Training of Trainers Bootcamp on Machine Learning for Earth Observations

Introduction to Machine Learning

05/05/2021

Joyce Nakatumba-Nabende, PhD
Department of Computer Science
Makerere University

Introduction

- All useful programs "learn" something.
- Early definition of machine learning: - "Field of study that gives computers the ability to learn without being explicitly programmed." (Arthur Samuel, 1959)
- There is no need to "learn" to calculate payroll.
- Learning is used when:
    - Human expertise does not exist (navigating on Mars)
    - Humans are unable to explain their expertise (speech recognition)
    - Solution changes in time (routing on a computer network)

What is Machine Learning?

Traditional Programming

Square root finder

Machine Learning

Curve fitting by linear regression

How are Things Learned?

- Memorization
    - Accumulation of individual facts
    - Limited by
        - Time to observe facts
        - Memory to store facts
- Generalization
    - Deduce new facts from old facts
    - Limited by the accuracy of deduction process
        - Essentially a predictive activity
        - Assumes that the past predicts the future
- Interested in extending to programs that can infer useful information from implicit patterns in data.

Declarative knowledge

Imperative knowledge

Basic Paradigm

- Observe a set of examples: training data

Prices of houses based on number of bedrooms,

- Infer something about process that generated data

Fit a linear model

- Use inference to make predictions about previously unseen data: test data

Predict house price for a new house

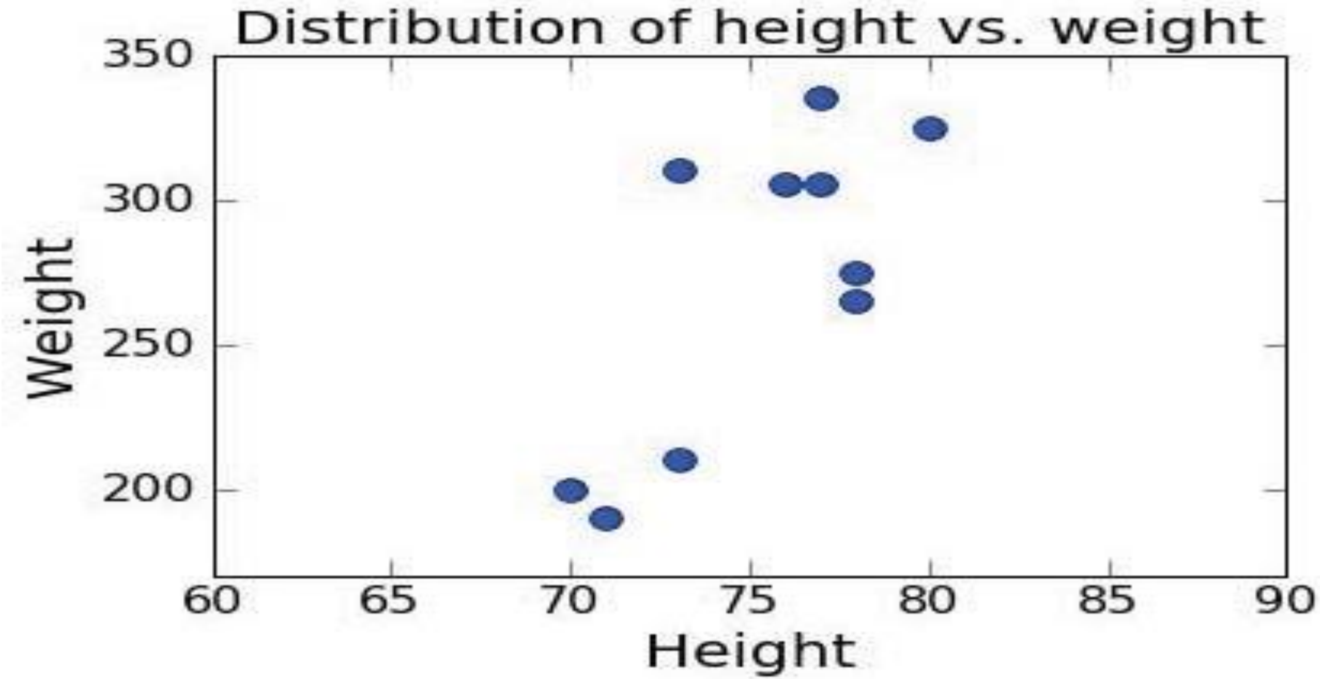
Two Variations

- Supervised: given a set of feature f of label pairs, find a rule that predicts the label associated with a previously unseen input.
- Unsupervised: given a set of feature vectors (without labels) group them into “natural clusters” (or create labels for groups)

Some examples of classifying and clustering

- Here are some data on the students in a game
  - Name, height, weight
  - Labeled by type of position
- Team 1:
  - john = ['john', 70, 200]
  - jane = ['jane', 73, 210]
  - jackie = ['jackie', 78, 265]
  - jon = ['jon', 71, 190]
  - bennett = ['bennett', 78, 275]
- Team 2:
  - jake = ['jake', 77, 335]
  - tonny = ['tonny', 80, 325]
  - candice = ['candice', 73, 310]
  - fred = ['fred', 77, 305]
  - annie = ['annie', 76, 305]

