Module 1 - Introduction

Saturday, 31 August 2019 15:01

Machine Learning Techniques

- Regression / Estimation
  - Predicting Continuous values
  - Like cost of a house with given features
- Classification
  - Predicting the item class/category of a case
  - If a cell is beanie or malignant
- Clustering
  - Finding the structure of data: summarization
  - Customer segmentation in banking field
- Associations
  - Associating frequent co - occurring items
  - Grocery items which frequently occur together.
- Anomaly Detection
  - Discovering abnormal unusual cases
  - Credit card fraud detection
- Sequence mining
  - Predicting next events
  - click-stream (Markov Model, HMM)
- Dimension Reduction
  - Reduce the size of Data(PCA)
- Recommendation Systems
  - Recommending items
  - Books or movies recommendation

Machine Learning components

AI components:
- Computer Vision
- Language Processing
- Creativity

Machine Learning:
- Classification
- Clustering
- Neural Network

Revolution in ML:
- Deep Learning

Types of Machine Learning

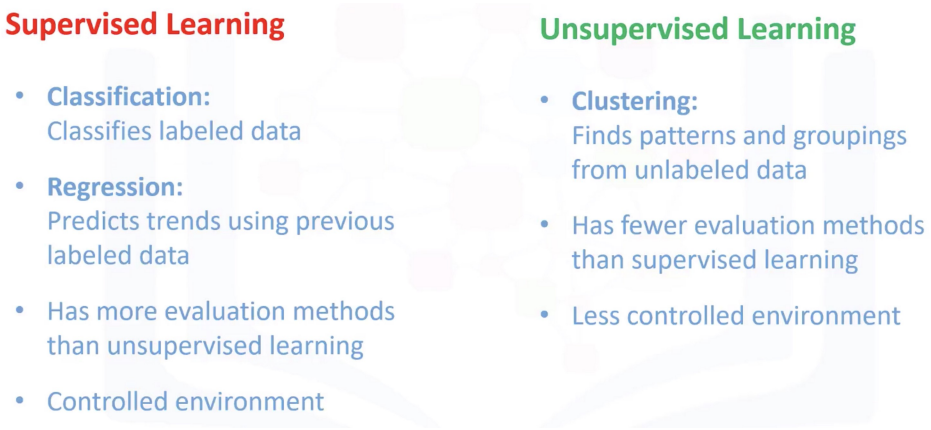
Supervised Learning - (works on data with its labels)

- **Regression** - Dependent Variable(y) is continuous values
  - Simple Linear Regression
  - Multiple Linear Regression
  - Simple/Multiple Non linear Regression

- **Classification** - Dependent Variable(y) is Categorical values
    - K-Nearest Neighbors
    - Decision Trees
    - Logistic Regression
    - Support Vector Machine

Unsupervised Learning - (the model works on its own to discover information labels)

- Clustering
    - K - Means
    - Agglomerative Clustering
    - DBSCAN
- Dimension Reduction
- Density Estimation
- Market Basket Analysis



Module 2 - Regression

Saturday, 31 August 2019 16:42

Data set for Regression where the label or the dependent value is called y (CO2EMISSION) and other parameters are called independent values x

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Regression

- **Simple Regression** - predict CO2 vs EngineSize
  - Simple Linear
  - Simple Non Linear
- **Multiple Regression** - predict CO2 vs EngineSize and Cylinder
  - Multiple Linear
  - Multiple Non Linear

Applications of Regression

- Sales Forecasting
- Satisfaction Analysis
- Price estimation
- Employment Income

Regression Algorithms

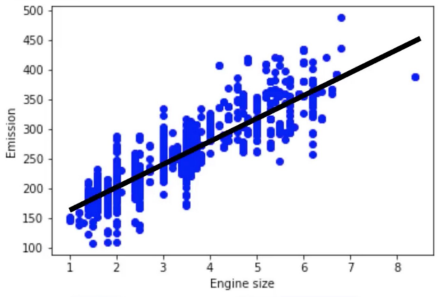
- Ordinal Regression
- Poisson Regression
- Fast forest quantile regression
- Linear, Polynomial, Lasso, Stepwise, Ridge Regression
- Bayesian linear regression
- Neural network regression
- Decision forest regression
- Boosted decision tree regression
- KNN (K-nearest neighbors)

Simple Linear Regression

06 September 2019 19:56

Simple Linear Regression

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

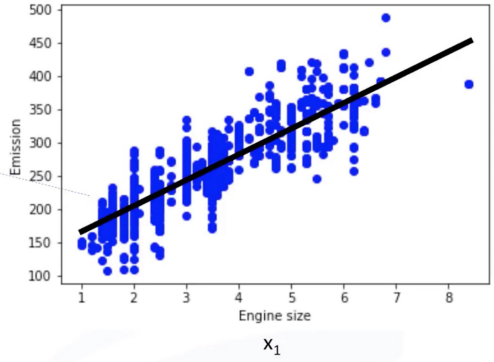


$$\hat{y} = \theta_0 + \theta_1 x_1$$

response variable

a single predictor

y



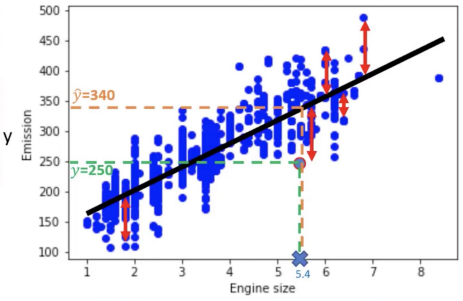
x_1

$x_1 = 2.4$ independent variable

$y = 250$ actual Co2 emission of x1

$\hat{y} = \theta_0 + \theta_1 x_1$

$\hat{y} = 340$ the predicted emission of x1



Error = $y - \hat{y}$

= 250 – 340

= -90

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267

X_1 y

$$\hat{y} = \theta_0 + \theta_1 x_1$$

$$\theta_1 = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s}(x_i - \bar{x})^2}$$

$\bar{x}$ = (2.0 + 2.4 + 1.5 + ...)/9 = 3.34

$\bar{y}$ = (196 + 221 + 136 + ...)/9 = 256

$$\theta_1 = \frac{(2.0 - 3.34)(196 - 256) + (2.4 - 3.34)(221 - 256) + \dots}{(2.0 - 3.34)^2 + (2.4 - 3.34)^2 + \dots}$$

$\theta_1 = 39$

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

$\theta_0 = 256 - 39 * 3.34$

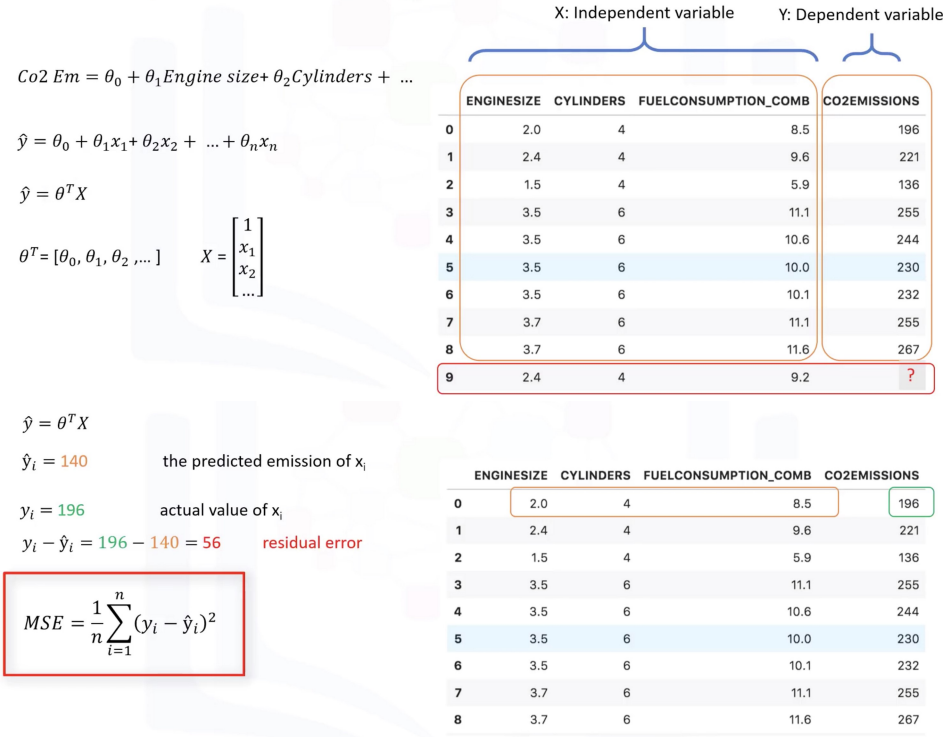
$\theta_0 = 125.74$

$$\hat{y} = 125.74 + 39x_1$$

Multiple Linear Regression

06 September 2019 19:57

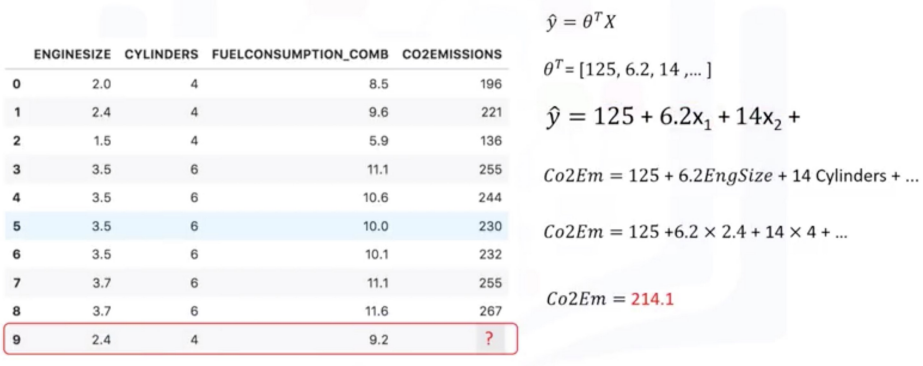
Multiple Linear Regression



How to minimize MSE (mean squared error) ?

Find best parameter Theta

1. Ordinary Least Square
    - Linear Algebra operations on matrix
    - Takes longer time for 10k+ rows
2. An Optimization Algorithm
    - Gradient Descent (which iteratively finds best Theta by taking random value initially)



Q&A

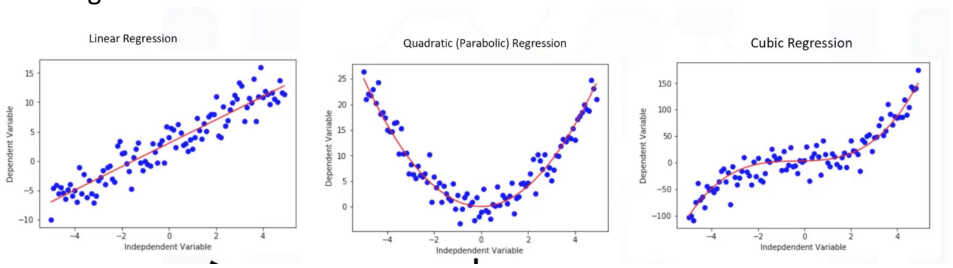
- How many Independent Variables should you use?
  Using too many x's could cause overfit, so we have to take care about it.
- Should the independent variable be continuous?
  A discrete classified variable (categorical variable) could also be written in terms of numeric digits
  Example: car Type
    Automatic : 0
    Manual : 1

Simple Non Linear Regression

06 September 2019 19:59

Simple Non Linear Regression

- Polynomial Regression



  - Quadratic
  - Cubic .. Etc

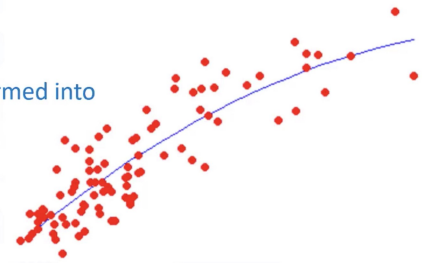
    This could be converted to Linear Regression as:
    - Some curvy data can be modeled by a polynomial regression
    - For example:

      $\hat{y} = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$

    - A polynomial regression model can be transformed into linear regression model.

      $x_1 = x$
      $x_2 = x^2$
      $x_3 = x^3$

      $\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$ → Multiple linear regression → Least Squares



- Non Linear Regression
  - Exponential
  - Logarithmic
  - Logistics/Sigmoid

$$\hat{y} = \theta_0 + {\theta_2}^2 \mathrm{x}$$

$$\hat{y} = \theta_0 + \theta_1 {\theta_2}^x$$

$$\hat{y} = \log(\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3)$$

$$\hat{y} = \frac{\theta_0}{1 + {\theta_1}^{(x-\theta_2)}}$$

- Transform data

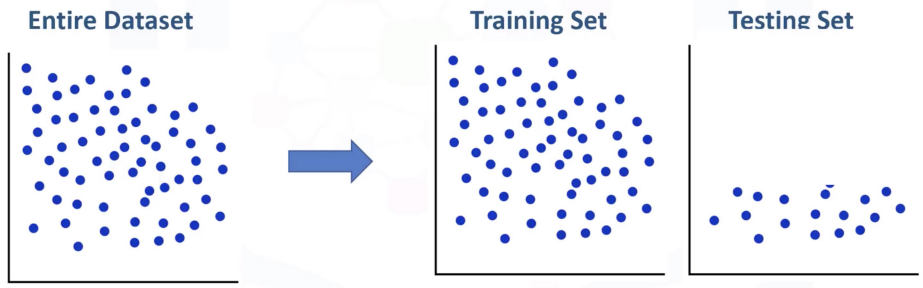
  ~~Out of scope of the course~~

Model Evaluation Approaches

06 September 2019 19:58

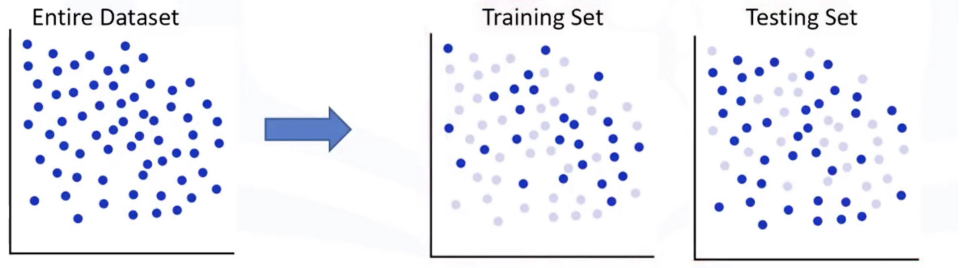
Model Evaluation Approaches

- Train and Test on same Data set



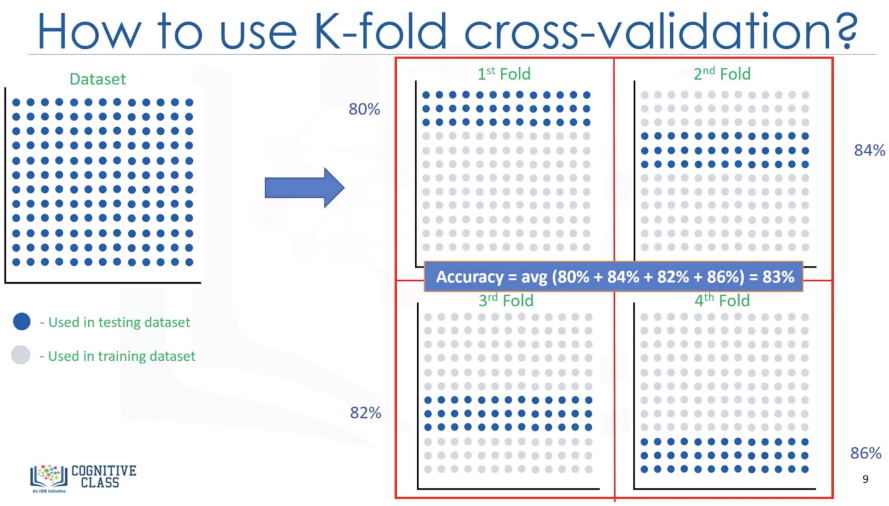
Training Accuracy : high
Out of sample Accuracy : Low

- Train/Test Split



Out of sample Accuracy : Better than above

- Kfold cross validation



Training Accuracy?

