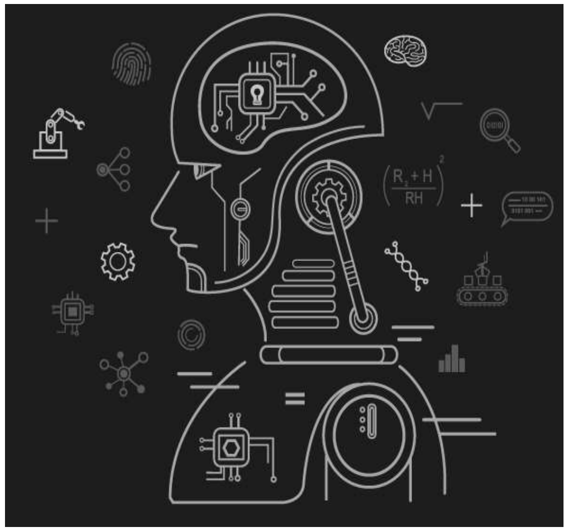


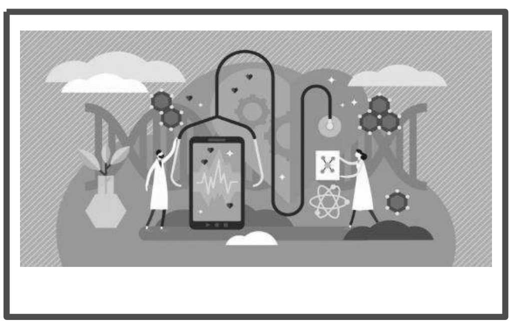
Introduction to Machine Learning

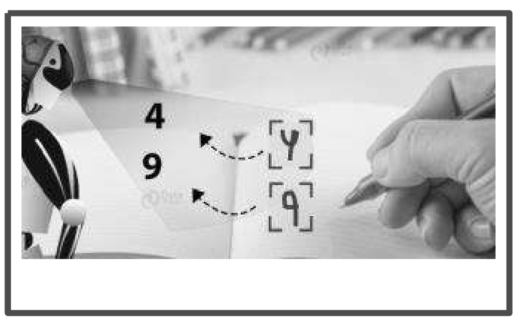
Outline of Presentation

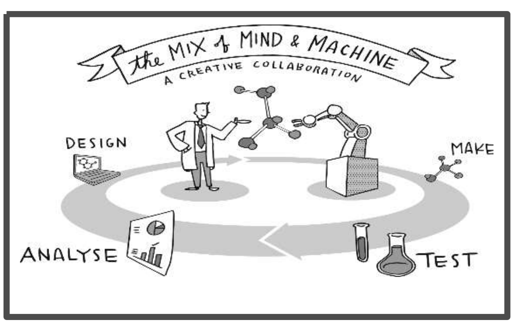
What is Machine Learning?

- How are Things Learnt
- Basic Paradigm
- Some Examples of Classifying and Clustering
- Machine Learning Methods
- Machine Learning Methods Requirements
- Feature Representation
- Issues of Concern when Model Learns
- Clustering Approaches
- Classification Approaches
- Confusion Matrices
- Performance Parameters of the Model
- Summary

Few Examples of Machine Learning Application in Real World







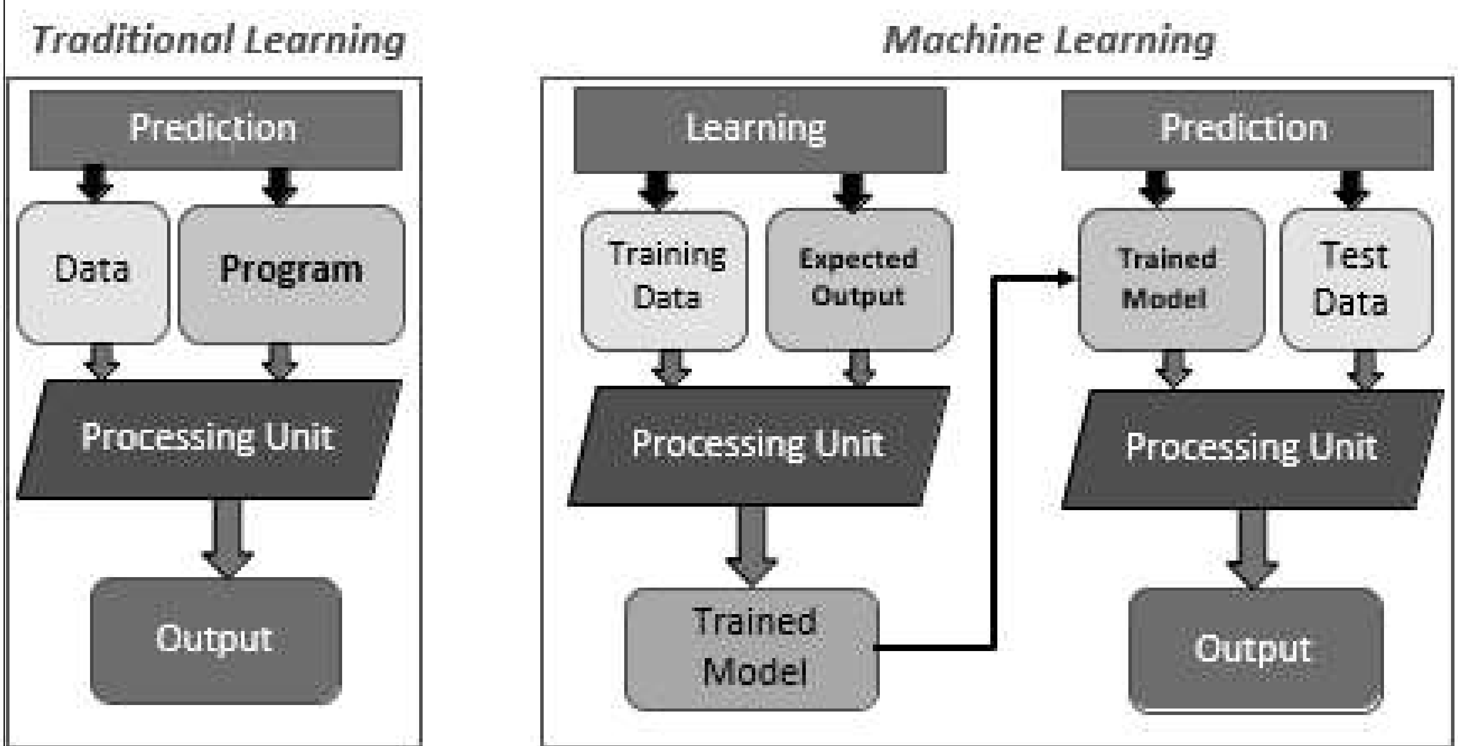
What is Machine Learning?

