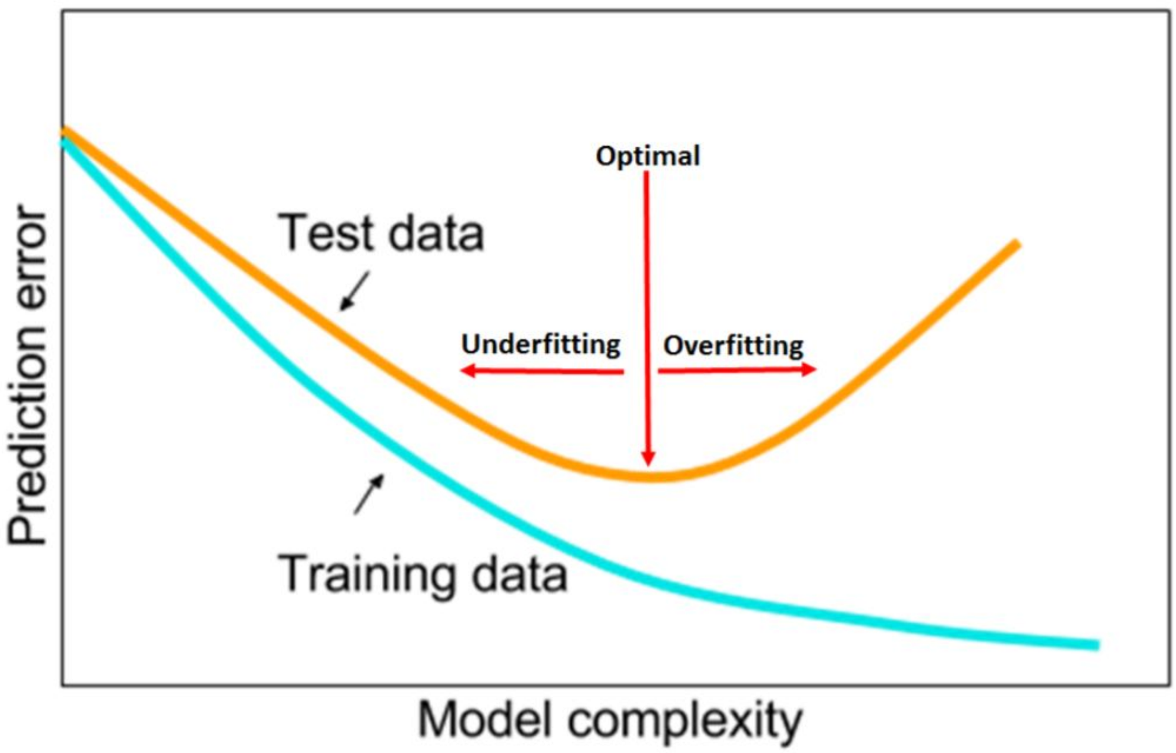
We split data into a “train set” and a “test set”.

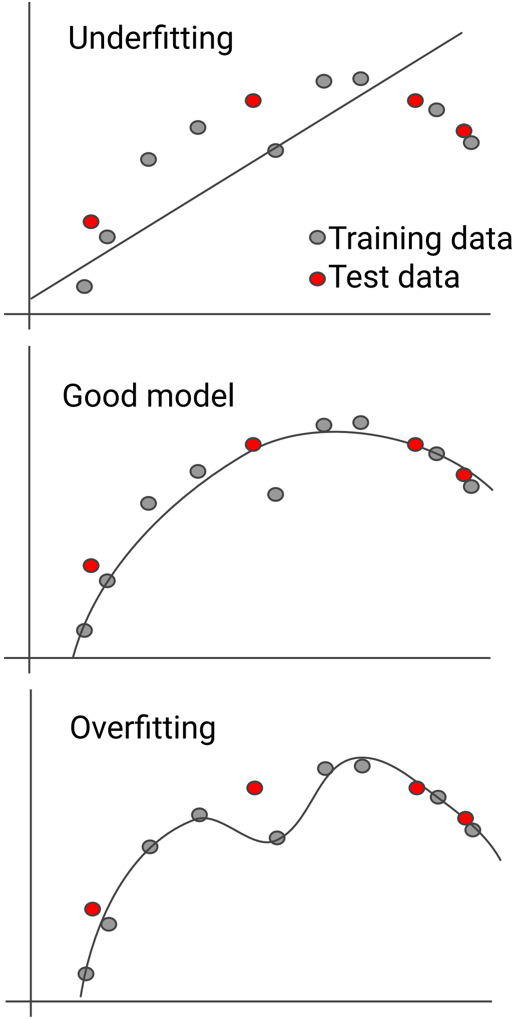
Why? It helps us to better ensure our model is not overfitting or underfitting.