Percentage of correct predictions made when using the test data set which was already used for training.

Out of sample accuracy?

Percentage of correct predictions made when using the

test data set which has not been trained on.

Note:

Hence we should have high Out of Sample Accuracy than the training Accuracy so that we avoid overfitting of data.

Evaluation Metrics

- Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

- Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Root Mean Squared Error

$$\mathrm{R}MSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

- Relative Absolute Error

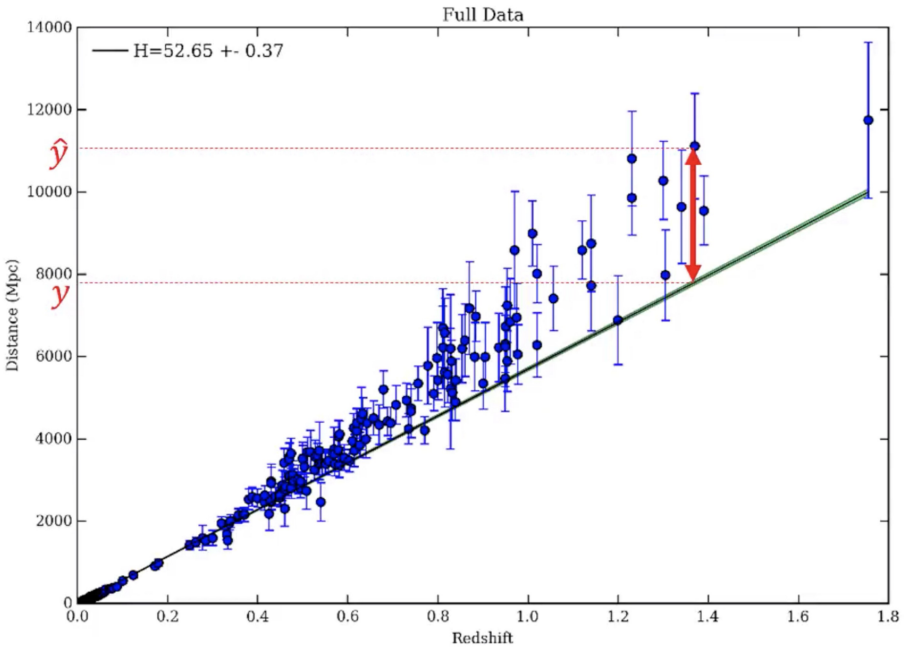
$$RAE = \frac{\sum_{j=1}^{n}|y_j - \hat{y}_j|}{\sum_{j=1}^{n}|y_j - \bar{y}|}$$

- Relative Squared Error

$$RSE = \frac{\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}{\sum_{j=1}^{n}(y_j - \bar{y})^2}$$

- R squared (coefficient of determination)

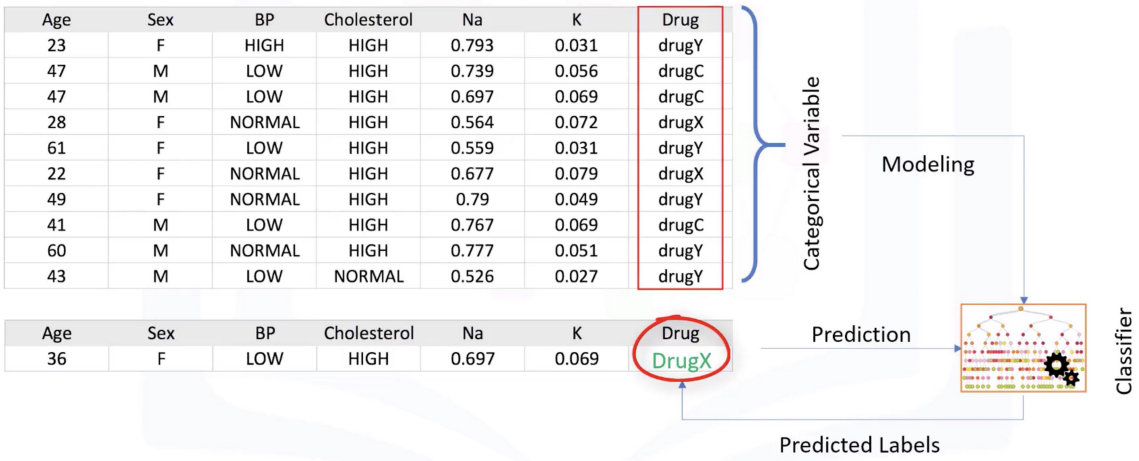
$$R^2 = 1 - RSE$$



Module 3 - Classification

Tuesday, 3 September 2019 21:08

The dependent value which is the label have unique values having 2 or more values which could be classified based on the independent values x.



Classification Algorithms

- Decision Trees (ID3, C4.5, C5.0)
- Naïve Bayes
- Linear Discriminant Analysis
- K-Nearest Neighbor
- Logistics Regression
- Neural Networks
- Support Vector Machines

Applications of Classification

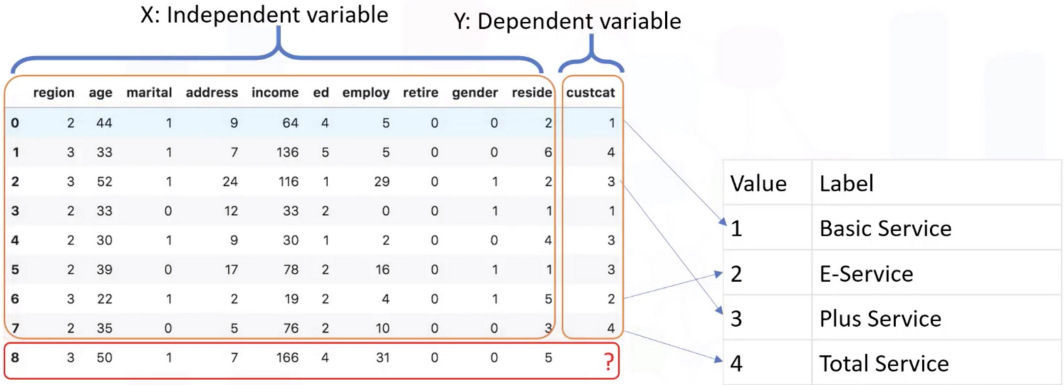
- Drug Detection
- Disease Identification
- Email Spam or not Spam

K - Nearest Neighbor

Tuesday, 3 September 2019 19:25

K-Nearest Neighbors

Given below is a dataset for classification:



	region	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	44	1	9	64	4	5	0	0	2	1
1	3	33	1	7	136	5	5	0	0	6	4
2	3	52	1	24	116	1	29	0	1	2	3
3	2	33	0	12	33	2	0	0	1	1	1
4	2	30	1	9	30	1	2	0	0	4	3
5	2	39	0	17	78	2	16	0	1	1	3
6	3	22	1	2	19	2	4	0	1	5	2
7	2	35	0	5	76	2	10	0	0	3	4
8	3	50	1	7	166	4	31	0	0	5	?

Value	Label
1	Basic Service
2	E-Service
3	Plus Service
4	Total Service

- Using 1st KNN for 2 independent Variables (age, income)

- Using 5 KNN for the same data

K-Nearest Neighbors Algorithm

1. Pick a value for K
2. Find all the distances to other points from the unknow point(predicting point)
3. Select the K shortest distances from the above distances.
4. Find the majority case (label) in the K shortest distances

How to?:

How to Step 2: How to find the distance between two points (Euclidean distance)?

- 1 Feature (independent variable x)

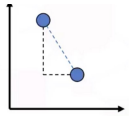
Customer 1
Age
54

Customer 2
Age
50

$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

- 2 Features

Customer 1	
Age	Income
54	190



Customer 2	
Age	Income
50	200

$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

$$= \sqrt{(54-50)^2+(190-200)^2} = 10.77$$

- 3 Features

Customer 1		
Age	Income	Education
54	190	3

Customer 2		
Age	Income	Education
50	200	8

$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

$$= \sqrt{(54-50)^2+(190-200)^2+(3-8)^2} = 11.87$$

NOTE:

- ★ We need to normalize the data (features) to find the Euclidean distances

How to Step 1: How to pick value for k?

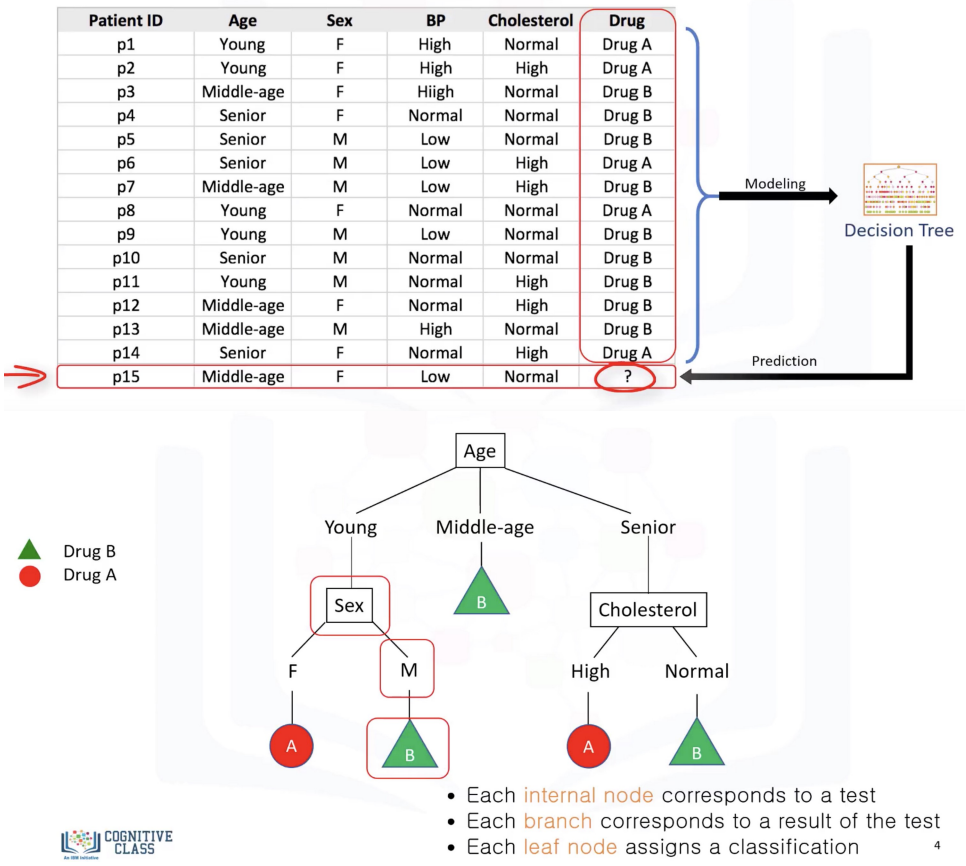
- K = 1, overfitting happens
- K very large, Overly generalization
- So what's the right K for my model?

  Find the accuracy of prediction of various K values for a given Test Data, and pick the best K which has the highest prediction accuracy

Decision Tree

Tuesday, 3 September 2019 19:25

Decision Tree



How to Build a decision Tree?

a. For each attribute(independent variable y) in the data set
   Calculate its significance (information gain)
   in splitting the data
b. attribute having highest information gain is used to split the data
   The child with least impurity (Entropy/gini impurity) becomes a leaf
c. Step a: Is repeated for all non leaf children

NOTE:

★ We need to set all the x values as numbers as sklearn doesn't accept categorical values

How to?

Step a: which attribute is best?

Attribute split having low Entropy (randomness in the set, or impurity)
And high information Gain

Weighted Entropy ↓

Information Gain ↑

$$\text{Entropy} = - p(A)\log(p(A)) - p(B)\log(p(B))$$

$$\text{Information Gain} = (\text{Entropy before split}) - (\text{weighted entropy after split})$$

Entropy before splitting

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

S: [9 B, 5 A]

E = - p(B)log(p(B)) - p(A)log(p(A))

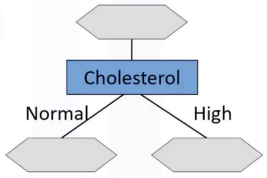
E = - (9/14)log(9/14) - (5/14)log(5/14)

E = 0.940

Entropy after Splitting with cholesterol

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

S: [9 B, 5 A]
E = 0.940



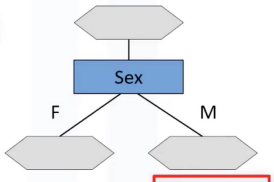
S: [6 B, 2 A]
E = 0.811

S: [3 B, 3 A]
E = 1.00

Entropy after splitting with Sex

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

S: [9 B, 5 A]
E = 0.940

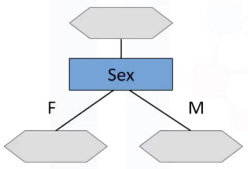


S: [3 B, 4 A]
E = 0.985

S: [6 B, 1 A]
E = 0.592

Step b: Tree with the Highest Information Gain?

S: [9 B, 5 A]
E = 0.940

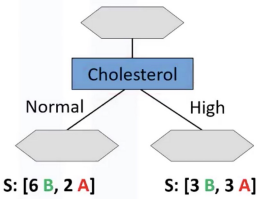


S: [3 B, 4 A]
E = 0.985

S: [6 B, 1 A]
E = 0.592

?