"Field of study that gives computers the ability to learn without being explicitly programmed"

***On February 24, 1956**, Arthur Samuel's Checkers program was developed for play on the IBM 701, In 1962, Self-proclaimed checkers master Robert Nealey played the game on an IBM 7094 computer. The computer won. It is still considered a milestone for artificial intelligence, as an example of the capabilities of an electronic computer.*

The Samuel Checkers-playing Program was among the world's first successful self-learning programs, and as such a very early demonstration of the fundamental concept of artificial intelligence.

Traditional Learning vs. Machine Learning



How are Things Learnt

• Memorization

Declarative knowledge • Individual facts are accumulated over time

• However, the memorization of facts is limited by ✓Time to observe facts
✓Memory to store facts

• Generalization

Imperative knowledge • Deduce new facts from old facts

• Limited by accuracy of deduction process • Essentially a predictive activity

• Assumes that the past predicts the future

• Infer useful information from implicit patterns in data

Basic Paradigm

Create a training set

Spatial Deviations relative to mass displacements of spring

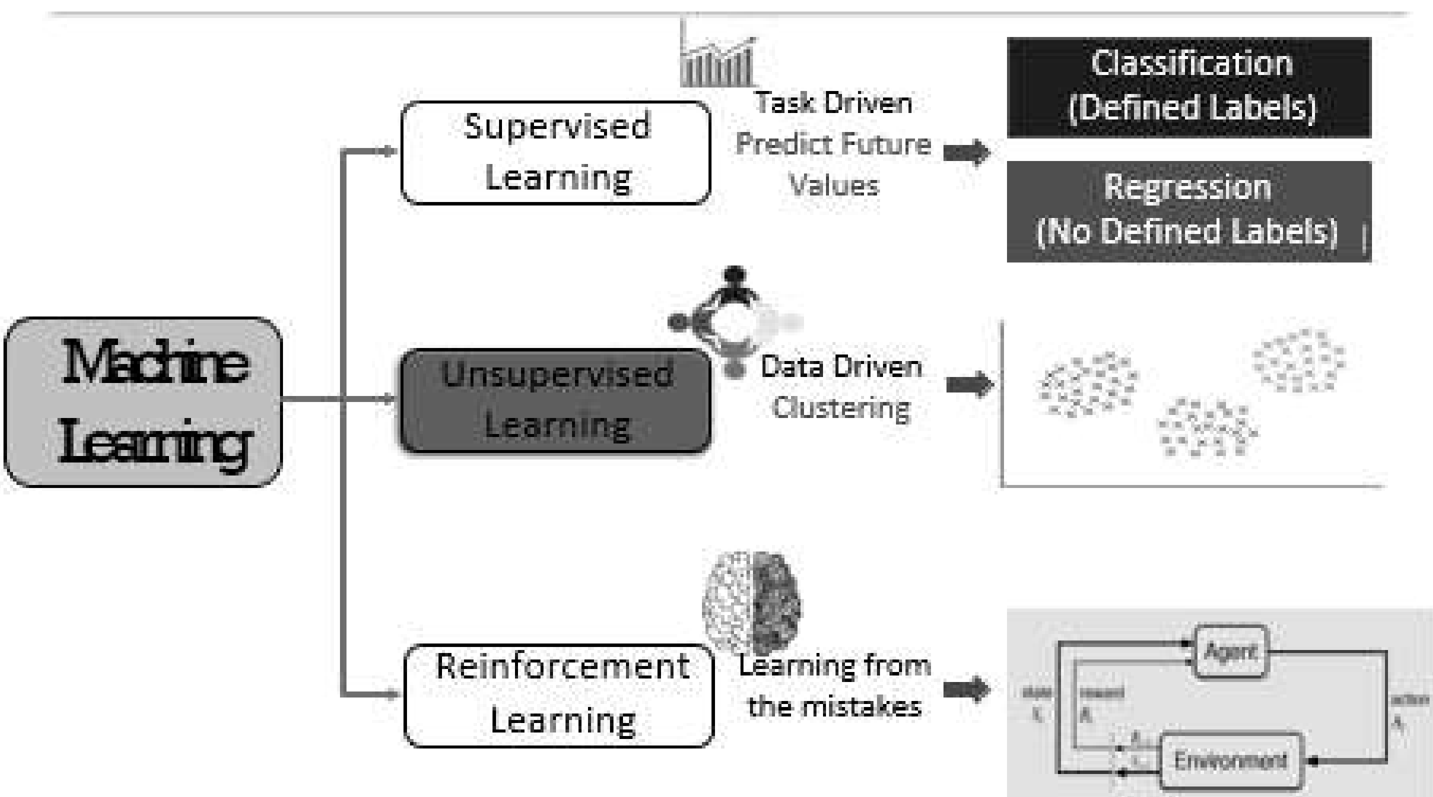
- Infer something from about the process that generated that data.

Fit polynomial curve using linear regression

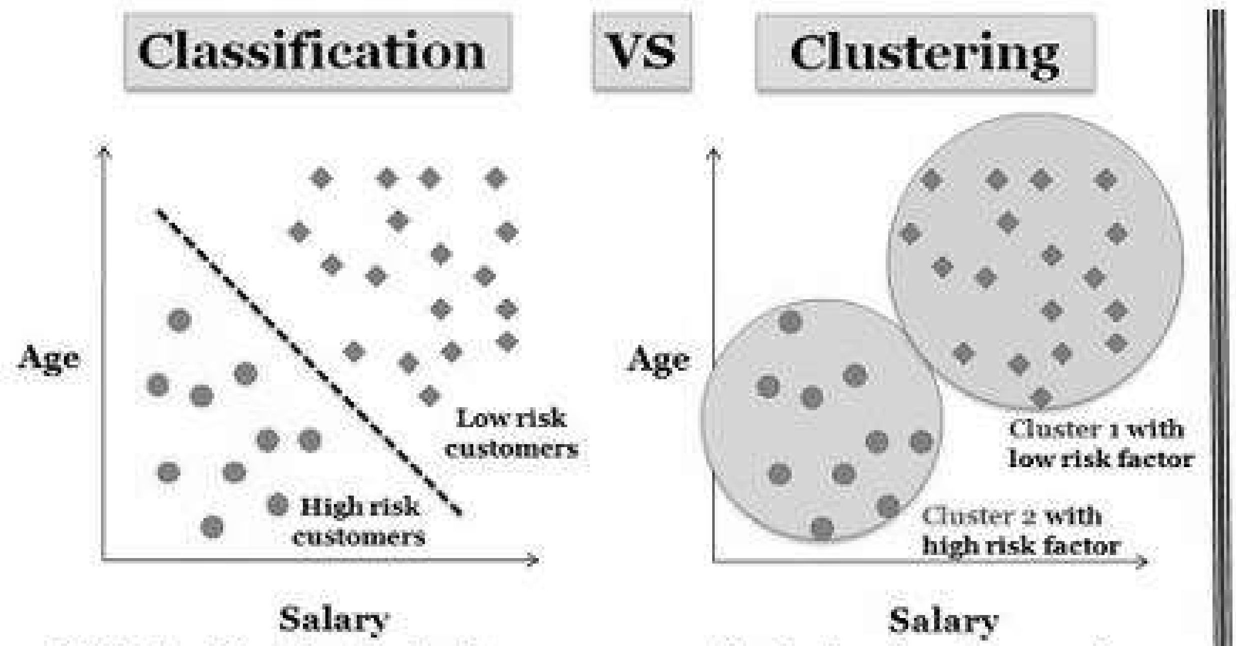
- Use the learning to make predictions about previously unseen data, i.e. the testing set