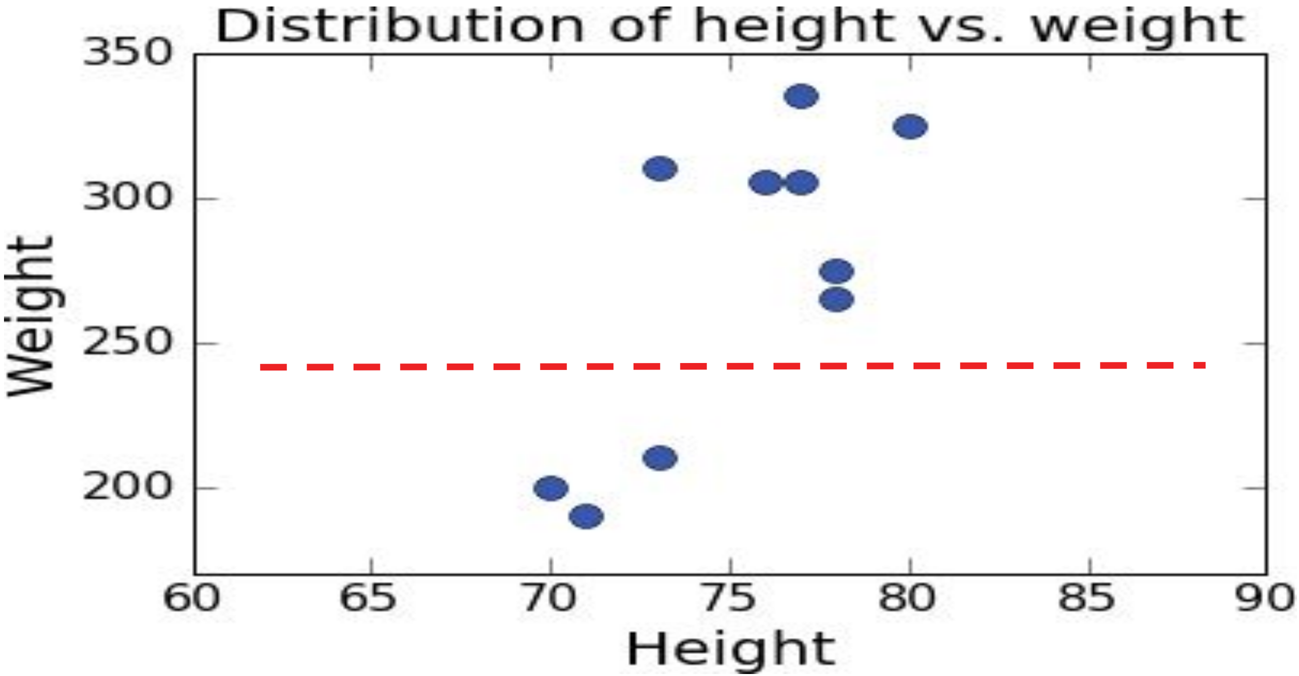
Unlabelled data



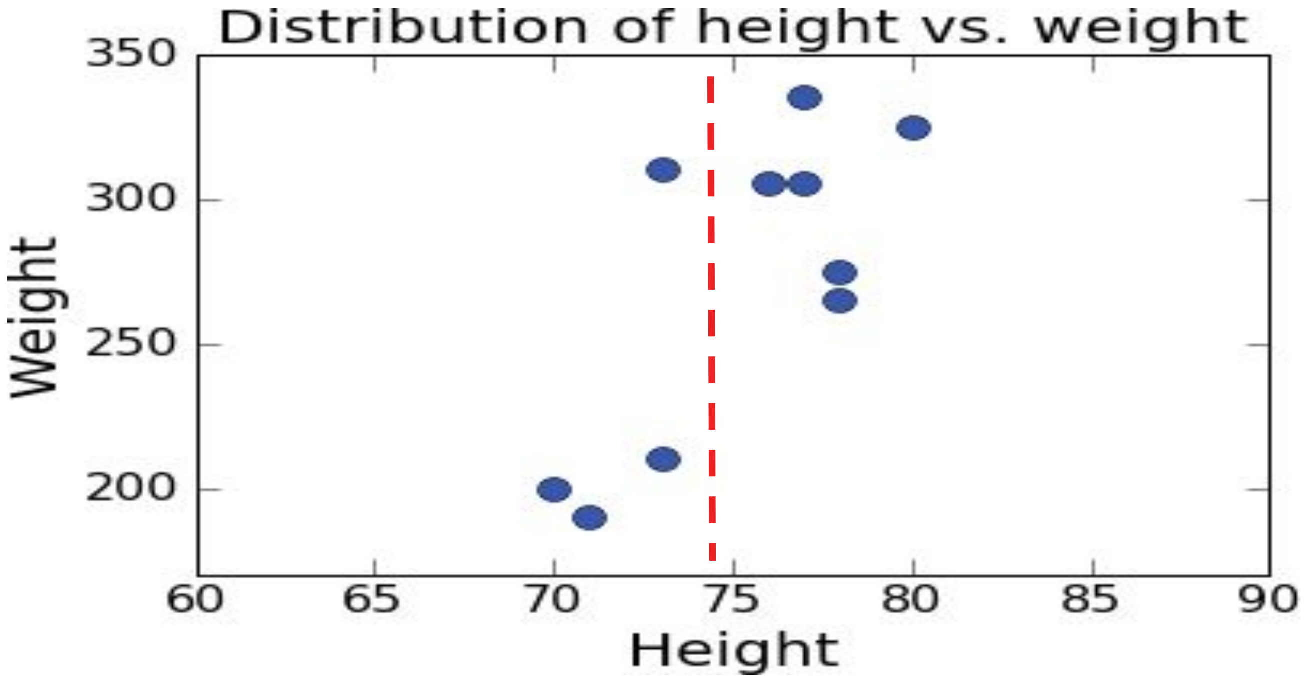
Clustering examples into groups

- Want to decide on “similarity” of examples, with goal of separating into distinct, “natural”, groups
  - Similarity is a distance measure
- Suppose we know that there are k different groups in our training data, but don't know labels (here k = 2)
  - Pick k samples (at random?) as exemplars
  - Cluster remaining samples by minimizing distance between samples in same cluster (objective function) − put sample in group with closest exemplar
  - Find median example in each cluster as new exemplar
  - Repeat until no change