For example, if a child can recite all the multiplication tables up to 4x4, we can test if they actually understand the rules of multiplication by checking if they can do 5x or 6x...



Model Tuning





https://towardsdatascience.com/hyper-parameter-tuning-techniques-in-deep-learning-4dad592c63c8

Scoring Metrics

TP True Positives: Model says yes, answer really is yes

FP False Positive: Model says yes, answer should be no

TN True Negative: Model says no, answer really is no

FN False Negative: Model says no, answer should be yes

Confusion Matrix

TN	FP
FN	TP

Accuracy: number the model guesses correctly divided by the total number of samples

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

Source: Wendy Carande

Confusion Matrices

	Predicted Class	
Actual Class	True Positive	False Positive
	False Negative	True Negative

Accuracy may not always be the best scoring metric.

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}}$$

$$\mathrm{ERR} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{P} + \mathrm{N}}$$

```
Precision = (TP)/(TP+FP)
```

```
Recall = (TP)/(TP+FN)
```

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Confusion Matrices

Dogs: 0s, Cats: 1s

----

actual   = [1,1,1,1,1,1,1,1,0,0,0,0]
prediction = [0,0,1,1,1,1,1,1,0,0,0,1]

$$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FN}+\mathrm{FP}} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{P}+\mathrm{N}}$$

$$\mathrm{ERR} = \frac{\mathrm{FP}+\mathrm{FN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FN}+\mathrm{FP}} = \frac{\mathrm{FP}+\mathrm{FN}}{\mathrm{P}+\mathrm{N}}$$

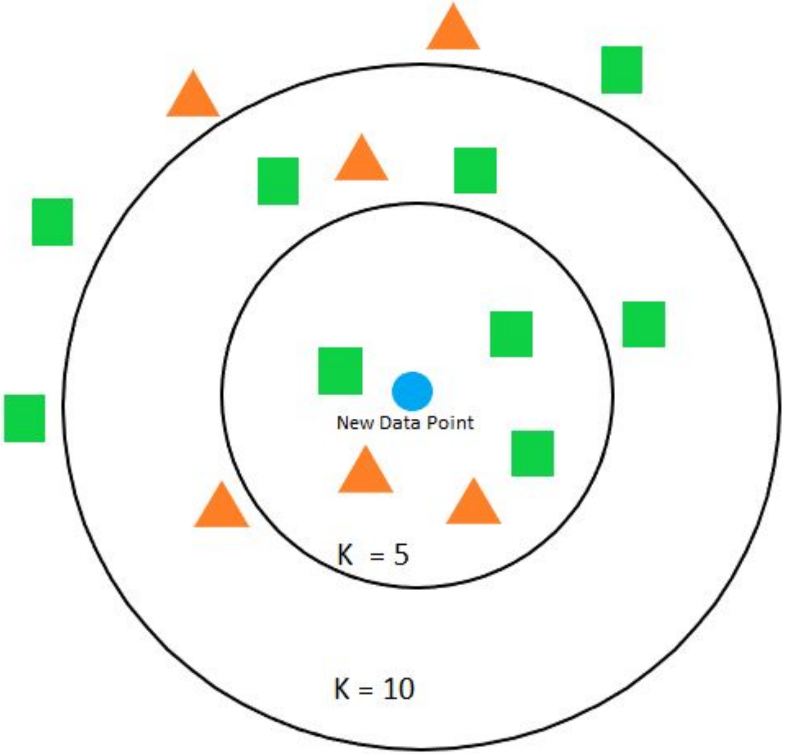
Predicted Class ~~~~~~~~~~~ Actual Class	**Cat**	**Dog**
Cat		
Dog		