Predict displacements for other weights

Machine Learning - Types



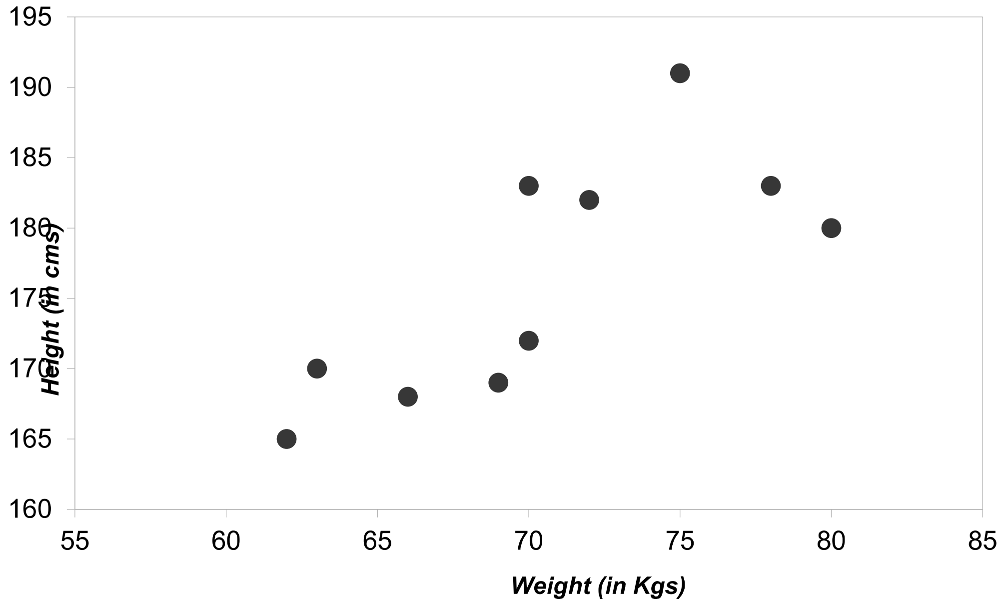
Some Examples of Classifying and Clustering



Risk classification for the loan payees on the basis of customer salary

- Lets consider an example on the Indian World Cup Cricket Squad 2011
  - Labelled by type of role in the team
  - Each player is characterized by Name, weight (in kgs), height (in cms)
- Batsman:
  - Sehwag = [‘sehwag’, 63, 170]
  - Sachin = [‘sachin’ , 62, 165]
  - Gambhir = [‘gambhir’ , 66 ,168]
  - Virat = [‘virat’, 69, 169]
  - Dhoni = [‘dhoni’, 70, 172]
- Bowlers:
  - Zaheer = [‘zaheer’, 78, 183]
  - Sreesanth = [‘Sreesanth’, 80, 180]
  - Munaf = [‘munaf’, 75, 191]
  - Harbhajan = [‘harbhajan’, 72, 182]
  - Nehra = [‘nehra’, 70, 182]