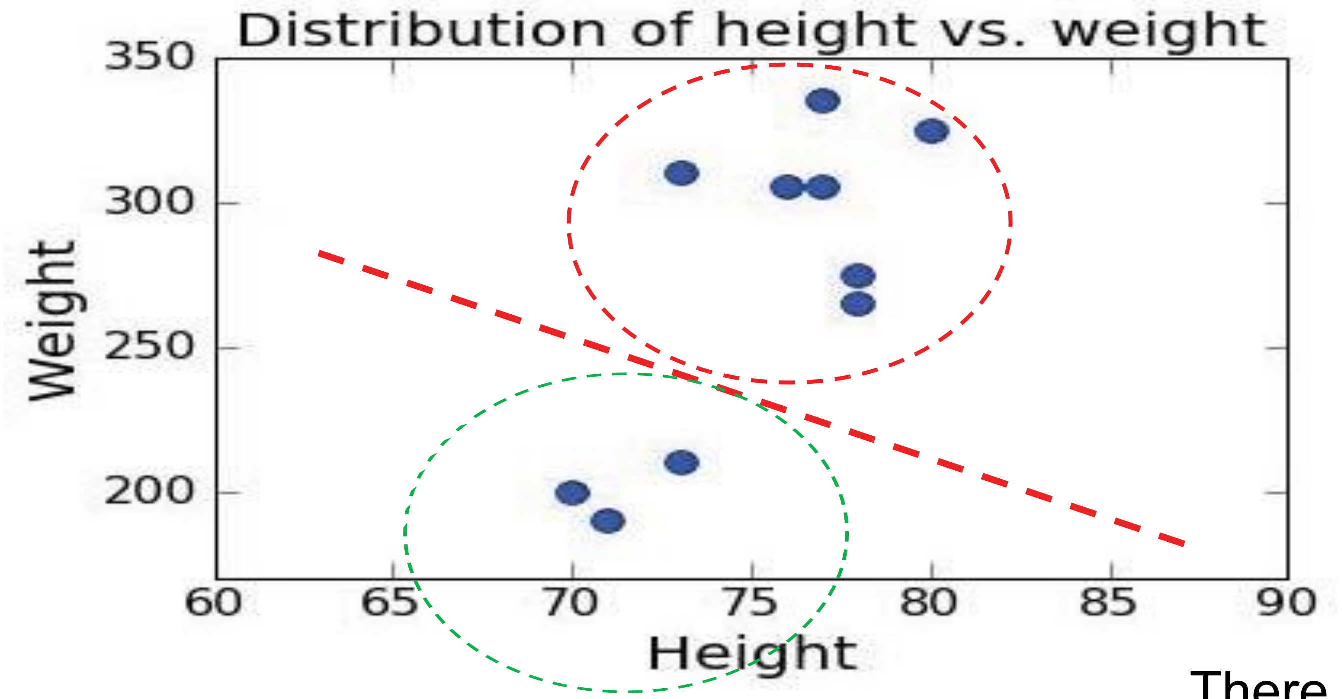
Similarity based on Weight



Similarity based on Height

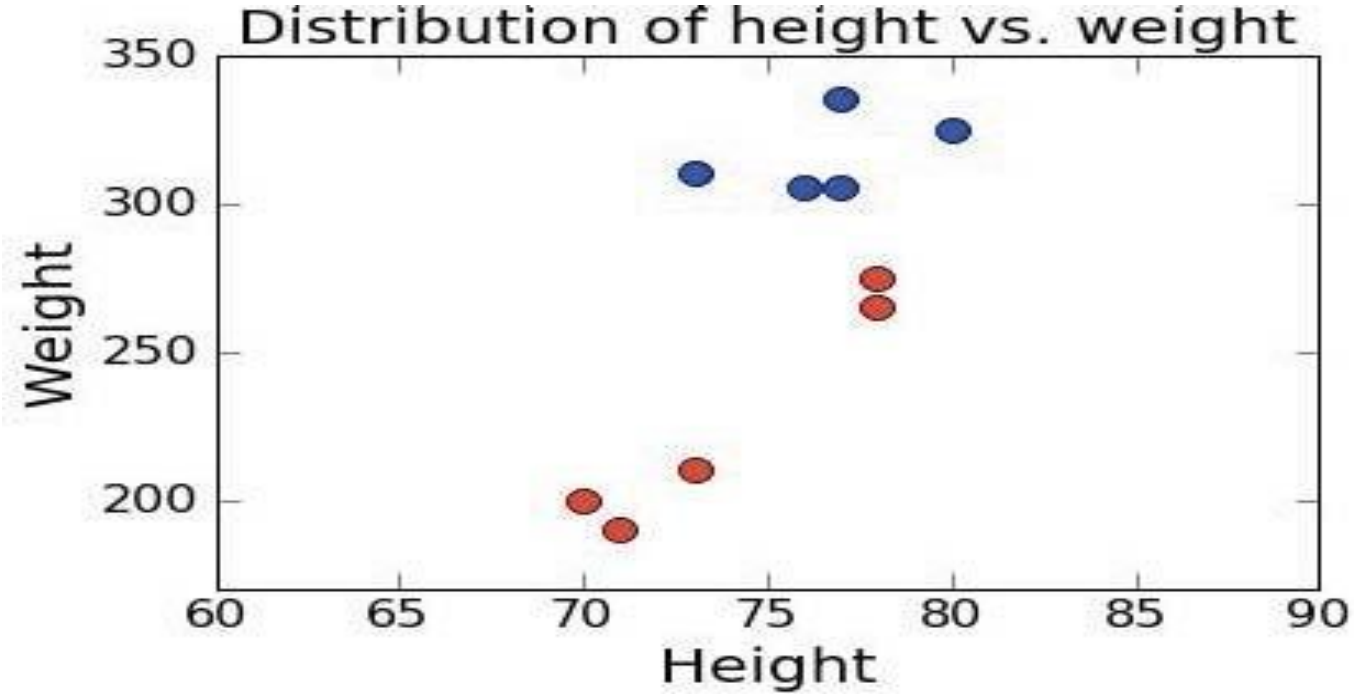


Cluster into two groups using both attributes



There are two different classes of objects, then here are best clusters

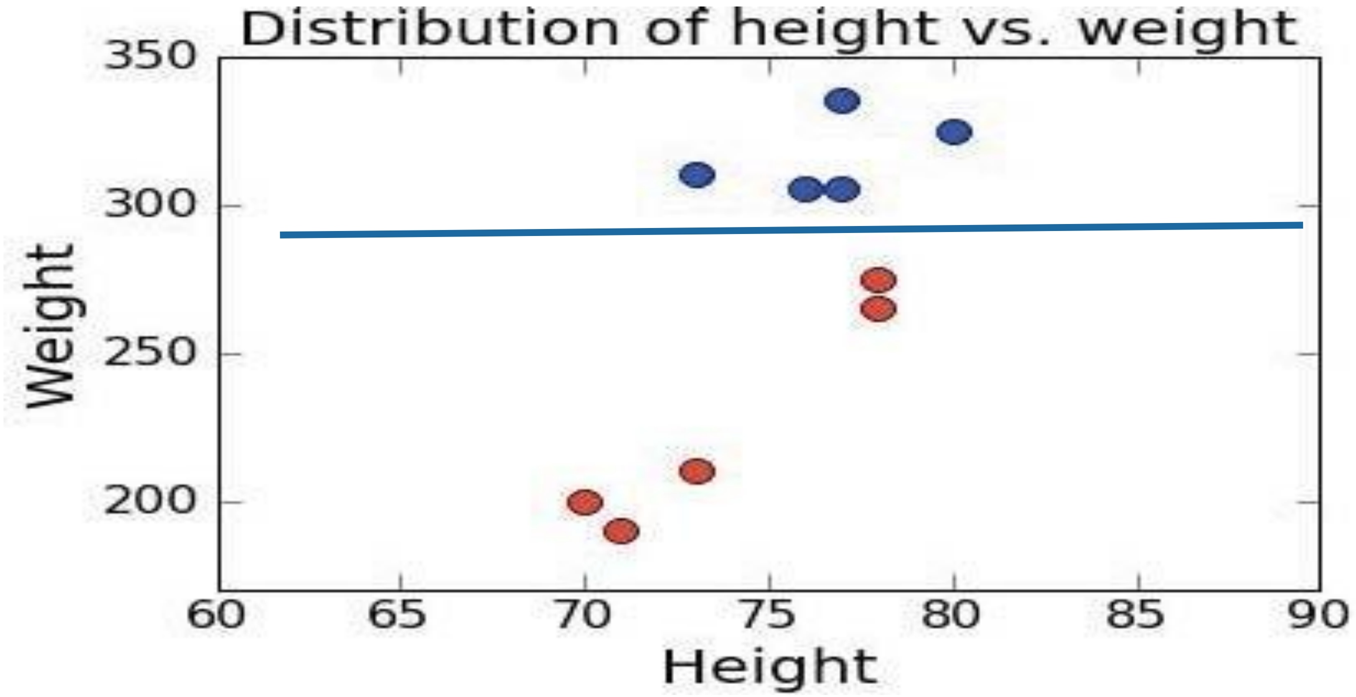
Suppose data was Labeled



Finding Classifier Surfaces

- Given labeled groups in feature space, want to find subsurface in that space that separates the groups
  - Subject to constraints on complexity of subsurface
- In this example, have 2D space, so find line (or connected set of line segments) that best separates the two groups
- When examples well separated, this is straightforward.
- When examples in labeled groups overlap, may have to trade off false positives and false negatives