S: [9 B, 5 A]
E = 0.940



S: [6 B, 2 A]
E = 0.811

S: [3 B, 3 A]
E = 1.00

Gain (s, Sex)
= 0.940 – [(7/14)0.985 + (7/14)0.592]
=0.151

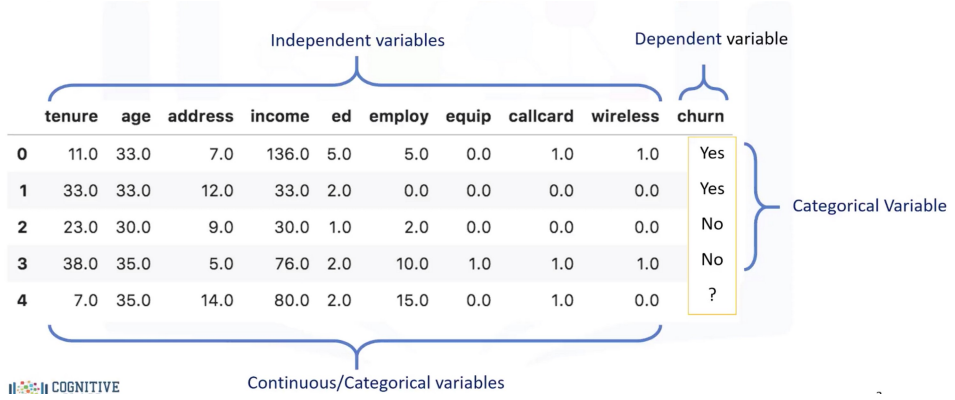
Gain (s, Cholesterol)
= 0.940 – [(8/14).811 + (6/14)1.0]
=0.048

Sex attribute split

Logistics Regression

Tuesday, 3 September 2019 19:26

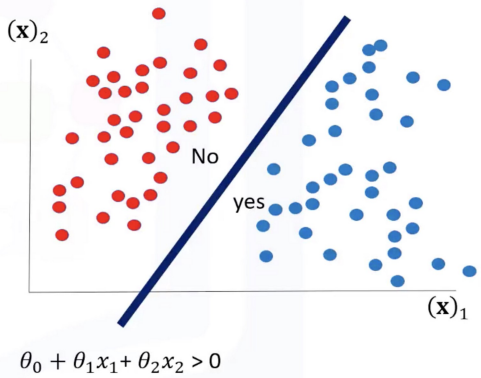
Logistics Regression



	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	Yes
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	Yes
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	No
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	No
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	?

When to Use Logistics Regression?

- If you have labels (dependent variable y) is binary
  - 0/1
  - Yes/No
  - True/False
- If you need **probabilistic** results
- When you need a linear decision boundary



$\theta_0 + \theta_1 x_1 + \theta_2 x_2 > 0$

- If you need to understand the impact of a feature.

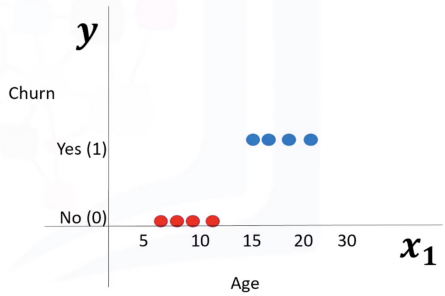
Applications of Logistic Regression

- Predicting the probability of a person having a heart attack
- Predicting the mortality in injured patients
- Predicting a customer's propensity to purchase a product or halt a subscription
- Predicating the probability of failure of a given process or product
- Predicting the likelihood of home owner defaulting on mortgage

Can we use Linear Regression for Categorical Labels?? [NO]

When we try to plot labels(y) for regression purpose..

	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	1
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	1
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	0
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	0
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	0



Now trying to fit a linear line using Theta1 and Theta0 as parameters…
Note: thetas are called the Weights or the confidence of the line

$\theta^T X = \theta_0 + \theta_1 x_1$

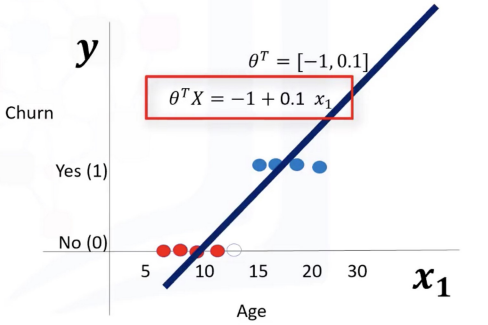
$\theta^T X = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ \ldots$

$\theta^T = [\theta_0, \theta_1, \theta_2, \ldots]$ $\quad X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \ldots \end{bmatrix}$

$\theta^T = [\theta_0, \theta_1]$

$\theta_0 + \theta_1 x_1$

a + b x_1



Now, predicting class (y_hat) for x(age) = 13..

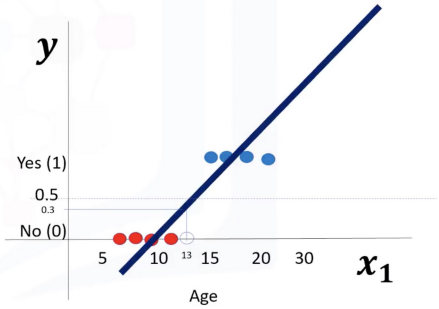
$\theta^T X = \theta_0 + \theta_1 x_1$

$p_1 = [13]$ → $\theta^T X = -1 + 0.1 \,.\, x_1$
= -1 + 0.1 x 13
= 0.3

$$\hat{y} = \begin{cases} 0 & \text{if } \theta^T X < 0.5 \\ 1 & \text{if } \theta^T X \geq 0.5 \end{cases}$$

$\theta^T X = 0.3$
$\theta^T X < 0.5$ → Class 0

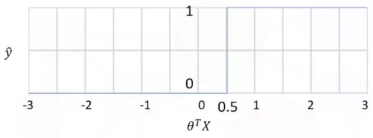
$\theta^T X = -1 + 0.1 \,.\, x_1$



Problem to use linear regression for Categorical Labels:

We cant say about the probability of y_hat belonging to a particular class, hence we cant be sure, therefore we instead use a sigmoid function (which is nothing but Logistic regression)

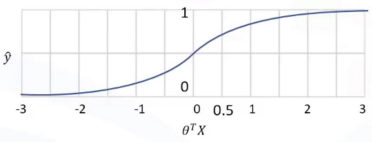
$\theta^T X = \theta_0 + \theta_1 x_1 + \cdots$



$$\hat{y} = \begin{cases} 0 & \text{if } \theta^T X < 0.5 \\ 1 & \text{if } \theta^T X \geq 0.5 \end{cases}$$

So, lets use Sigmoid function instead… hence this gives probability

$\sigma(\theta^T X) = \sigma(\theta_0 + \theta_1 x_1 + \cdots)$



$\hat{y} = \sigma(\theta^T X)$

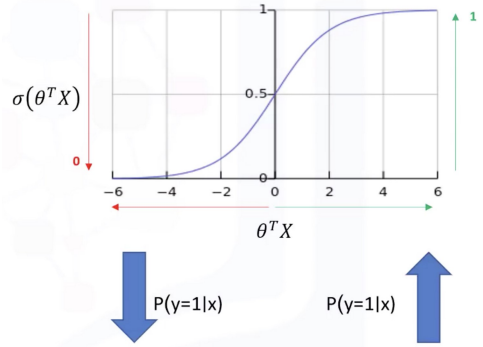
P(y=1|x)

What is Sigmoid or Logistics Function?

$$\sigma(\theta^T X) = \frac{1}{1 + e^{-\theta^T X}}$$

$\sigma(\theta^T X) = 1$

$\sigma(\theta^T X) = 0$



We see, as $\Theta^T X$ Increases then value of $\sigma(\Theta^T X)$ becomes close to 1
And when $\Theta^T X$ decreases then value of $\sigma(\Theta^T X)$ becomes close to 0

Hence Sigmoid Function can give probability of y(Binary categorical variable) being 1 wrt to $\Theta^T X$

So the output of Logistics Regression is

$P(Y = 1 \mid x)$
$P(Y = 0 \mid x) = 1 - P(Y = 1 \mid x)$

Example: P(churn = 1 | income, age) = 0.8
Hence the model:

$\boldsymbol{\sigma(\theta^T X)}$ → **P(y=1|x)**

$\boldsymbol{1 - \sigma(\theta^T X)}$ → **P(y=0|x)**

Training the Logistics Model

$\boldsymbol{\sigma(\theta^T X)}$ → **P(y=1|x)**

1. Initialize θ.
2. Calculate $\hat{y} = \sigma(\theta^T X)$ for a customer.
3. Compare the output of $\hat{y}$ with actual output of customer, y, and record it as error.
4. Calculate the error for all customers.
5. Change the θ to reduce the cost.
6. Go back to step 2.

$\theta = [-1,2]$

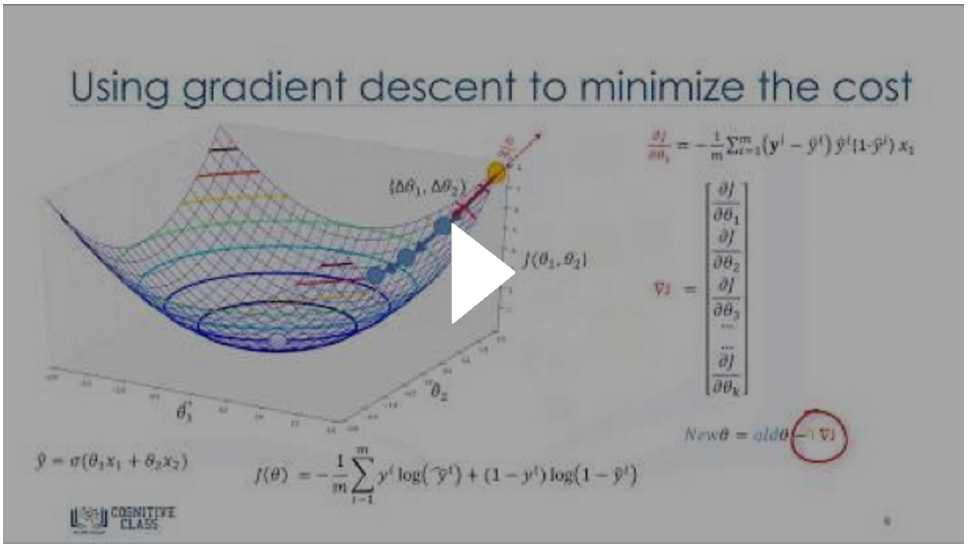
$\hat{y} = \sigma([-1,2] \times [2,5]) = 0.7$

Error = 1-0.7 = 0.3

$Cost = J(\theta)$

θ_{new}

Click here to Learn more ML0101EN v3 - Logistic Regression - Training 13:50


Using gradient descent to minimize the cost
$\frac{\partial J}{\partial \theta_1} = -\frac{1}{m}\sum_{i=1}^{m}(y^i - \hat{y}^i)\,\hat{y}^i(1-\hat{y}^i)\,x_1$
$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \\ \frac{\partial J}{\partial \theta_3} \\ \cdots \\ \frac{\partial J}{\partial \theta_k} \end{bmatrix}$
$(\Delta\theta_1, \Delta\theta_2)$
$J(\theta_1, \theta_2)$
θ_1
θ_2
$New\theta = old\theta - \nabla J$
$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$
$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^i \log(\hat{y}^i) + (1-y^i)\log(1-\hat{y}^i)$
COGNITIVE CLASS

Support Vector Machine

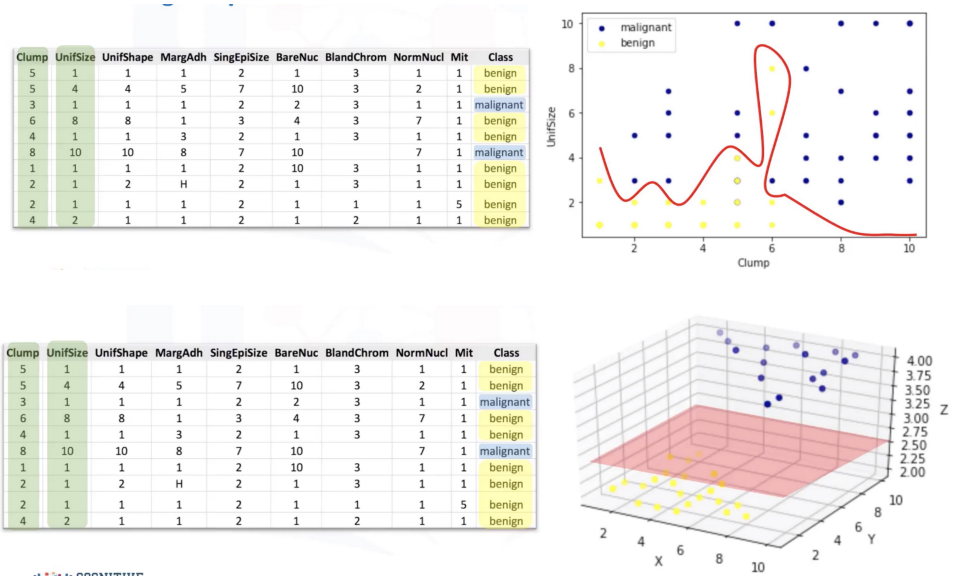
Tuesday, 3 September 2019 19:28

Support Vector Machine

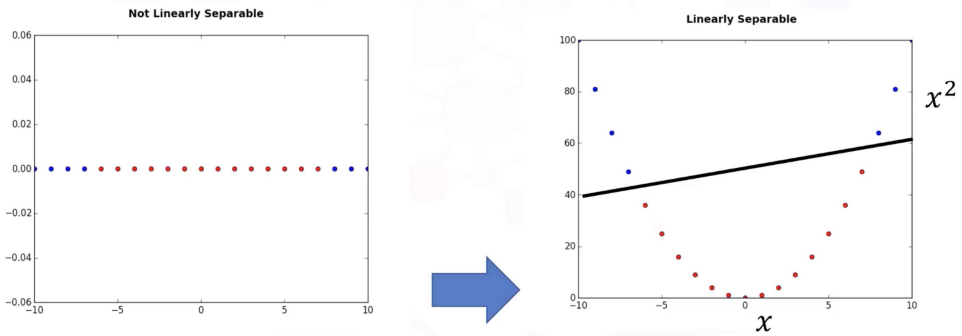
SVM is as supervised algorithm that classifies cases by finding a separator

- Mapping data to a high dimensional feature space (kernelling)
- Finding a separator