Unlabelled Data

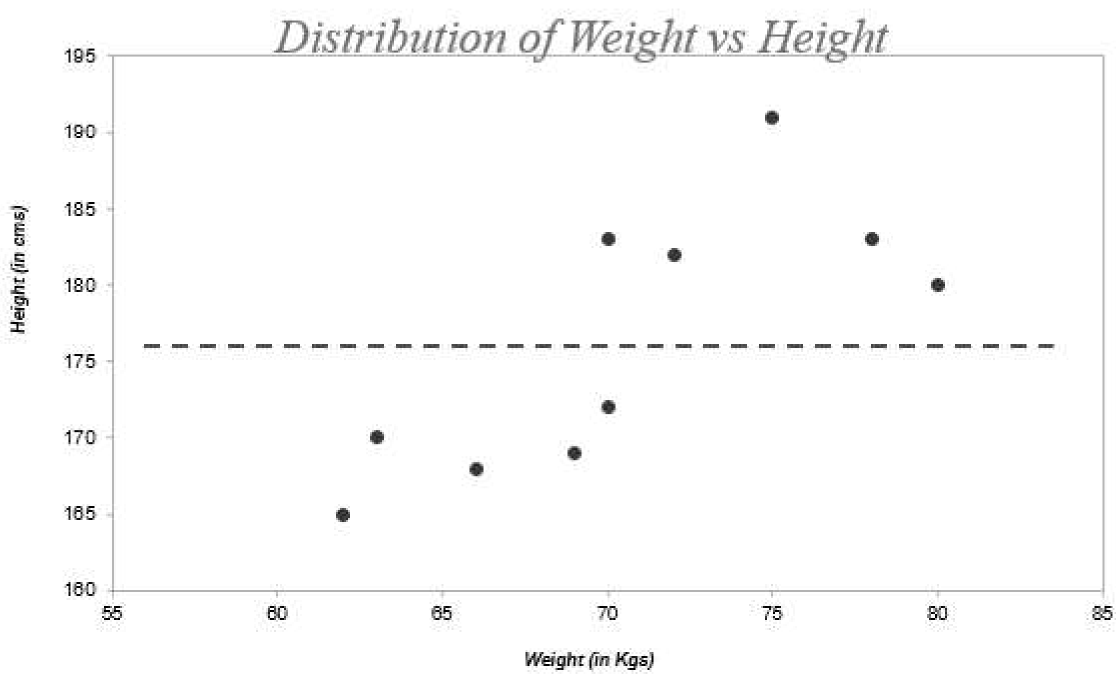


Clustering Examples into Groups

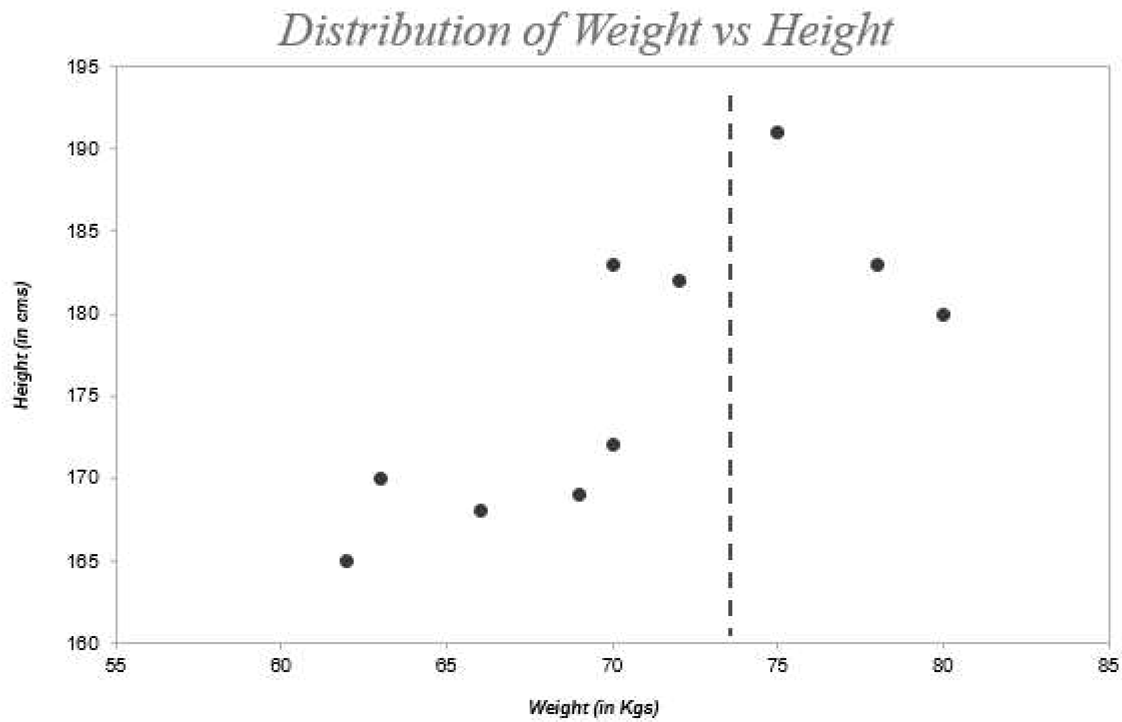
- With a goal of separating the samples into distinct groups, the similarity between sample spaces is assessed.
  - Similarity is a distance measure
- Suppose there are "k" different groups in our training data, but don't know their labels (here k=2)
  - Pick k samples (at random) from the training set of data.

- Minimize the distance between remaining samples from the selected data points.
- Find a new median sample data point in each cluster.
- Repeat until no change is observed.