Suppose the Data was Labelled

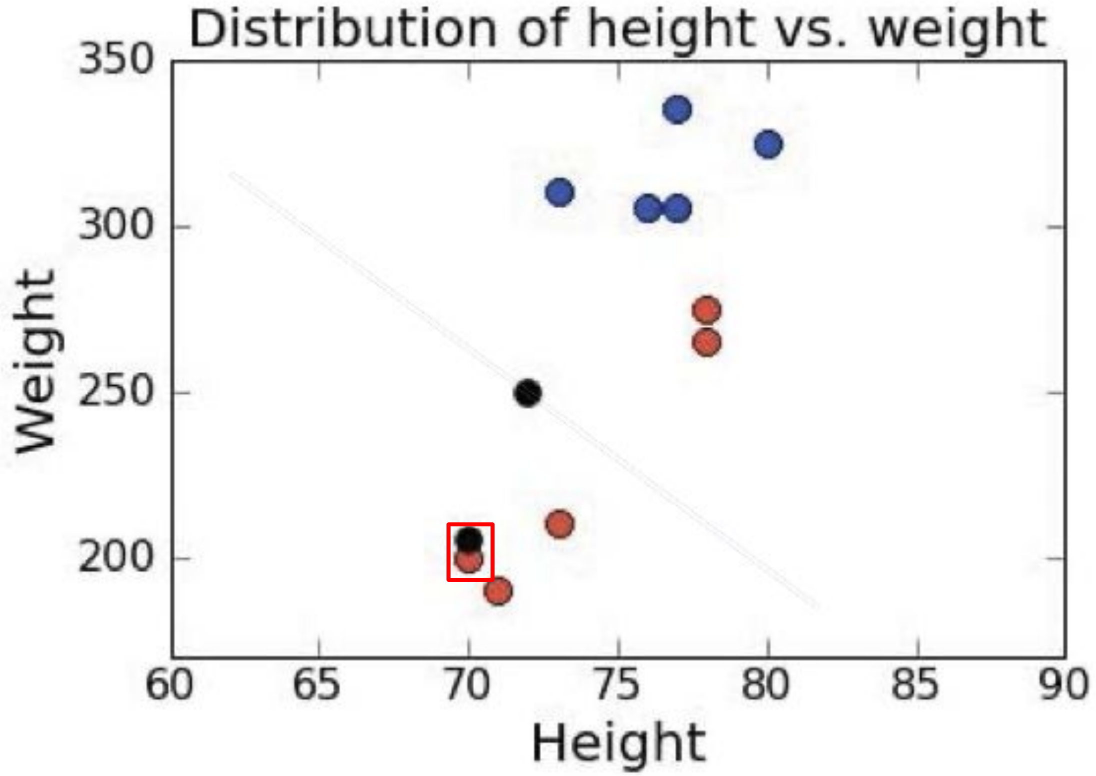


Given we have labeled data, then an obvious separator of two groups is shown

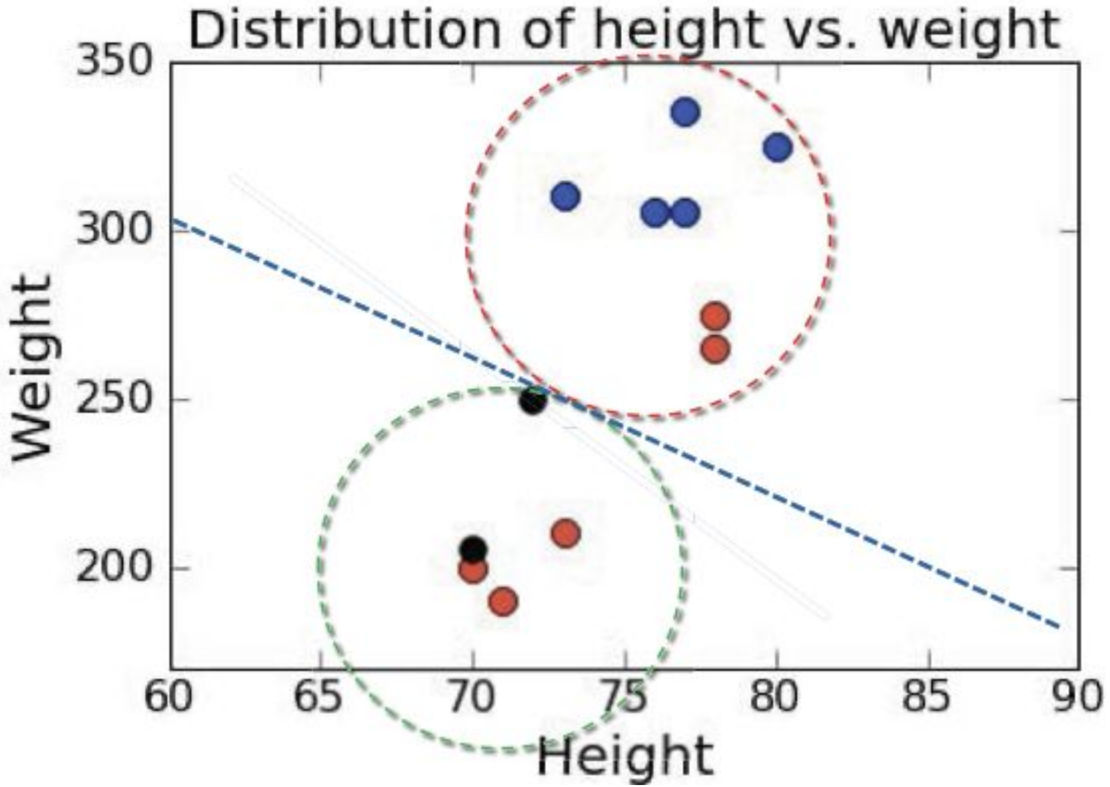
Adding some new data

- Suppose we have learned to separate **Team 1** versus **Team 2**
- Now we have members, and want to use model to decide if they are more like Team 1 or Team 2
    - ken = ['ken',72,250]
    - damian = ['damian',70, 205]