1. Mapping data to higher dimension to divide the data using a hyper plane instead of a non linear



Non linearly Separable to Linearly Separable (mapping data to higher space : kernelling)



$$\phi(x) = [x, x^2]$$

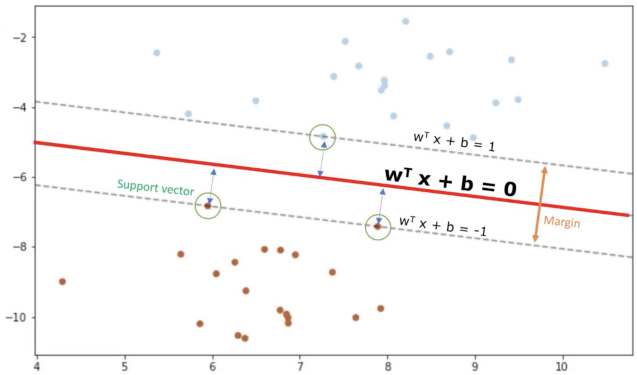
Kernelling types

a. Linear

b. Polynomial
c. Radial Bases Function RBF
d. Sigmoid

2. **Choosing a separator**

Find $\mathbf{w}$ and b such that
$\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$



Pros and Cons of SVM

- Advantage
  - Accurate in high Dimension space
  - Memory Efficient (support vectors)

- Disadvantage
  - Prone to overfitting when number of features are more than samples
  - No probabilistic results
  - Not efficient computationally when data is big (1000+ rows)

Applications of SVM

- Image recognition (pics and handwritten letters)
- Text category assignment
- Detecting spam
- Sentiment analysis
- Gene Expression Classification
- Regression, outlier detection and clustering

Evaluation Metrics

Tuesday, 3 September 2019 19:26

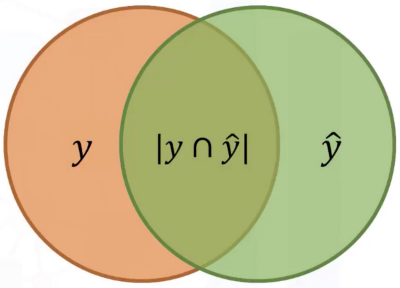
Evaluation Metrics in Classification

- **Jaccard Index**

y: Actual labels

$\hat{y}$: Predicted labels

$$J(y,\hat{y}) = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|} = \frac{|y \cap \hat{y}|}{|y| + |\hat{y}| - |y \cap \hat{y}|}$$



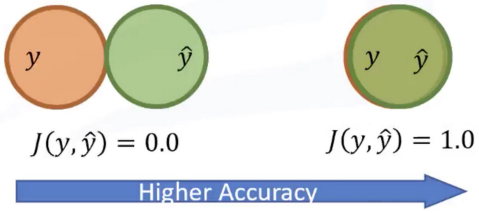
Example:

y: [0, 0, 0, 0, 0, 1, 1, 1, ,1 ,1]

$\hat{y}$: [1, 1, 0, 0, 0, 1, 1, 1, ,1 ,1]

$$J(y,\hat{y}) = \frac{8}{10+10-8} = 0.66$$

Accuracy:



- **F1 Score**

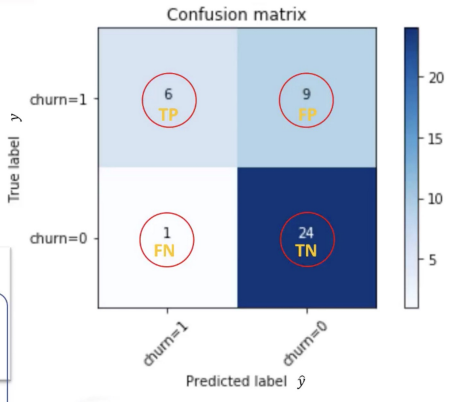
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1-score = 2x (prc x rec) / (prc+rec)

F1-score: 0.00 ... 0.20 .... 0.55 .... 0.83 ... 1.00

Higher Accuracy

	precision	recall	f1-score
Churn = 0	0.73	0.96	0.83
Churn = 1	0.86	0.40	0.55

Avg Accuracy = 0.72



TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives

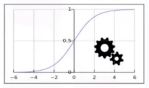
- **Log Loss**

  Used when predicted labels are in terms of accuracy (percentage) to be that label.

Performance of a classifier where the predicted output is a probability value between 0 and 1.

Test set

	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	1
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	1
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	0
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	0
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	0



Test

Predicted churn	LogLoss
0.91	0.11
0.13	2.04
0.04	0.04
0.23	0.26
0.43	0.56

$LogLoss = 0.60$

Actual Labels y

$\hat{y}$ Predicted Probability

$$LogLoss = -\frac{1}{n}\sum(y \times \log(\hat{y}) + (1-y) \times \log(1-\hat{y}))$$

Accuracy:

LogLoss: 0.00 ... 0.35 .... 0.60 ... 1.00

Higher Accuracy

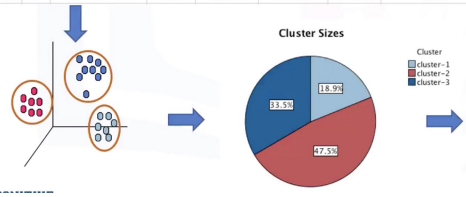
Module 4 - Clustering

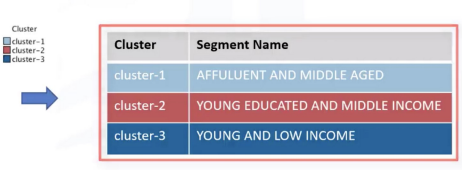
Tuesday, 3 September 2019 18:16

Introduction

Clustering is used for segmentation of a dataset without any labels

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1





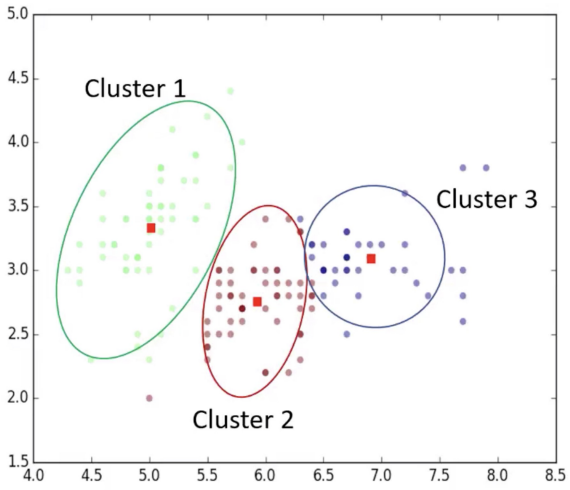
Cluster	Segment Name
cluster-1	AFFULUENT AND MIDDLE AGED
cluster-2	YOUNG EDUCATED AND MIDDLE INCOME
cluster-3	YOUNG AND LOW INCOME

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1

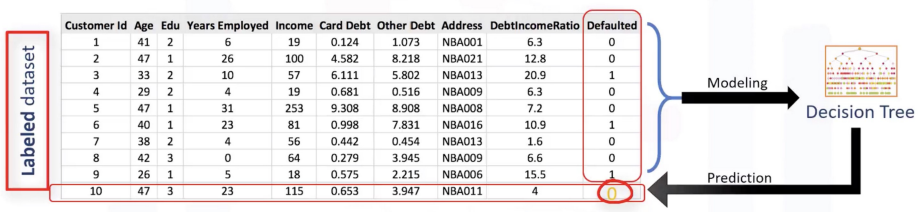
Customer ID	Segment
1	YOUNG AND LOW INCOME
2	AFFULUENT AND MIDDLE AGED
3	AFFULUENT AND MIDDLE AGED
4	YOUNG AND LOW INCOME
5	AFFULUENT AND MIDDLE AGED
6	AFFULUENT AND MIDDLE AGED
7	YOUNG AND LOW INCOME
8	YOUNG AND LOW INCOME
9	AFFULUENT AND MIDDLE AGED

So what is clustering?

A group of objects which are similar to other objects in the cluster, and dissimilar to data points in other clusters



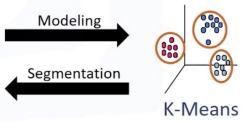
Classification vs Clustering



Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1
10	47	3	23	115	0.653	3.947	NBA011	4	0

Unlabeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1



Applications of Clustering

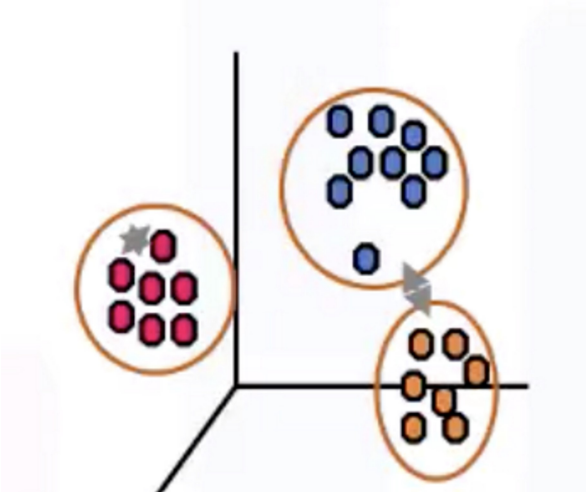
- Retail / marketing
  - Identifying buying patterns of customers
  - Recommendation of movies to new customers
- Banking
  - Patterns of Fraud credit card detection
  - Clusters of customer wither loyal or churn
- Insurance
  - Insurance Risk of customers based on segments
- Medicine
  - Characterize patents with similar behavior to identify successful medical therapy
- Biology
  - Find Gene and family ties

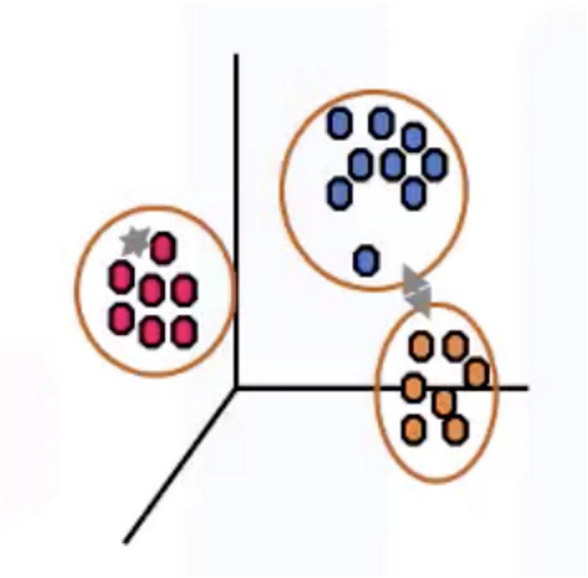
Why Clustering?

- Exploratory data analysis
- Summary generation
- Outlier Detection
- Finding duplicates
- Pre processing step

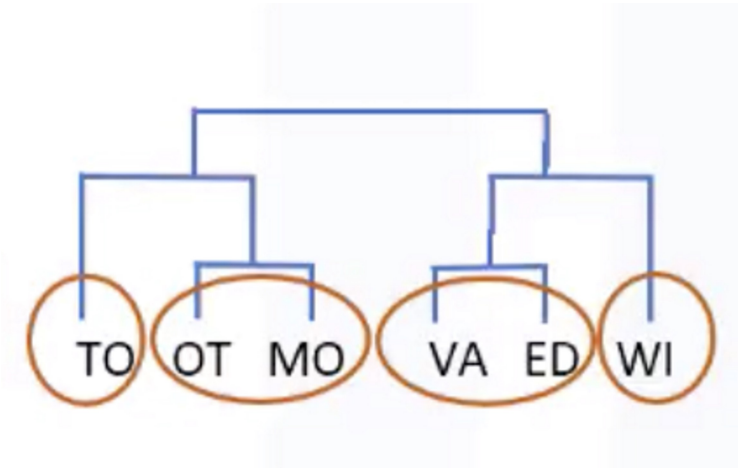
Clustering Algorithms

- **Partitioned Based Clustering**
  - Relatively Efficient
  - Medium or large size of datasets
  - Example:
    - K - Means
    - K - Median
    - Fuzzy C - Means

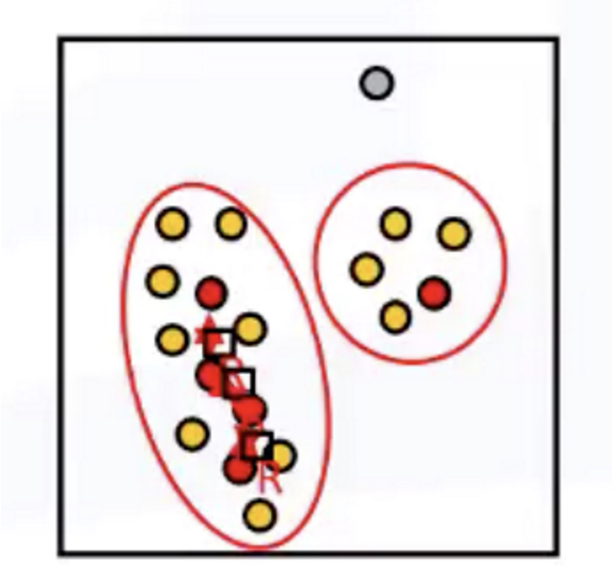




- **Hierarchical Clustering**
  - Produces Trees of Clusters
  - Small Size Datasets
  - Example:
    - Agglomerative
    - Divisive



- **Density Based Clustering**
  - Produces arbitrary shaped Clustering
  - When there is noise in Dataset
  - Example:
    - DBSCAN



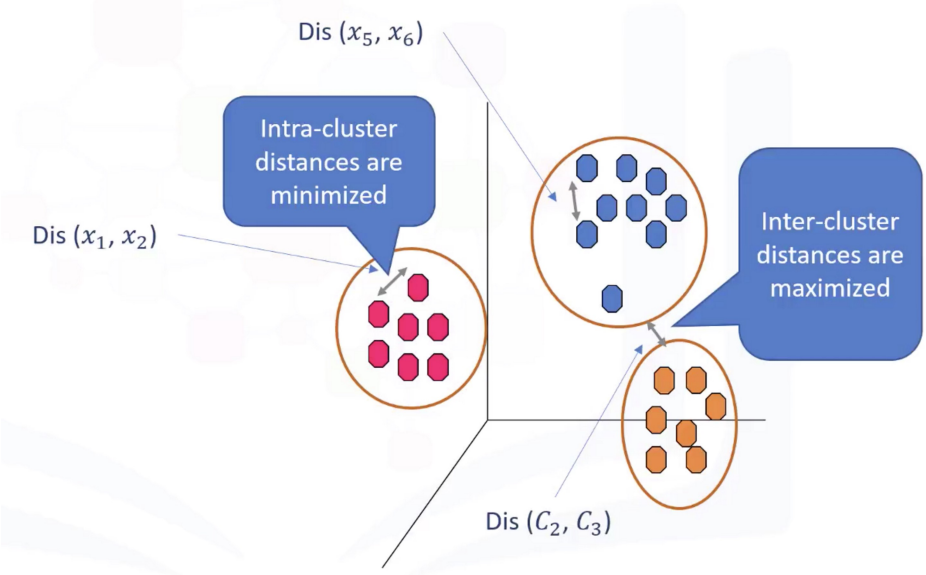
K - Means

Tuesday, 3 September 2019 19:24

K - Means

- Unsupervised clustering based on similarity of features based on **Partitioning**
- K means has 'k' number of clusters which are **non overlapping**
- Produces **Sphere** like clusters
- Examples in the cluster are very similar
- Examples across different clusters are different
- Can find only local optimums (why?)