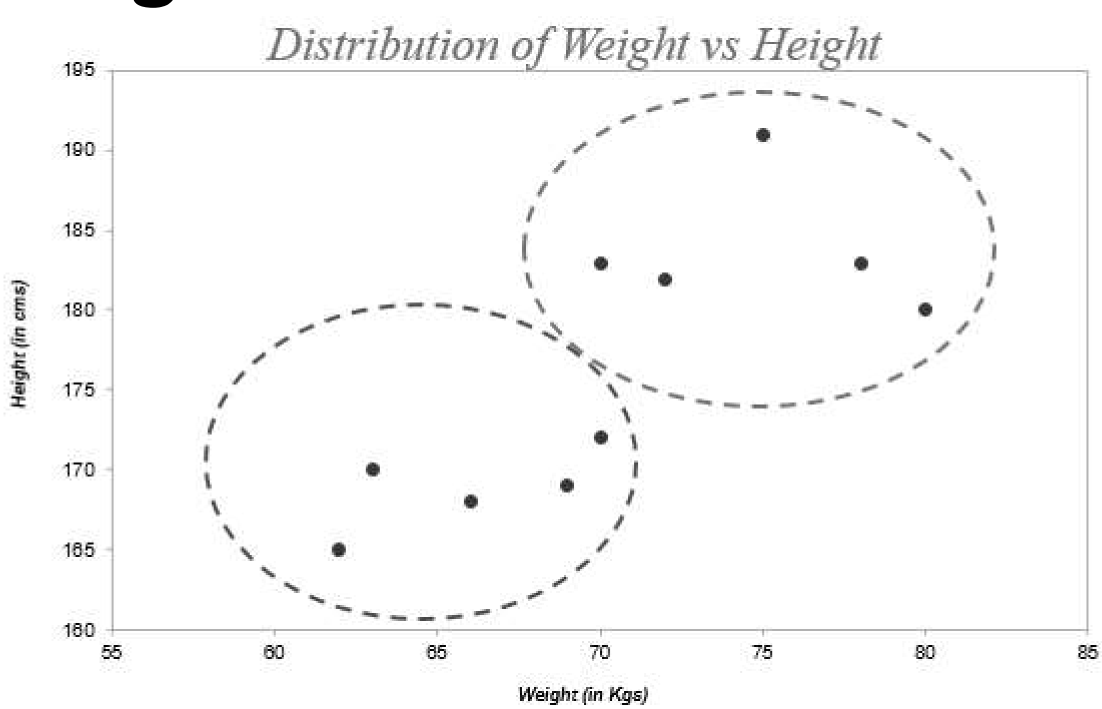
Similarity Based on Height



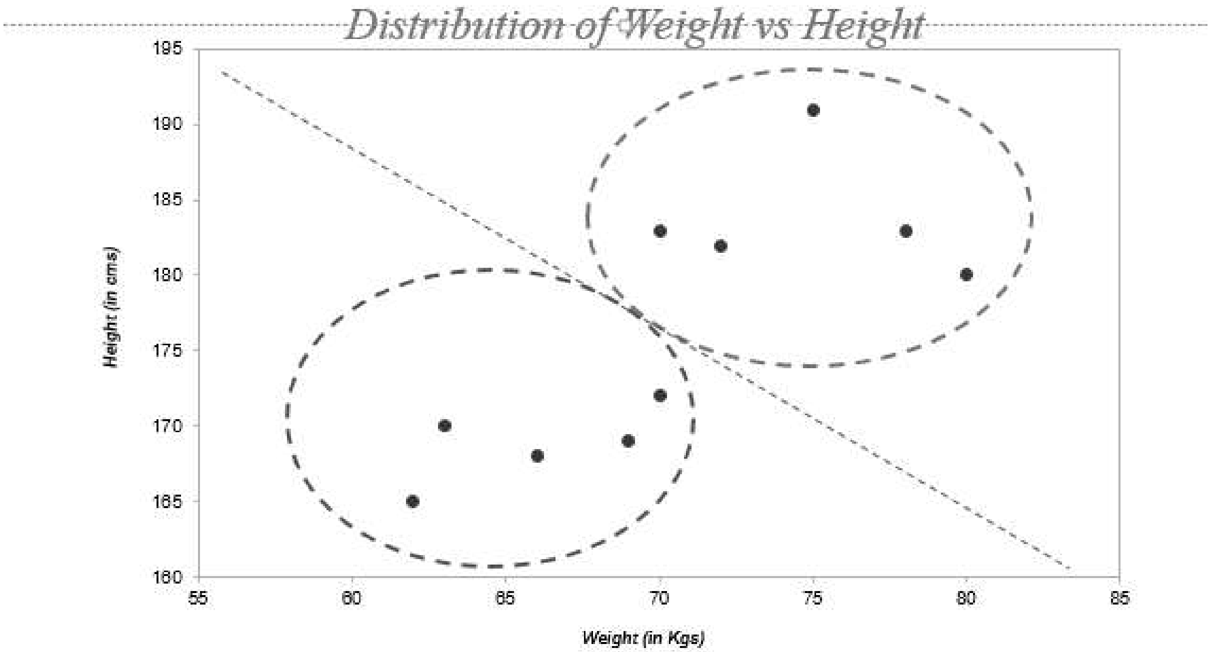
Similarity Based on Weight



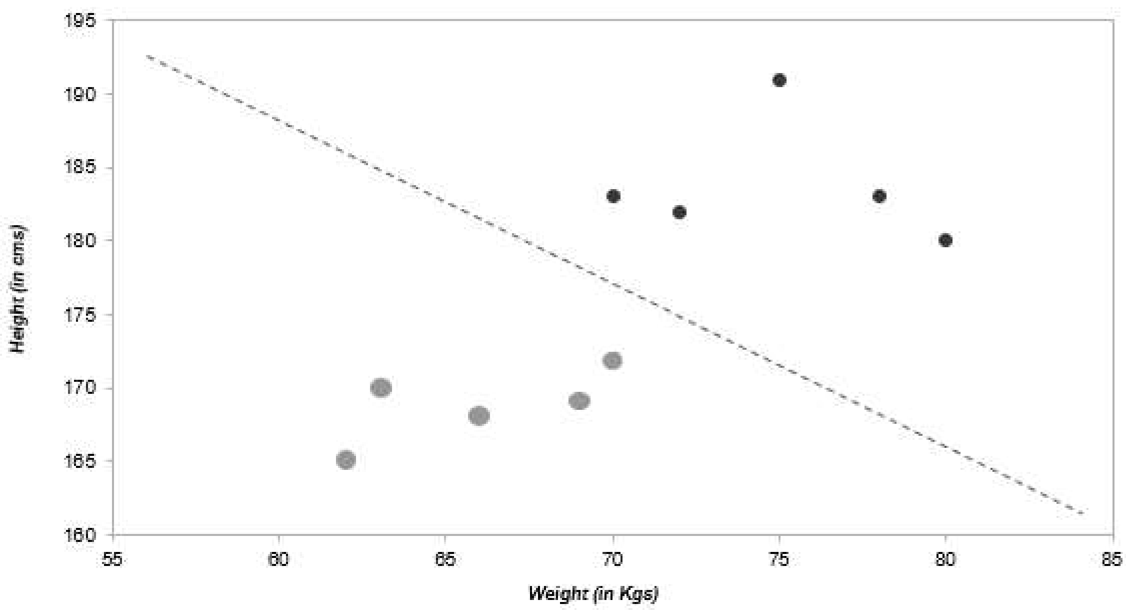
Cluster into Two Groups Using Both Attributes



Cluster into Two Groups Using Both Attributes



Suppose Data was Labelled



Finding Classifier Surfaces

- Given labelled groups in feature space, it is desirable to determine a surface which is capable of separating the data points of that group.
- Subject to constraints on complexity of surface
  - In this example, 2D feature space is available, so a LINE (or connected set of line segments) best separates the points in training set into two separate groups.
  - When data points are well distinguished, then this is straightforward and easy to achieve a surface.
  - When data points in the labelled group overlaps, then a need for trade off arises, and, hence, false positives and false negatives come into play.

Adding Some New Data

- Suppose we have learned to separate batsmen from bowlers
- Now we are given some new players, and want to use model to decide if they are more like batsmen or bowlers

  yuvraj = [‘yuvraj’, 72, 167]
  shami = [‘shami’, 75, 173]

Machine Learning Methods

- We will see some examples of machine learning methods:
- Learn models based on unlabelled data,
    - By clustering training data into groups of nearby points
    - Resulting clusters can assign labels to new data
- Learn models that separates the data points with similar label to other group label
    - It may not be possible to perfectly separate the data points into distinct groups, without "over fitting".
    - Hence, decisions with respect to trade-off are undertaken, i.e. "false positives" and "false negatives" are taken into consideration.
    - Labels can be assigned to new test data using the Resulting classifiers.

All Machine Learning Methods Require.....

- Choosing training data and evaluation method
- Representation of features
- Distance metric for feature vectors
- Loss function and constraints
- Optimisation method for learning the model

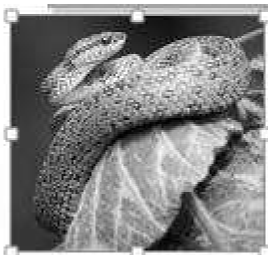
Feature Representation

- Features never fully describe the situation.

"All models are wrong, but some are useful." - George Box.

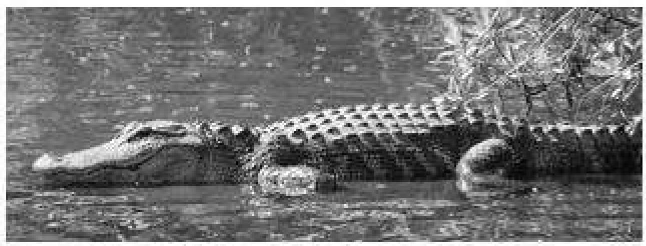
- Feature engineering: Important pieces of data that help us understand a problem. These are fed in to a Machine Learning algorithm to help it learn.
  - Represent data points by feature vectors that will facilitate generalization.
  - Suppose I want to use 100 data points from past to predict which students will get an A grade in Computer Science, at the start of the subject.
  - Some features surely helpful, e.g., GPA, prior programming experience (not a perfect predictor)
  - Others might cause an overfit scenario, e.g., birth month, eye color.
- Want to maximize ratio of useful features to irrelevant features.

Classification of Animals

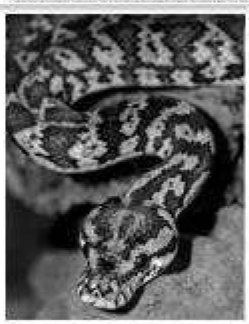

Viper


Rohu


Cobra


Crocodile


Emerald tree boas


Python


Golden Poison Frog

Example

FEATURES LABEL

Name	Egg-laying	Scales	Poisonous	Cold-blooded	#legs	Reptile
Viper	True	True	True	True	0	Yes
Cobra	True	True	True	True	0	Yes
Emerald tree boas	False	True	False	True	0	Yes
Crocodile	True	True	False	True	4	Yes
Golden Poison Frog	**True**	**False**	**True**	**False**	**4**	**No**
Rohu	True	True	False	True	**0**	**No**
Python	True	True	False	True	**0**	Yes

Now, features are as follows:

- Scales
- Cold-blooded
- Has 0 or 4 legs

Features of Rohu and Python are **exactly same,** but their **labels** are **different**
No easy way to add or remove features that help to classify correctly

Need to Measure Distances between Features

Feature engineering

- Decide which features to include in the model

(As using all features merely add noise to the learning process)

- Second step is to define distance.
  - It help to measure distances between training data points and new instances.
- Decide how to weigh relative importance of different dimensions in feature vector, which impacts definition of distance.