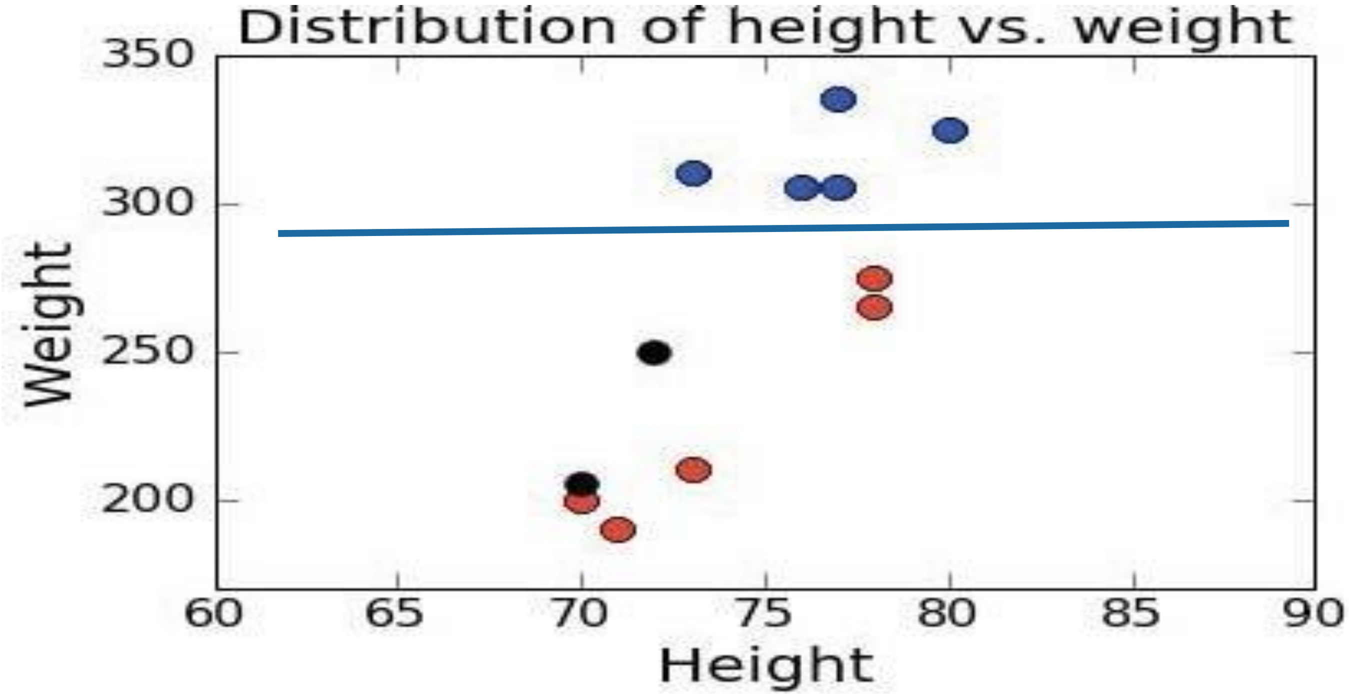
Adding some new data



Clustering using Unlabeled data



Classified using Labeled Data



Machine Learning Methods

- We will see some examples of machine learning methods:
  - Learn models based on unlabeled data, by clustering - training data into groups of nearby points
  - Resulting clusters can assign labels to new data
- Learn models that separate labeled groups of similar data from other groups by classification
  - May not be possible to perfectly separate groups, without “overfitting”
  - But can make decisions with respect to trading off “false positives” versus “false negatives”
  - Resulting classifiers can assign labels to new data

Machine Learning Methods Need

- Training data and evaluation method use to show success
- Representation of the features
  - How do we represent each instance (average speed..??)
- Distance metric for feature vectors
  - How to measure distance between features (decide whats close and what's not)
- Objective function and constraints
  - Take features and build an objective function that you want to minimize to find what is the best cluster to use..
- Optimization method for learning the model

Feature Representation

- Features never fully describe the situation
  - “All models are wrong, but some are useful.” − George Box
- Feature engineering
  - Represent examples by feature vectors that will facilitate generalization
  - Suppose use 100 examples from past to predict, at the start of the subject, which students will get an A in a course.
  - Some features surely helpful, e.g., GPA, prior programming experience (not a perfect predictor)
  - Others might cause me to overfit, e.g., birth month, eye color
- Want to maximize ratio of useful input to irrelevant input
  - Signal−to−Noise Ratio (SNR)
  - Maximise those features that carry the most information

An Example

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	No. of legs	Reptile
Cobra	True	True	True	True	0	Yes

- Initial model
    - Not enough information to generalize

An Example

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	No. of legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes

- Initial model
  - Egg laying, has scales, is poisonous, cold blooded, no legs

An Example

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	No. of legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes

- Initial model
  - Egg laying, has scales, is poisonous, cold blooded, no legs
- Current model
  - Has scales, Cold blooded, No legs

Boa doesn't fit the model - but is labelled as reptile - need to refine the model.

An Example

Features

Label

Name	Egg-layin g	Scales	Poisonous	Cold-blooded	No. of legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes

- Current model