So to group similar data points, rather than grouping datapoints based on **'similarity'** metrics we use **'dissimilarity'** metrics as dissimilarity is the **'distance'** between 2 datapoints.



The distance or the dissimilarity metrics is measured by Euclidian distances between the data points.

Customer 1		
Age	Income	education
54	190	3

Customer 2		
Age	Income	education
50	200	8

$$\text{Dis}\ (x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

NOTE:

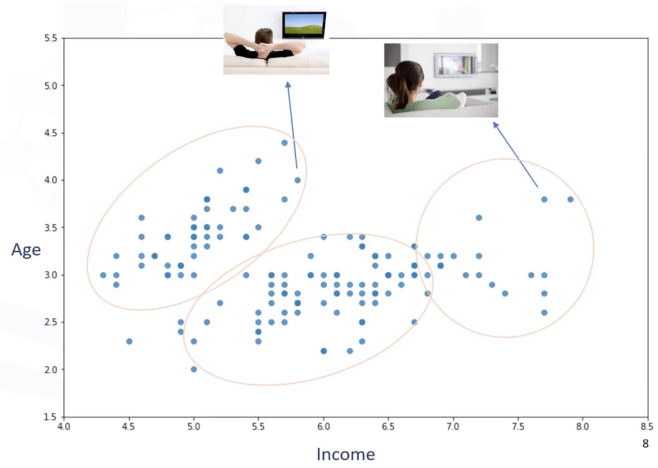
★ Just like KNN Classification Algorithm, we have to Normalize the dataset in order to find the Euclidian distances for any number of features.

Algorithm for K - Means

Dataset:

For simplicity we take 2 dimensional Dataset (2 features)

Customer ID	Age	Income
1	3	4
2	2	6
3	3.5	2
...	...	..



COGNITIVE CLASS

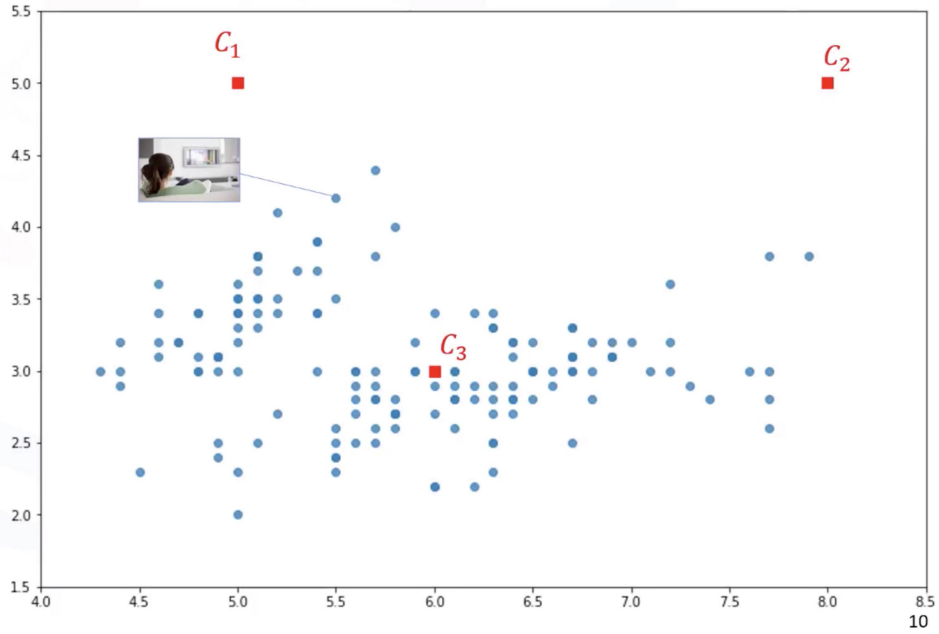
- **Step 1: Initialize K value & pick k centroids for k clusters**

  Lets take k = 3

  How to pick Centroids?

  i. Pick random 3 observations from the dataset

  ii. Select 3 random points

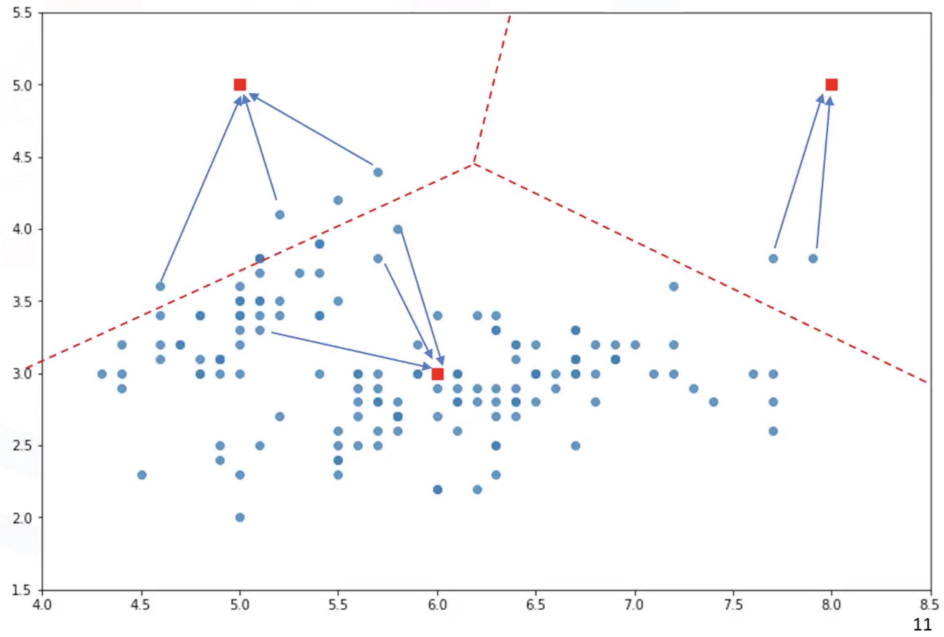


- **Step 2: Assign distance for each Datapoint for each Centroid**

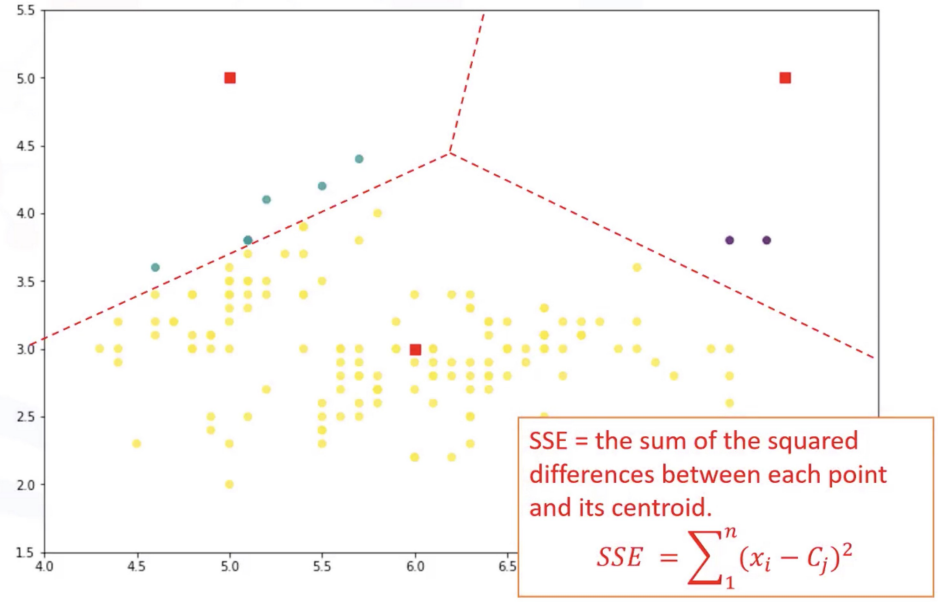
  Distance Matrix (Euclidian Distance)

$$\begin{matrix} C_1 & C_2 & C_3 \end{matrix}$$
$$\begin{bmatrix} d(p1,c1) & d(p1,c2) & d(p1,c3) \\ d(p2,c1) & d(p2,c2) & d(p2,c3) \\ d(p3,c1) & d(p3,c2) & d(p3,c3) \\ d(p4,c1) & d(p4,c2) & d(p4,c3) \\ d(p\ldots,c1) & d(p..,c2) & d(p..,c3) \\ d(pn,c1) & d(pn,c2) & d(pn,c3) \\ d(p..,c1) & d(p..,c2) & d(p..,c3) \\ d(p\ldots,c1) & d(p..,c2) & d(p..,c3) \\ d(pn,c1) & d(pn,c2) & d(pn,c3) \\ d(p..,c1) & d(p..,c2) & d(p..,c3) \\ d(p\ldots,c1) & d(p..,c2) & d(p..,c3) \\ d(pn,c1) & d(pn,c2) & d(pn,c3) \\ d(p..,c1) & d(p..,c2) & d(p..,c3) \\ d(p\ldots,c1) & d(p..,c2) & d(p..,c3) \\ d(pn,c1) & d(pn,c2) & d(pn,c3) \end{bmatrix}$$

- **Step 3: Assign each dataset to the Closest Centroid**



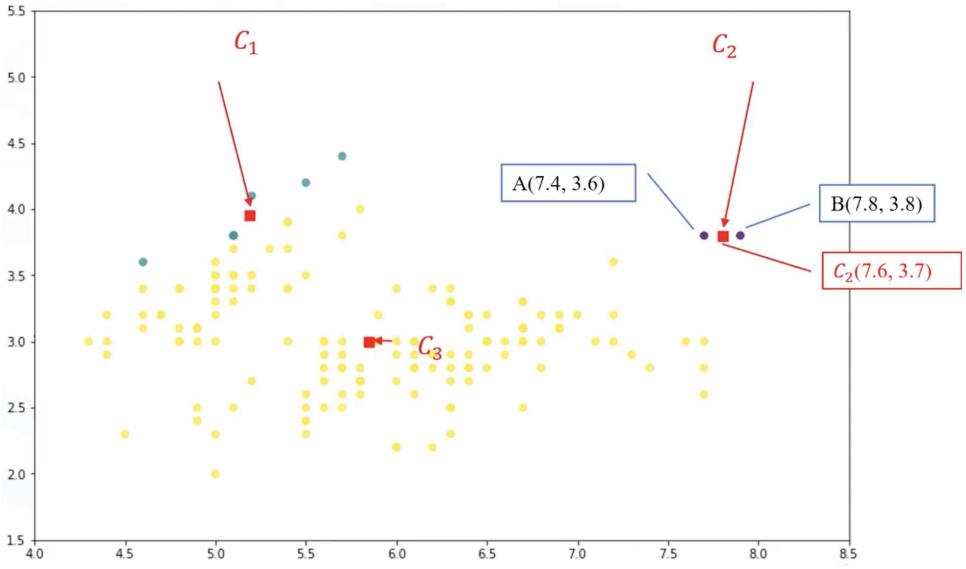
We see that the clusters are not good, this could be measured by the sum of squared distances between the datasets and the centroids of the clusters.



We need to minimize this SSE in order to get better clusters. So essentially we have to move the clusters.

- **Step 4: Compute the new Centroids for the Clusters**

  New centroid location is the **mean** of all the datasets in that particular cluster.

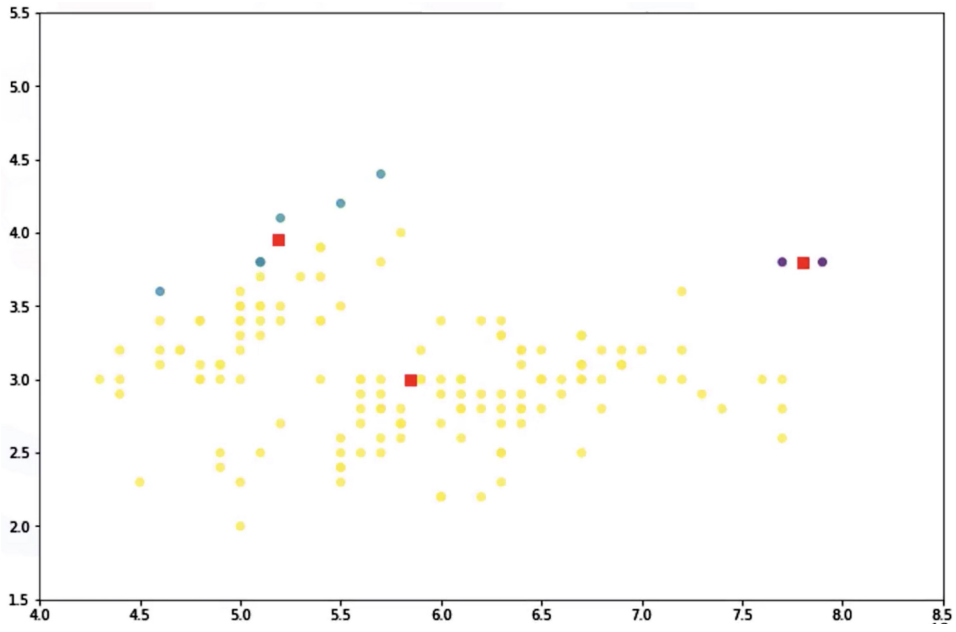


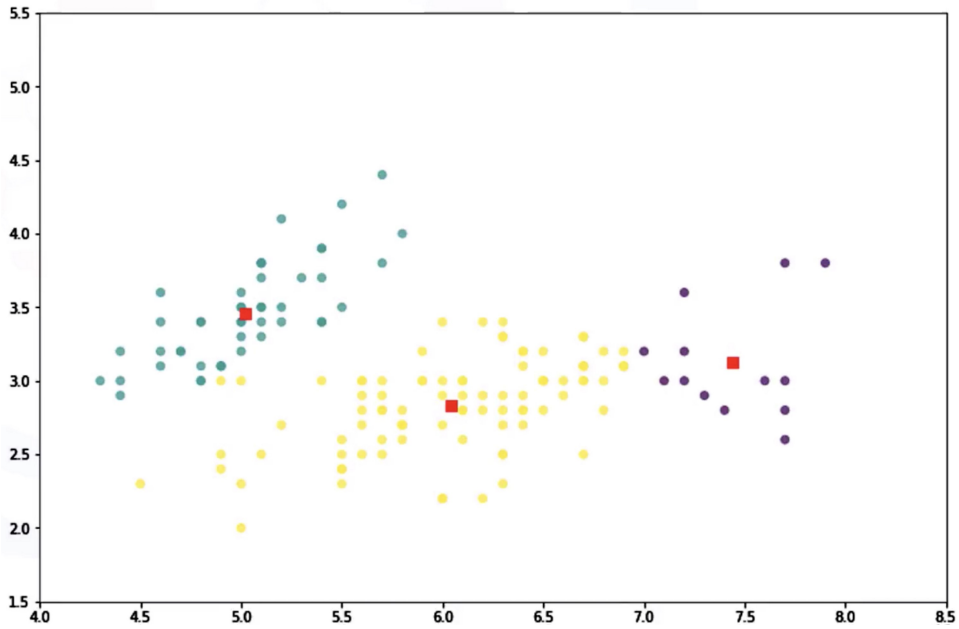
NOTE:

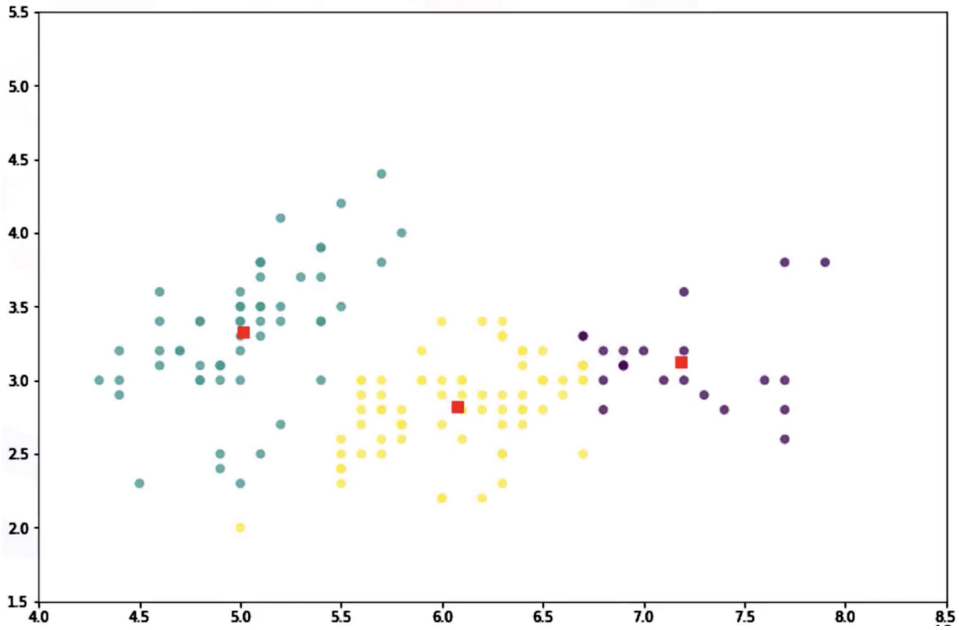
After moving the centroids we need to compute the Distance Matrix Again.

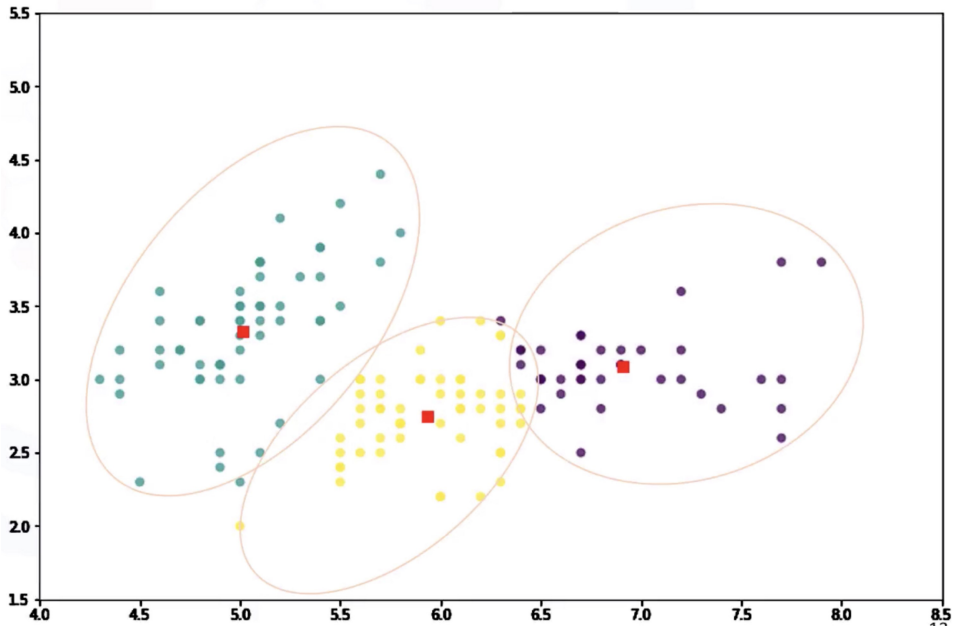
- **Step 5: Repeat the Process until no more change in the Centroids location**

Yes, its an iterative process (heuristic Algorithm)



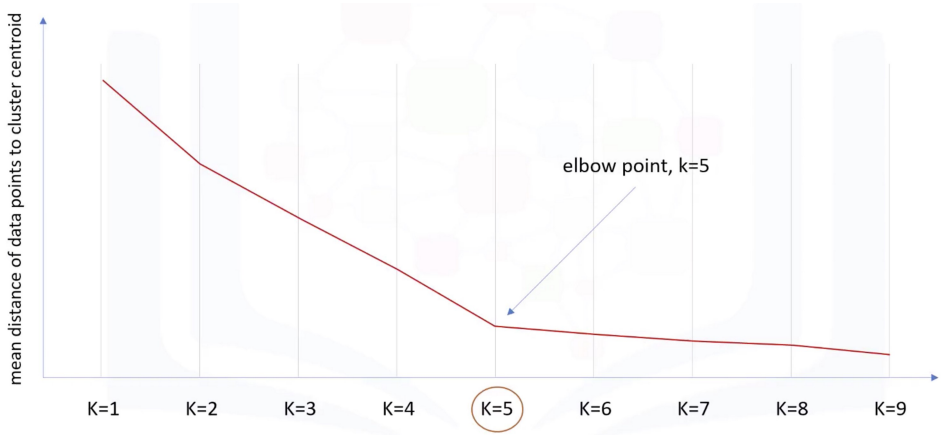

5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5


5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5


5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5

How to Initialize k value?

We check the mean distance of all datapoints in the cluster to its centroid for various k values
Note: that as we increase the k value, the accuracy increases (why?? as, mean distance bw points and centroid decreases) so, we need to pick the elbow point of accuracy for the k value

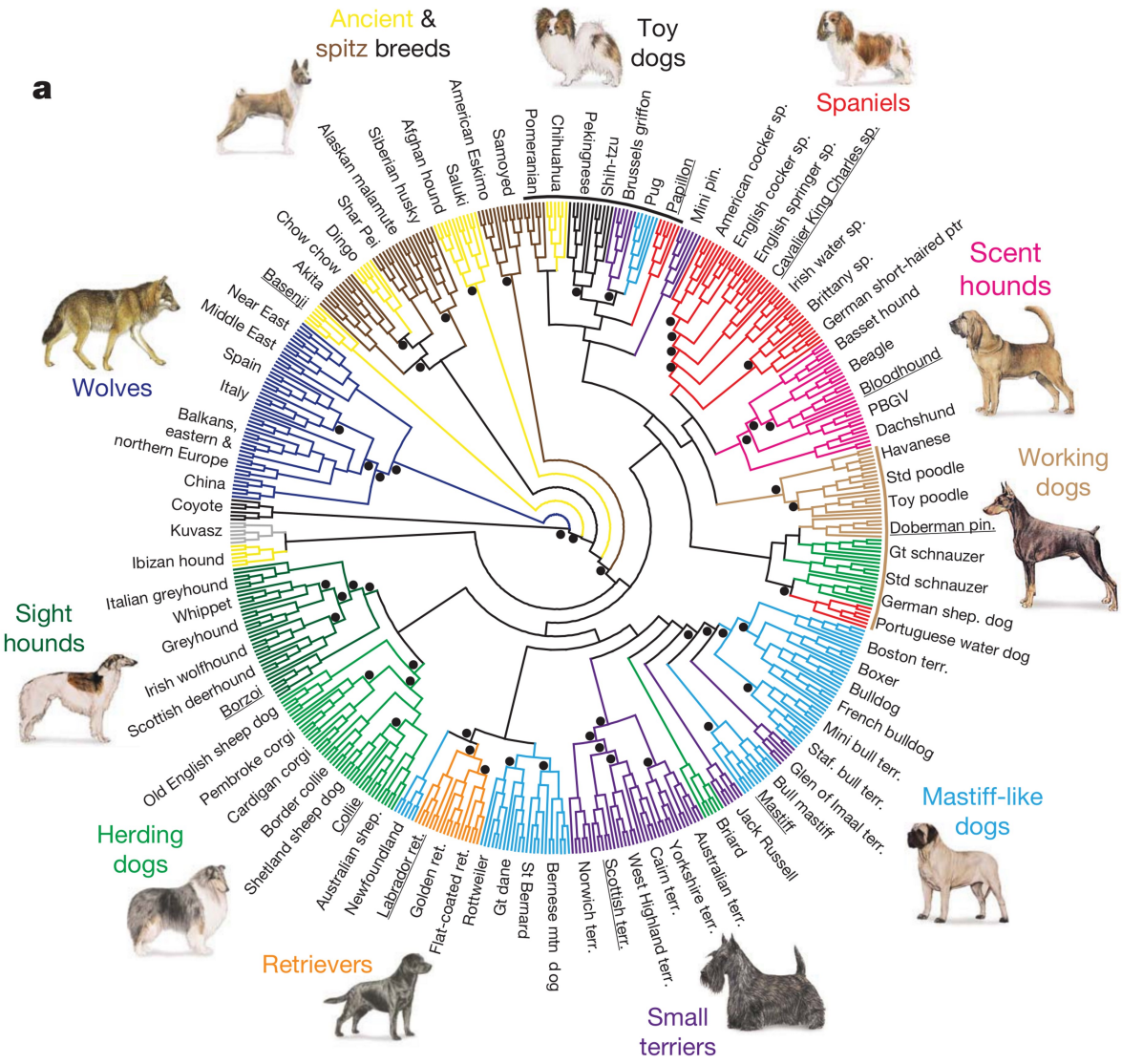


Agglomerative Clustering

Wednesday, 4 September 2019 08:31

Hierarchical Clustering

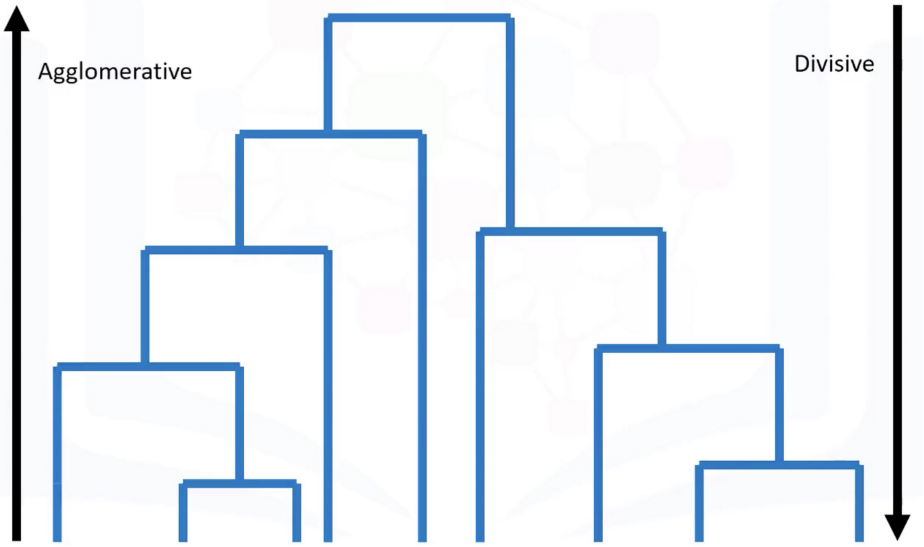
Dendrogram of Dogs n wolfs based on similarities of genetics



Hierarchical Clustering algorithm builds a hierarchy of clusters where each node is a cluster consists of clusters of its daughter nodes.

Types of Hierarchical clustering :

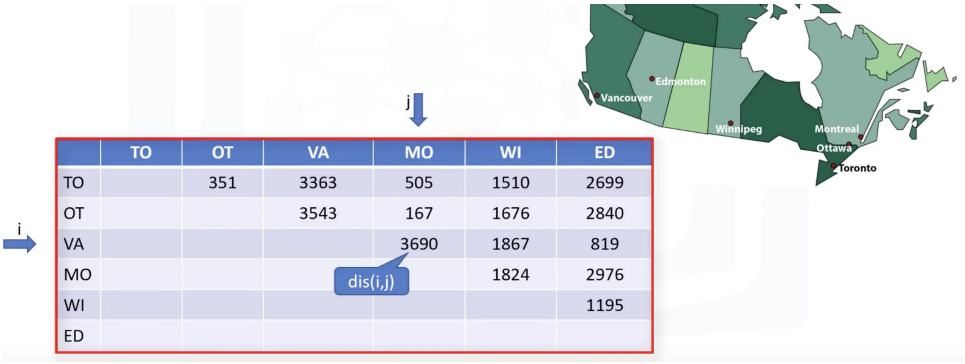
- **Agglomerative (Bottom up) - to collect**
- Divisive (top Down) - to divide



Agglomerative Hierarchical Clustering

Clustering 6 cities in Canada based on the distances between them.