Using sklearn

```
from sklearn.model_selection import train test split
from sklearn.linear_model import LinearRegression
linreg = LinearRegression()
linreg.fit(train_set)
prediction = linreg.predict(test_set)
```

```
from sklearn.model_selection import train test split
from sklearn.tree import DecisionTreeClassifier
X_train, X_test, y_train, y_test = train_test_split(X, y)
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
```

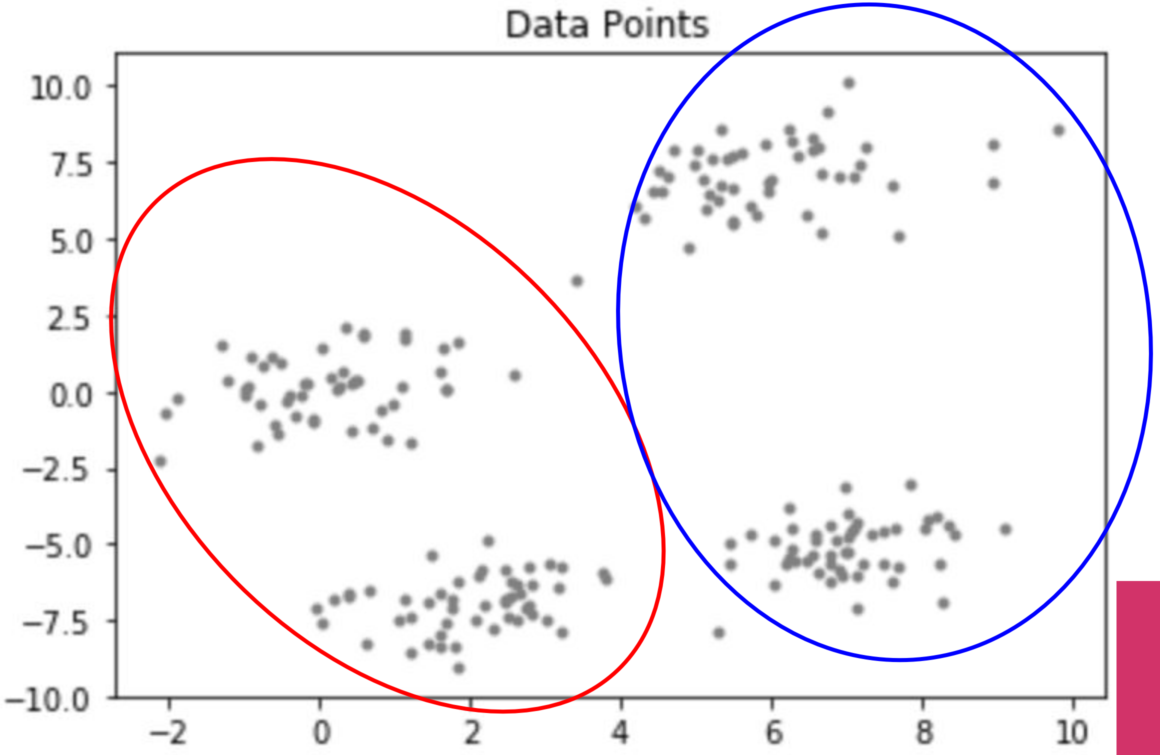
K Nearest Neighbors



https://debuggercafe.com/an-introduction-to-k-nearest-neighbors-in-machine-learning/

Parameter Tuning: Draw circles around the “clusters”