  - Has scales, Cold blooded, No legs

An Example

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	No. of legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No

- Current model
  - Has scales, Cold blooded, No legs

An Example

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	No. of legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes

- Current model
  - Has scales, Cold blooded, Has 0 to 4 legs

Alligator doesn't fit the model - but is labelled as reptile - need to refine the model.

An Example

Features Label

Name	Egg-la ying	Scales	Poisonous	Cold-blooded	No. of legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	True	4	No

- Current model
  - Has scales, Cold blooded, Has 0 to 4 legs

An Example

Features

Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	No. of legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	True	4	No
Salmon	True	True	False	True	0	No
Python	True	True	False	True	0	Yes

Current model - Has scales, Cold blooded, Has 0 to 4 legs

No (easy) way to add to rule that will correctly classify salmon and python (have identical feature values)

An Example

Features Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	No. of legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	True	4	No
Salmon	True	True	False	True	0	No
Python	True	True	False	True	0	Yes

Good model - Has scales, Cold blooded

Not perfect, but no false negatives (anything classified as not reptile is correctly labeled); some false positives (may incorrectly label some animals as reptile)

Need to Measure Distance between Features

- Feature engineering:
  - Deciding which features to include and which are merely adding noise to classifier
  - Defining how to measure distances between training examples (and ultimately between classifiers and new instances)
  - Deciding how to weight relative importance of different dimensions of feature vector, which impacts definition of distance.

Measuring Distance Between Animals

- We can think of our animal examples as consisting of four *binary features* and one *integer feature*.
- One way to learn to separate reptiles from non-reptiles is to measure the distance between pairs of examples, and use that:
  - To cluster nearby examples into a common class (unlabeled data), or
  - To find a classifier surface in space of examples that optimally separates different (labeled) collections of examples from other collections

Measuring Distance Between Animals

- We can think of our animal examples as consisting of four binary features and one integer feature.
- One way to learn to separate reptiles from non-reptiles is to measure the distance between pairs of examples, and use that:
  - *Can convert examples into feature vectors*

rattlesnake = [1,1,1,1,0]
boa constrictor = [0,1,0,1,0]
dart frog = [1,0,1,0,4]