- **Step 1: Create Distance Proximity Matrix for n clusters, each datapoint is a cluster**
    - Distance from each datapoint(city) is calculated to rest all datapoints(cities)
    - Distance can be calculated by **Euclidean Distance**
    - Each Data point from now on is called a Cluster, as its bottom up



j

i

	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO				dis(i,j)	1824	2976
WI						1195
ED						

- **Step 2: Merge the 2 closest Clusters**

TO OT MO VA ED WI

	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						

Merging the Clusters

TO OT MO VA ED WI

	TO	OT/MO	VA	WI	ED
TO		351	3363	1510	2699
OT/MO			3543	1676	2840
VA				1867	819
WI					1195
ED					

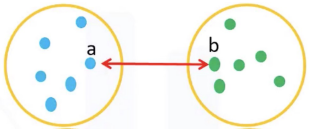
- **Step 3: Update the Distance Proximity Matrix as clusters are merged**

TO OT MO VA ED WI

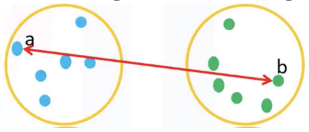
	TO	OT/MO	VA	WI	ED
TO		351	3363	1510	2699
OT/MO			3543	1676	2840
VA				1867	819
WI					1195
ED					

How to find distance between a newly formed cluster and the rest of the clusters? Aka what is the center of new cluster?

i. Single Linkage Cluster (shortest)

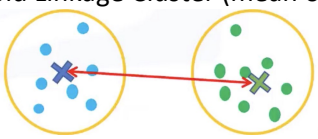


ii. Complete Linkage Cluster (longest)



iii. Average Linkage Cluster (mean)

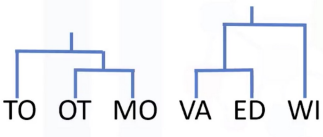
iv. Centroid Linkage Cluster (mean of features)



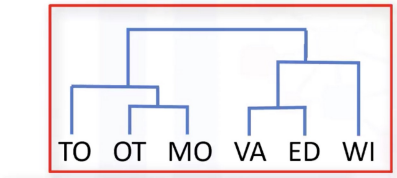
- **Step 4: Repeat from step 2 to find next smallest distance until there is one Cluster left**

	TO	OT/MO	VA	WI	ED
TO		351	3363	1510	2699
OT/MO			3543	1676	2840
VA				1867	819
WI					1195
ED					

	TO/OT/MO	VA	WI	ED
TO/OT/MO		3543	1676	2840
VA			1867	819
WI				1195
ED				

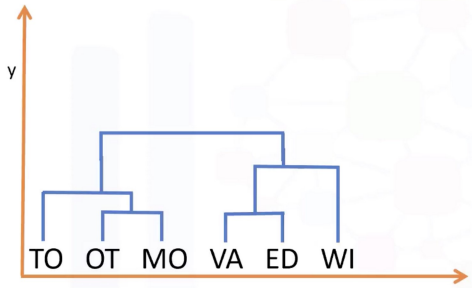


	TO/OT/MO	VA/ED	WI
TO/OT/MO		2840	1676
VA/ED			1667
WI			



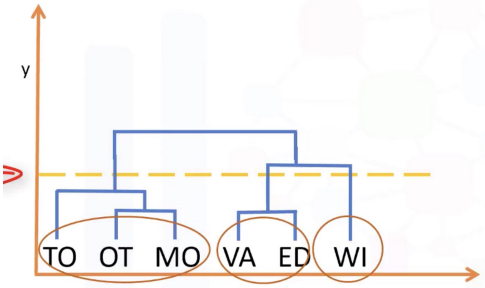
	TO/OT/MO	VA/ED/WI
TO/OT/MO		1676
VA/ED/WI		

Dendrogram



Dendrogram

We cut the dendrogram to get clusters at that point



Dendrogram

Note

- We can reconstruct the history by looking at the dendrogram
- Number of clusters (k) is not needed to be specified as each datapoint is considered as an individual cluster.

Advantages of Hierarchical Clustering

- No need to specify required number of clusters
- Easy to implement
- Produces Dendrogram, which helps to understand the data

Disadvantages of Hierarchical Clustering

- Generally has long runtimes
- Sometimes difficult to identify number of clusters by the dendrogram
- Can never undo any previous steps

Hierarchical Clustering vs K Means

K Means	Hierarchical Clustering
Much more efficient	Slow for Large Datasets
Number of clusters should be specified	No need to specify number of clusters
Gives only 1 partitioning of the data based on k	Gives more than one partitioning depending on resolution
Potentially returns different clusters each time it is run due to random initialization of centroids	Always generates the same clusters

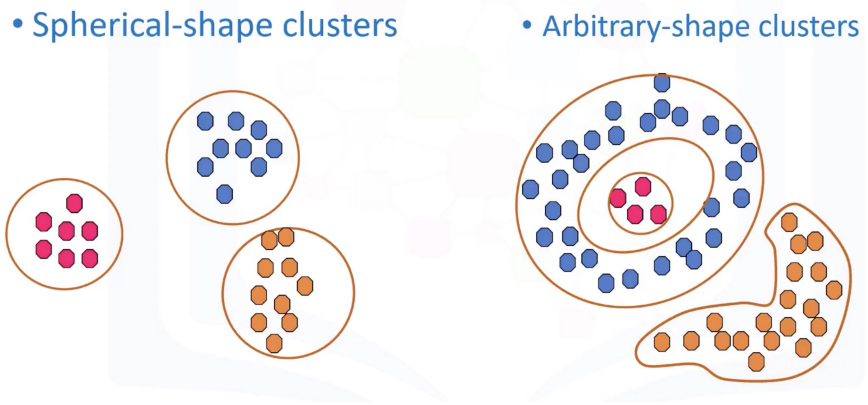
DBSCAN

Wednesday, 4 September 2019 09:20

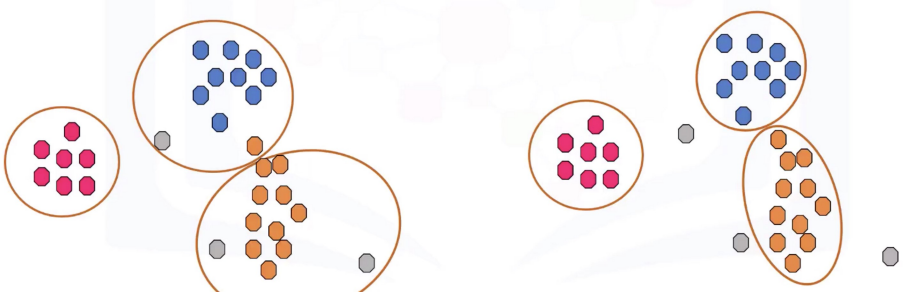
DBSCAN - Density Based Special Clustering of Applications with Noise

K Means and Agglomerative clustering has the following Drawbacks:

- Can't cluster Arbitrary Shape Clusters



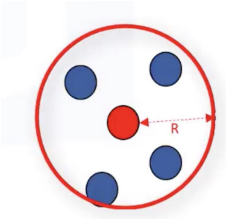
- Outliers are also clustered into a cluster



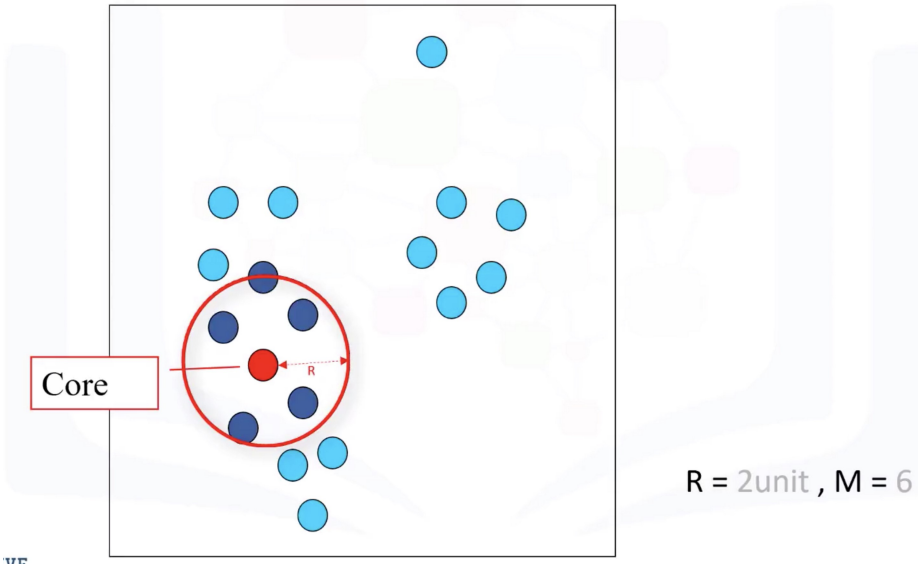
Hence we need to classify clusters based on Density of datapoints to ignore the outliers, instead of considering all the datapoints (as in k means and agglomerative clustering)

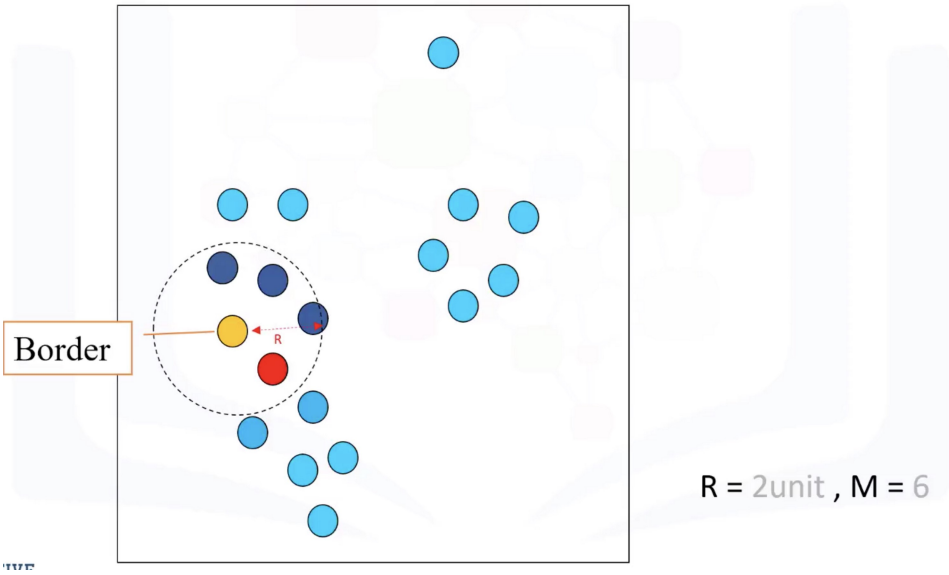
How to Cluster Based on Density of Datapoints? DBSCAN Algorithm

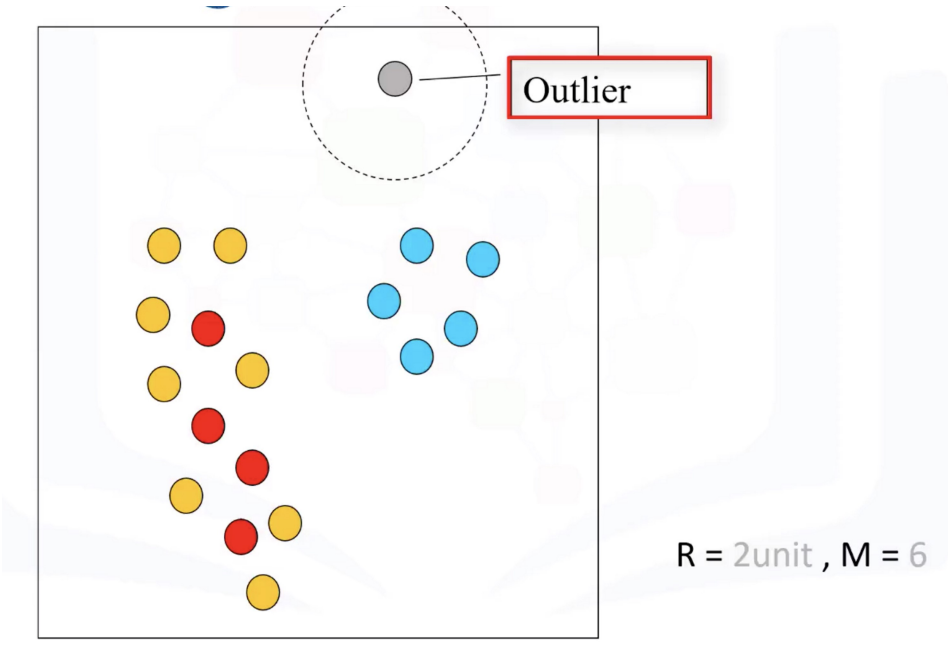
- **We use 2 parameters:**
  - R - Radius of Neighborhood
  - M - Minimum number of Neighbors



- **Step 1: Each Datapoint is Categorized as**
  - Core Point - in R there are at least M Points
  - Border Point - in R there are less than M points with a Core point
  - Outlier Point - in R there is no other points

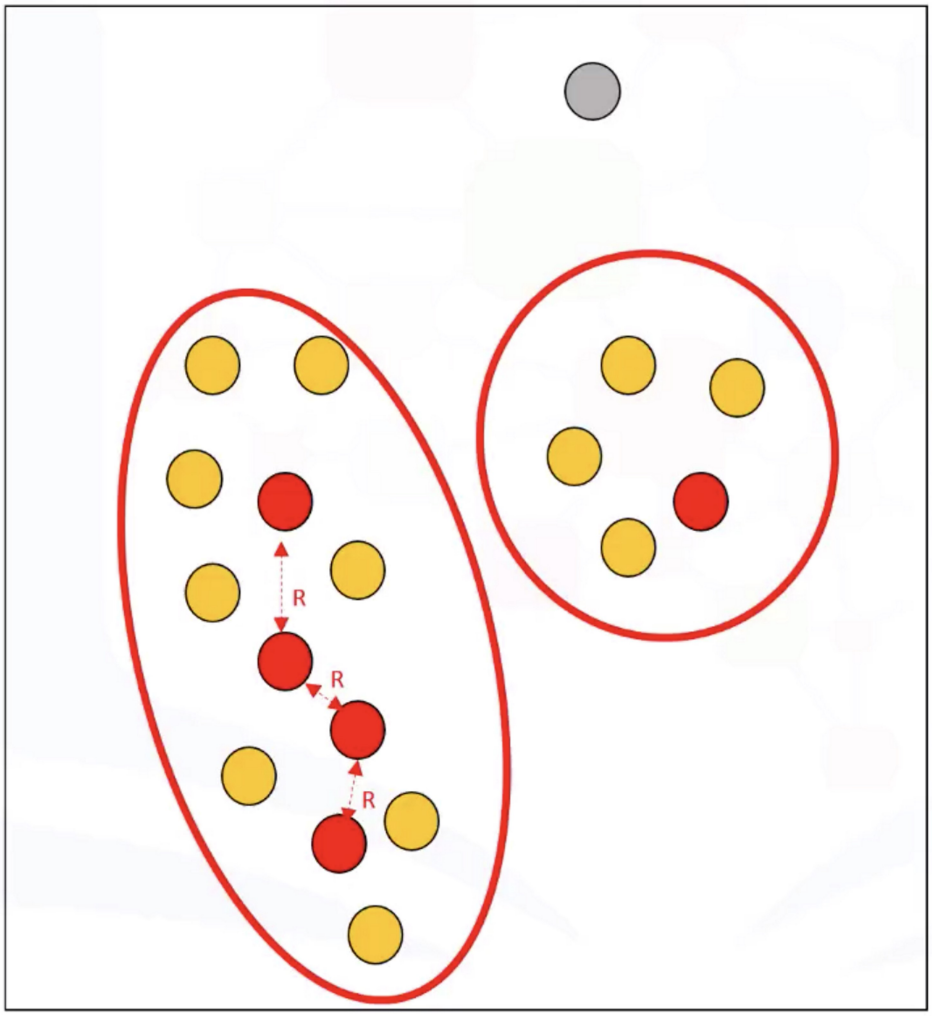






- **Step 2: Clusters are formed based on core points**

  For each core points in proximity(R) of other core points belongs to the same cluster



Advantages of DBSCAN

- Can find Arbitrary Shaped Clusters
- Robust for outliers
- No need to specify number of clusters

Module 5 - Recommender Systems

Thursday, 5 September 2019 04:40

Recommender Systems or Recommendation System

Captures the pattern of peoples' behavior and choices and use it to predict what else they might want or like.