Measuring Distance between Animals

- In considered example, each animal consist:
    - four binary features (Egg laying, scales, poisonous, cold-blooded)
    - one integer feature (leg-0 to 4)
- One way to learn to separate reptiles from non-reptile is to measure the distance between pairs of data points (animals) , and use that:
    - To cluster nearby examples into a common class (unlabelled data),

      or

    - To find a classifier surface in feature space that optimally separates different groups of data points from other groups

Viper = [1,1,1,1,0]

Emerald tree boas = [0,1,0,1,0]

Golden Poison Frog= [1,0,1,0,4]

Minkowski Difference

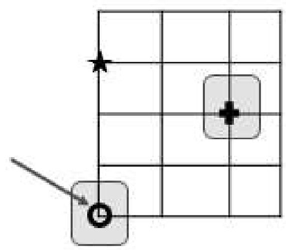
$$dist(X1, X2, p) = \sum_{k=1}^{len} abs\ (X1_k - X2_k)^p)^{1/p}$$

p = 1: Manhattan Distance
p = 2: Euclidean Distance

Need to measure distance between feature vectors

Is circle closer to star or cross?

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



Typically Euclidean distance is used; Manhattan may be appropriate if different dimensions are not comparable

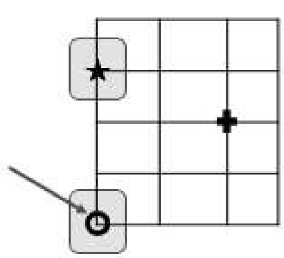
$$dist(X1, X2, p) = \sum_{k=1}^{len} abs\ (X1_k - X2_k)^p)^{1/p}$$

p = 1: Manhattan Distance
p = 2: Euclidean Distance

Need to measure distance between feature vectors

Is circle closer to star or cross?

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



Typically Euclidean distance is used; Manhattan may be appropriate if different dimensions are not comparable

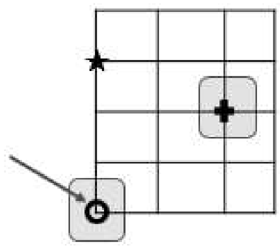
$$dist(X1, X2, p) = \sum_{k=1}^{len} abs\,(X1_k - X2_k)^p)^{1/p}$$

p = 1: Manhattan Distance
p = 2: Euclidean Distance

Need to measure distance between feature vectors

Is circle closer to star or cross?

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



Typically Euclidean distance is used; Manhattan may be appropriate if different dimensions are not comparable

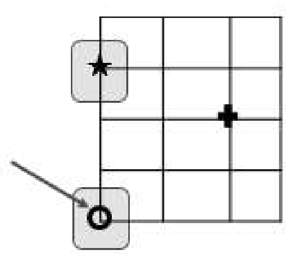
$$dist(X1, X2, p) = \sum_{k=1}^{len} abs\,(X1_k - X2_k)^p)^{1/p}$$

p = 1: Manhattan Distance
p = 2: Euclidean Distance

Need to measure distance between feature vectors

Is circle closer to star or cross?

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



Typically Euclidean distance is used; Manhattan may be appropriate if different dimensions are not comparable

Euclidean Distance between Animals

Viper = [1,1,1,1,0]
Emerald tree boas = [0,1,0,1,0]
Golden Poison Frog= [1,0,1,0,4]

=

	Viper	Emerald Tree Boas	Golden Poison Frog
Viper	--	1.414	4.243
Emerald Tree Boas	1.414	--	4.472
Golden Poison Frog	4.243	4.472	--

From the Table:
Both snakes are reasonably close to each other.
While golden poison frog is fairly distance away from them

Add a new animal

crocodile = Animal (‘crocodile’, [1,1,0,1,4])
Animals.append(crocodile)
compareAnimals(animals, 3)

	Viper	Emerald Tree Boas	Golden Poison Frog	Crocodile
Viper	--	1.414	4.243	4.123
Emerald Tree Boas	1.414	--	4.472	4.123
Golden Poison Frog	4.243	4.472	--	1.732
Crocodile	4.123	4.123	1.732	

Two snakes are closer, but crocodile is closer to golden poison

- Crocodile differs from frog in 3 features, from boas in only 2 features
- Other features are binary: true (1) or false (0)
- While “legs” dimension is disproportionately large consist from 0-4

Using Binary Features

In binary features, feature "leg" : either legs(1) or no legs (0)

Viper = [1,1,1,1,0]
Emerald tree boas = [0,1,0,1,0]
Golden Poison Frog= [1,0,1,0,1]
Crocodile = [1,1,0,1,1]

	Viper	**Emerald Tree Boas**	**Golden Poison Frog**	**Crocodile**
Viper	--	1.414	1.732	1.414
Emerald Tree Boas	1.414	--	2.236	1.414
Golden Poison Frog	1.732	2.236	--	1.732
Crocodile	1.414	1.414	1.732	

Now crocodile is closer to snakes than it is to dart frog --- make more sense

Features Engineering Matters !

Issues of Concern When Model Learns

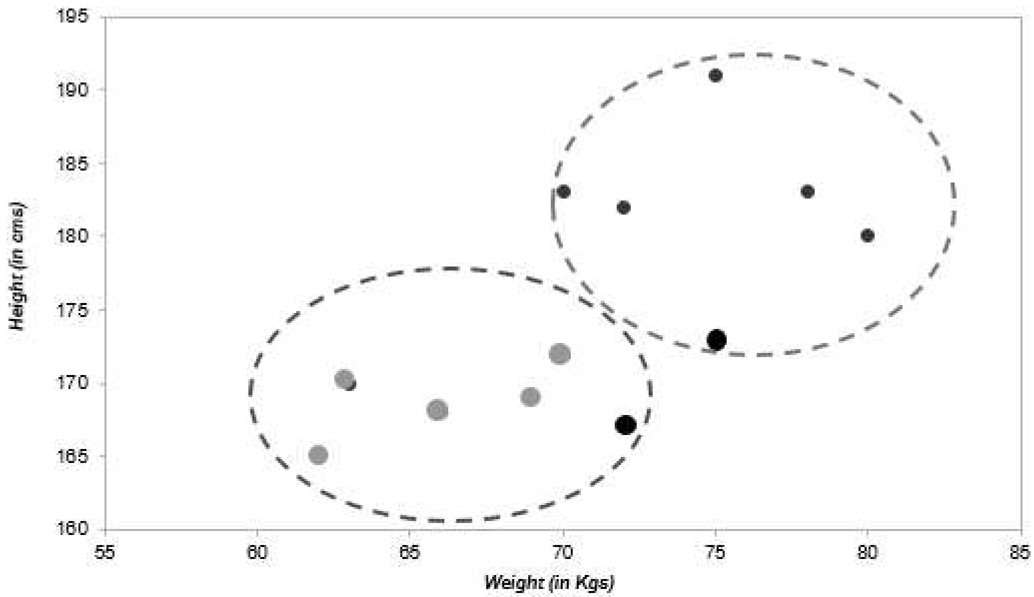
- Learned model depends on following considerations:
  - To measure distance between data points
  - Choice of right set of feature vectors
  - Constraints on complexity of model
    - ✓ Specified number of clusters (in case of Uncluttered data)
    - ✓ Complexity of separating surface
    - ✓ Want to avoid over fitting problem (each data point is in its own cluster, or a complex separating surface)