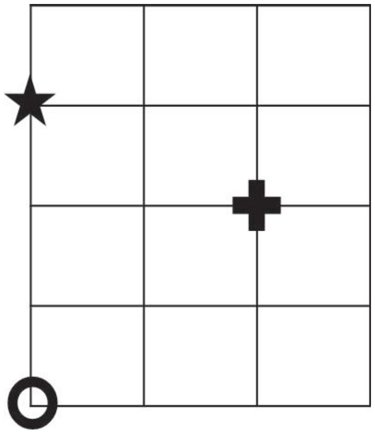
Minkowski Metric

$$dist(X1, X2, p) = \left(\sum_{k=1}^{len} abs(X1_k - X2_k)^p\right)^{\frac{1}{p}}$$

p = 1: Manhattan Distance

p = 2: Euclidean Distance

Typically use Euclidean metric; Manhattan may be appropriate if different dimensions are not comparable



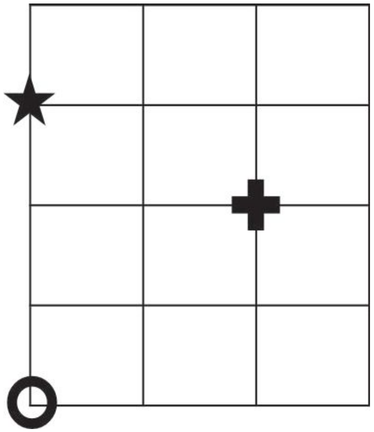
Minkowski Metric

$$dist(X1, X2, p) = (\sum_{k=1}^{len} abs(X1_k - X2_k)^p)^{\frac{1}{p}}$$

p = 1: Manhattan Distance

p = 2: Euclidean Distance

Typically use Euclidean metric; Manhattan may be appropriate if different dimensions are not comparable



Need to measure distances between feature vectors

Is circle closer to star or cross?

- Euclidean distance
  - Cross – 2.8
  - Star – 3
- Manhattan Distance
  - Cross – 4
  - Star – 3

Euclidean Distance Between Animals

rattlesnake = [1,1,1,1,0]
boa constrictor = [0,1,0,1,0]
dart frog = [1,0,1,0,4]

	rattlesnake	Boa constrictor	dart frog
rattlesnake	-	1.414	4.243
boa constrictor	1.414	-	4.472
dart frog	4.243	4.472	-

Using Euclidean distance, rattlesnake and boa constrictor are much closer to each other, than they are to the dart frog

Add an Alligator

- alligator = Animal('alligator', [1,1,0,1,4])
- animals.append(alligator)
- compareAnimals(animals, 3)

rattlesnake = [1,1,1,1,0]
boa constrictor = [0,1,0,1,0]
dart frog = [1,0,1,0,4]
Alligator = [1,1,0,1,4]

Add an alligator

	rattlesnake	Boa constrictor	dart frog	alligator
rattlesnake	-	1.414	4.243	4.123
boa constrictor	1.414	-	4.472	4.123
dart frog	4.243	4.472	-	1.732
alligator	4.123	4.123	1.732	-

Alligator is closer to dart frog than to snakes − why?

- Alligator differs from frog in 3 features, alligator from boa in only 2 features
- But scale on “legs” is from 0 to 4, on other features is 0 to 1
- “legs” dimension is disproportionately large

Using Binary Features

rattlesnake = [1,1,1,1,0]
boa constrictor = [0,1,0,1,0]
dart frog = [1,0,1,0,1]
Alligator = [1,1,0,1,1]

	rattlesnake	Boa constrictor	dart frog	alligator
rattlesnake	-	1.414	1.732	1.414
boa constrictor	1.414	-	2.236	1.414
dart frog	1.732	2.236	-	1.732
alligator	1.414	1.414	1.732	-

- Now the alligator is closer to snakes than it is to dart frog
  - Makes more sense
- Feature Engineering Matters

Supervised versus unsupervised learning

- When given unlabeled data, try to find clusters of examples near each other
  - Use centroids of clusters as definition of each learned class
  - New data assigned to closest cluster
- When given labeled data, learn mathematical surface that ”best” separates labeled examples, subject to constraints on complexity of surface (don't over fit)
  - New data assigned to class based on portion of feature space carved out by classifier surface in which it lies.

Issues of concern when learning models

- Learned models will depend on:
  - Distance metric between examples
  - Choice of features to use in the vector.
  - Constraints on complexity of model
    - Specified number of clusters
    - Complexity of separating surface
    - Want to avoid overfitting problem (each example is its own cluster, or a complex separating surface)

Summary

- Machine learning methods provide a way of building models of processes from data sets
  - Supervised learning uses labeled data, and creates classifiers that optimally separate data into known classes
  - Unsupervised learning tries to infer latent variables by clustering training examples into nearby groups
- Choice of features influences results
- Choice of distance measurement between examples influences results

References

1. Introdution to Computational thinking and datascience -https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0002-introduction-to-computational-thinking-and-data-science-fall-2016/lecture-slides-and-files/index.htm
2. Data normalisation - https://towardsdatascience.com/understand-data-normalization-in-machine-learning-8ff3062101f0

Thank you!