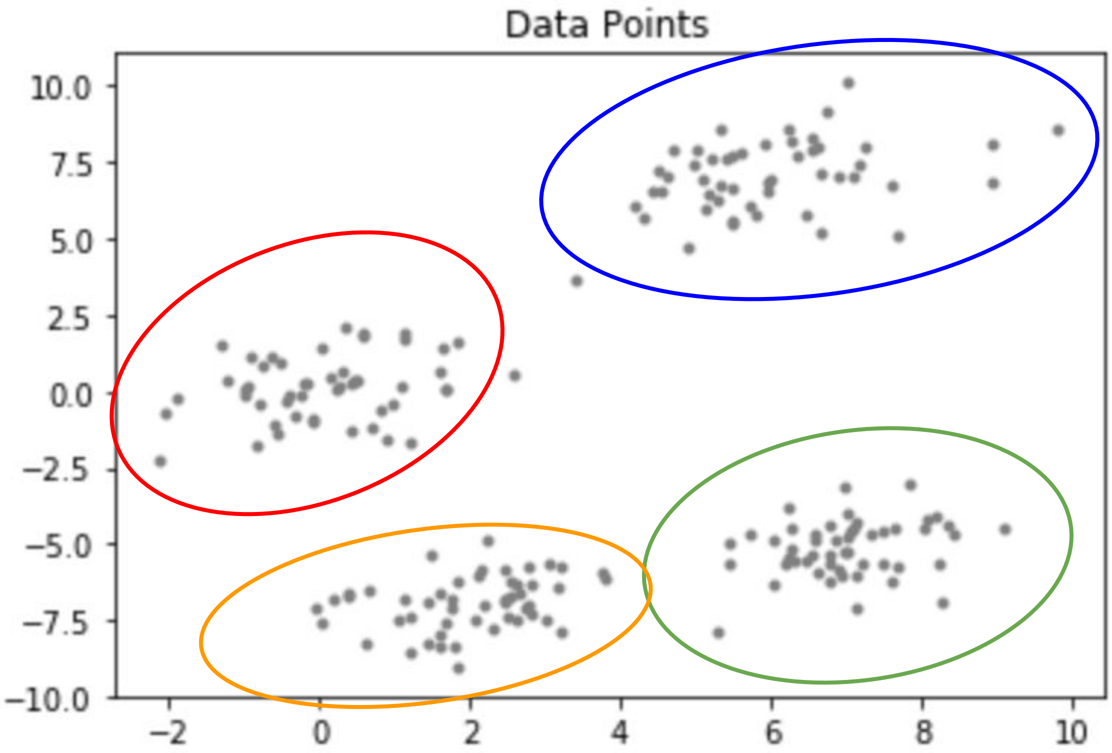
k=2



https://stackoverflow.com/questions/52752836/organizing-clusters-in-k-means-clustering/52770392

Parameter Tuning: Draw circles around the “clusters”

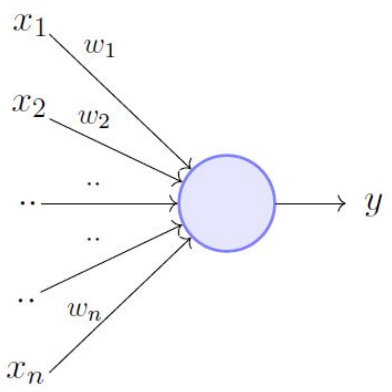
k=4



https://stackoverflow.com/questions/52752836/organizing-clusters-in-k-means-clustering/52770392

Perceptron

- The Perceptron algorithm is a two-class (binary) classification machine learning algorithm.
- The Perceptron is a linear classification algorithm. This means that it learns a decision boundary that separates two classes using a line (called a hyperplane) in the feature space.
- A type of neural network model, perhaps the simplest type of neural network model.
- It consists of a single node or neuron that takes a row of data as input and predicts a class label. This is achieved by calculating the weighted sum of the inputs and a bias (set to 1). The weighted sum of the input of the model is called the activation.
- **Activation** = Weights * Inputs + Bias



$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i \geq \theta$$

$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i < \theta$$

Rewriting the above,

$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i - \theta \geq 0$$

$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i - \theta < 0$$