Clustering approaches

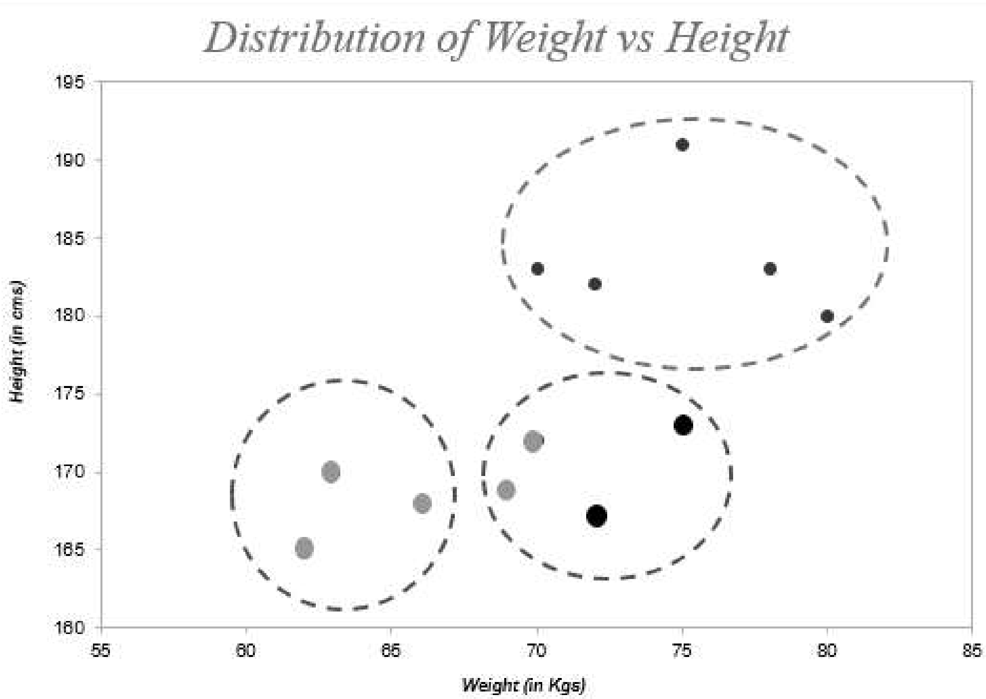
- Suppose there are "k" different groups in our training data, but don't know their labels (here k=2)
  - Pick k samples (at random) from the training set of data.
  - Minimize the distance between remaining samples from the selected data points.
  - Find a new median sample data point in each cluster.
  - Repeat until no change is observed.

- ***Issues:***
  - How do we decide on the best number of clusters?
  - How do we select the best features, the best distance metric?

Clustering using unlabelled Data



Fitting Three Clustering Unsupervised



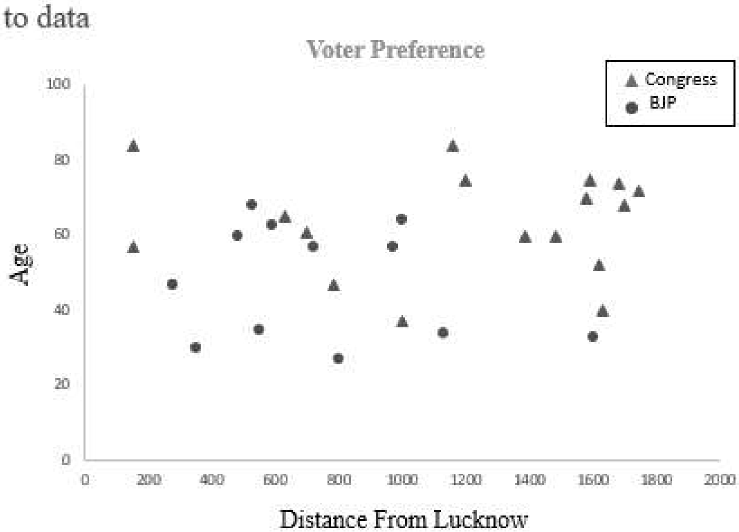
Classification approaches

- Want to find boundaries in feature space that separate different classes of labelled data points
  - Look for simple surface (e.g. best line or plane) that separates classes
  - Look for more complex surfaces (subject to constraints) that separate classes
  - Use voting schemes
    - Find k nearest training examples, use majority vote to select label
- ***Issues:***
  - How do we avoid over-fitting to data?
  - How do we measure performance?
  - How do we select best features?

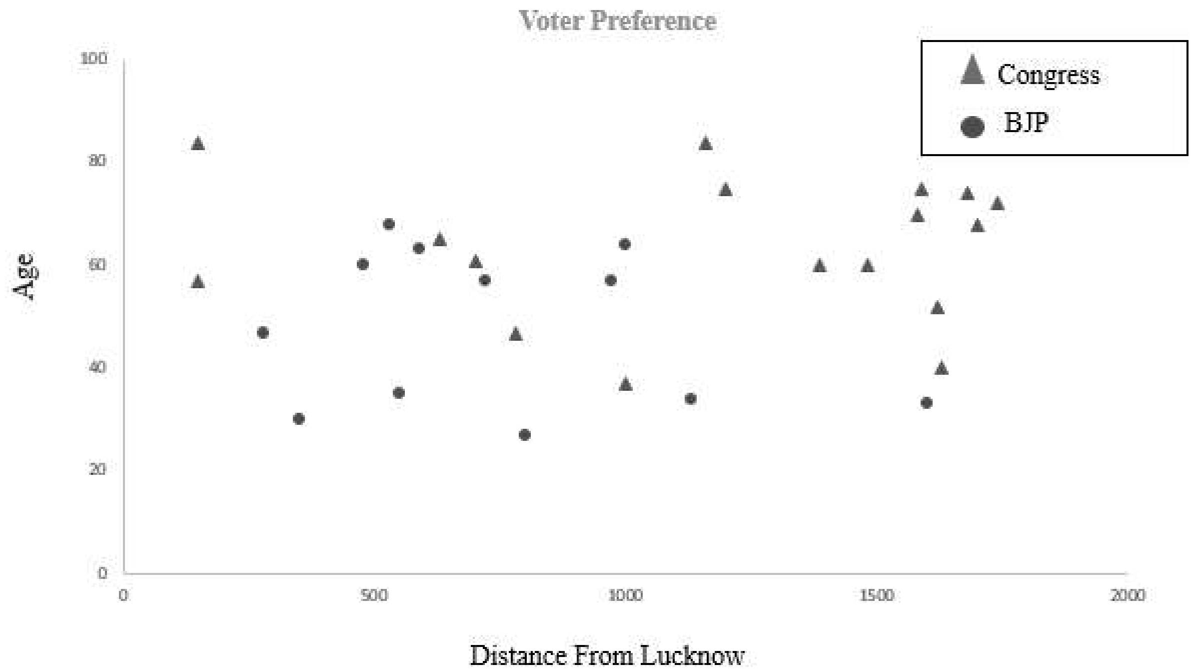
Classification

- Attempt to minimize error on training data
  - Similar to fitting a curve to data
- Evaluate on training data

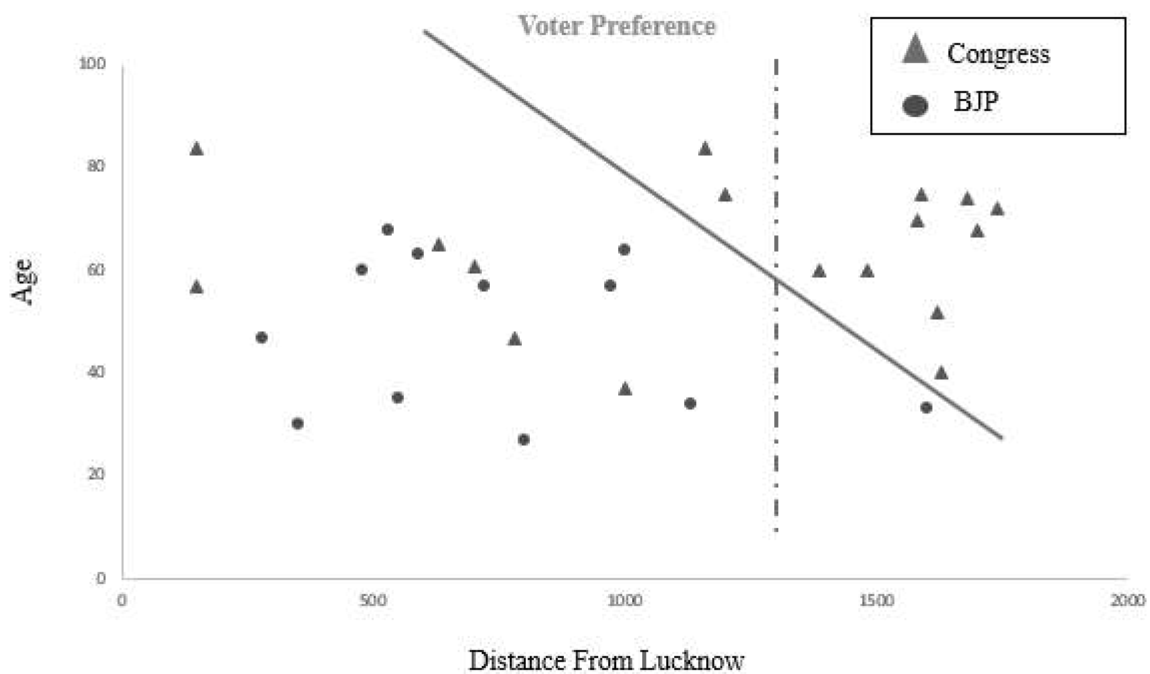
Voter Preference by age and distance from Lucknow



Random Division of Data into Training and Test Set



Possible Models for Training Set



Confusion Matrices (Training Error)

Predicted BJP

Actually BJP

	Pos	Neg
Pos	12	6
Neg	0	11

Solid Line

	Pos	Neg
Pos	11	6
Neg	1	11

Dashed Line

- **True Positive**: Number of BJP voter preference samples that have been classified correctly.
- **False Negative**: Number of Congress voter preference samples that have been classified as BJP voter.
- **True Negative**: Number of Congress voter preference samples that have been classified as correctly.
- **False Positive**: Number of BJP voter preference samples that have been classified as congress voters.

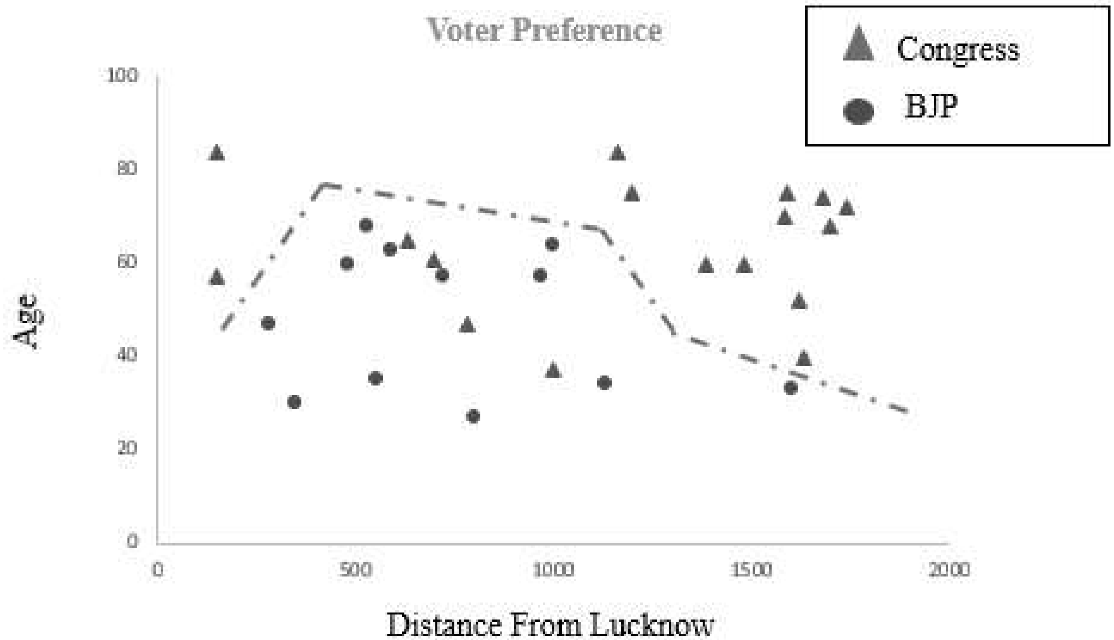
Training Accuracy of Models

$$Accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative}$$

For Solid Line, Accuracy= 0.85

For Dashed Line, Accuracy = 0.88

More Complex Model



TP= 12 FP= 4, TN = 12 FN=0

Accuracy = 24/26 = 0.92

Other Statistical Measures

$$Positive\ Predictive\ Value = \frac{true\ positive}{true\ positive + false\ positive}$$

❖ Solid line model: 0.48

❖ Dashed line model: 0.67

❖ Complex model: 0.61

- One can also use following parameters:

$$Sensitivity = \frac{true\ positive}{true\ positive + false\ negative}$$

Percentage correctly found

$$Specificity = \frac{true\ negative}{true\ negative + false\ positive}$$

Percentage correctly rejected

Summary

- Machine learning methods provide a way of building models from datasets.
- Supervised learning, uses labelled data and creates classifier that optimally separate data into known classes
- Unsupervised clustering tries to infer latent relation between the training set examples by clustering them into nearby groups
- Choice of features influence the results of data points grouping.
- Choice of distance metrics between examples also influence the results.