Applications of Recommender Systems

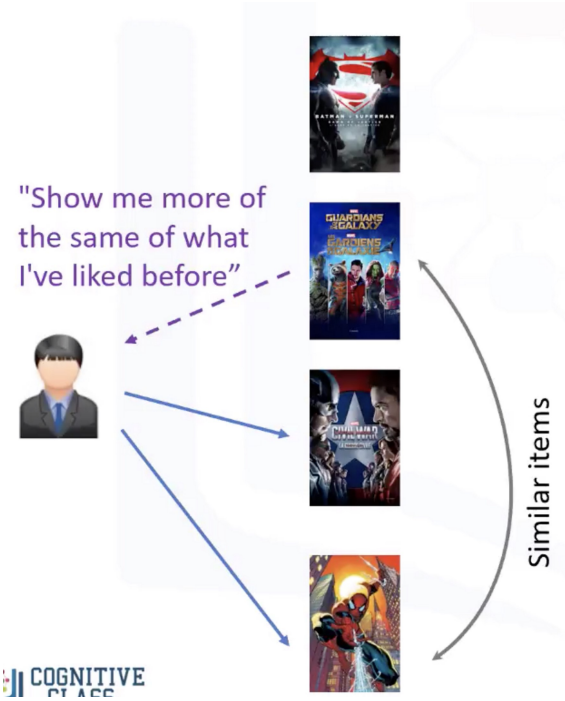
- What to buy?
  - Amazon - Books recommendation
  - Netflix - Movies recommendation
- Where to eat?
- Which job to apply to?
  - Linkedin
- Who you should firend with?
  - Facebook - friend suggestion
- Personalize your experience on the web
  - Google News
  - Youtube

Advantages of recommender System

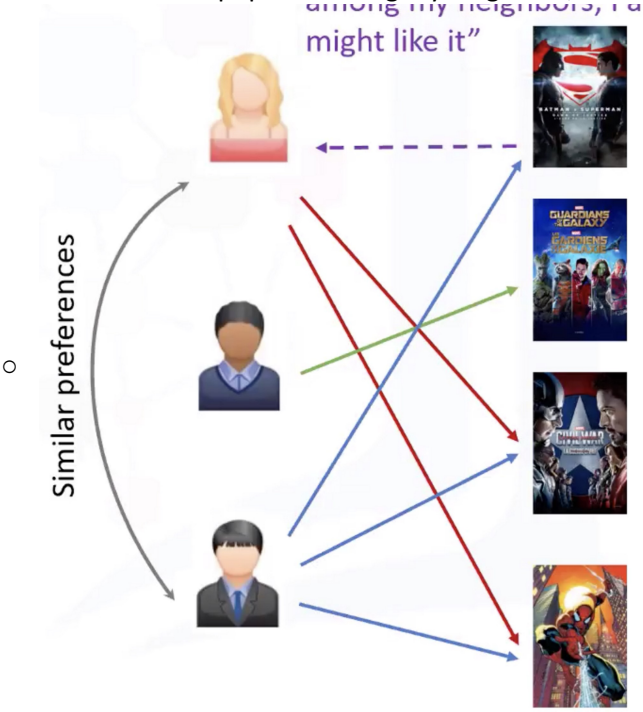
- Customers get Broader Exposure
- Possibility of continual usage or purchase of products (getting hooked)
- Better Experience for customers as well as the service providers.

Types of Recommender Systems

1. **Content Based**
   - Show me more of the same what I have liked before



2. **Collaborative Filtering**
   - Tell me what's popular among my neighbors, I also might like it



3. **Hybrid Recommender Systems**
   - Combination of multiple techniques

Implementation of Recommender System

1. **Memory Based**
   - Uses entire user-item dataset to generate a recommendation
   - Uses statistical techniques to approximate users or items
   - Example: Pearson Correlation, Cosine Similarity, Euclidean Distance

2. **Model Based**
    - Develops a model of users In an attempt to learn their preferences
    - Models can be created using Machine Learning Techniques
    - Example: Regression, classification, clustering

Content Based Recommender Systems

Thursday, 5 September 2019 05:09

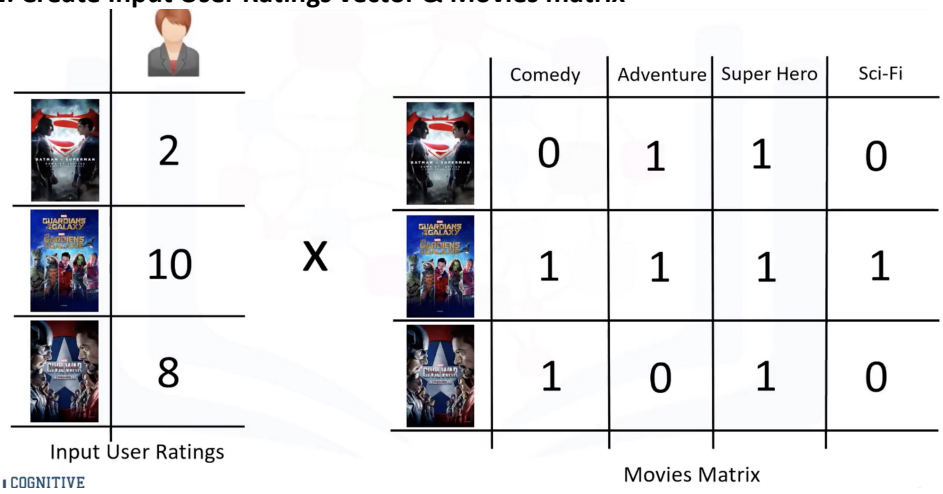
Content Based Recommender System

- Its also Called **Item item Recommendation System.**
- Recommendation based on similarity of other items based on users previously liked items
- Memory based Implementation

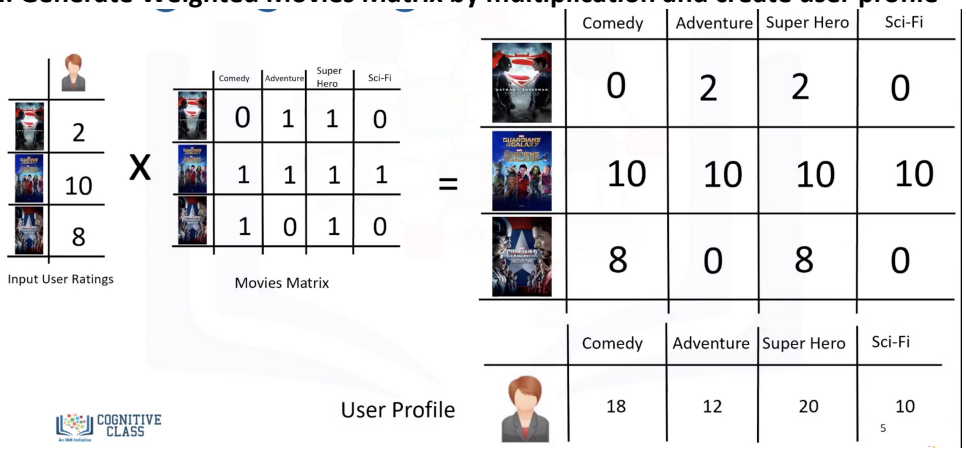
Algorithm for Content Based Recommender System

Here we are supposed to Recommend movies based on her previously rated movies

- **Step 1: Create Input User Ratings vector & Movies matrix**



- **Step 2: Generate Weighted Movies Matrix by multiplication and create user profile**



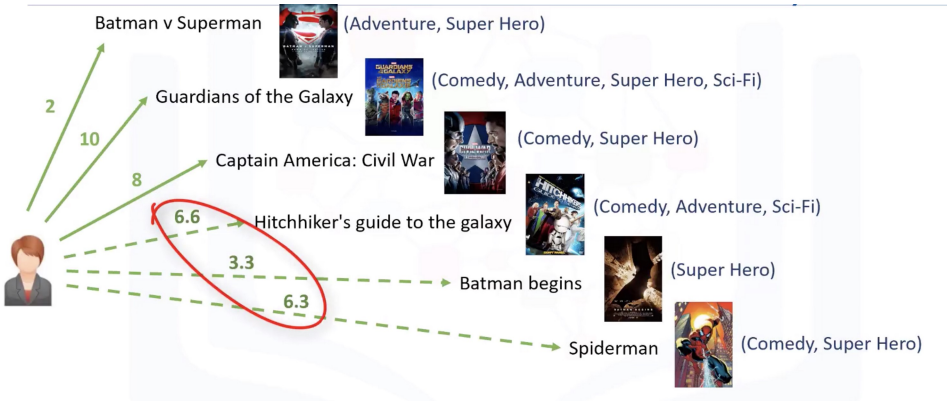
User profile is aggregate of columns

- **Step 3: Normalize the user Profile**

- **Step 4: now create Candidate Movies Matrix**

- **Step 5: Generating Candidate Weighted Movie Matrix to get Recommendation Vecotr**

Recommendation Matrix is aggregate of rows



Disadvantage of Content Based Recommender System

- Cant recommend movies in other genre like 'drama'

- Doesn't take into account what others think of the item, so low quality item recommendations might happen
- Extracting data is not always intuitive
- Determining what characteristics of the item the user dislikes or likes is not always obvious

Collaborative Filtering

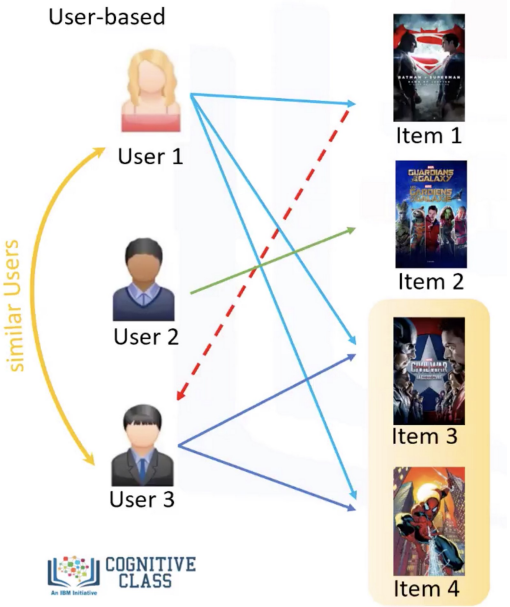
Thursday, 5 September 2019 14:16

Collaborative Filtering

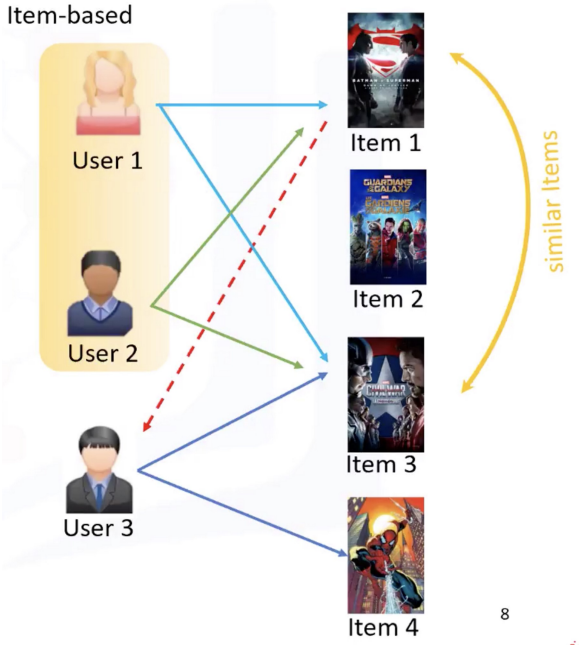
Is based on the fact that relationships exist between the products and peoples interests

There are 2 approaches for Collaborative Filtering

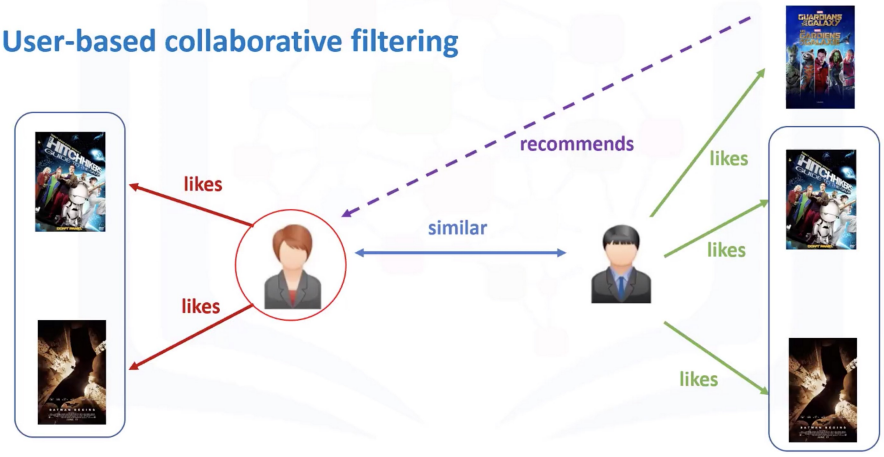
- **User based Collaborative filtering**
  - Based on similarities of users or neighborhood



- **Item based Collaborative Filtering**
  - Based on similar Items



User Based Collaborative Filtering Algorithm



- **Step 1: Construct user Rating matrix**

	9	6	8	4	
	2	10	6		8
	5	9		10	7
	?	10	7	8	?

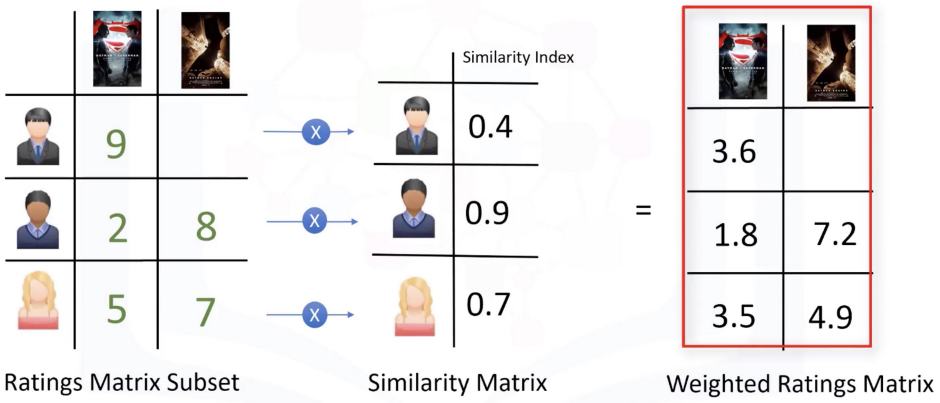
Ratings Matrix

- **Step 2: Find Similarity between users (similarity Matrix)**

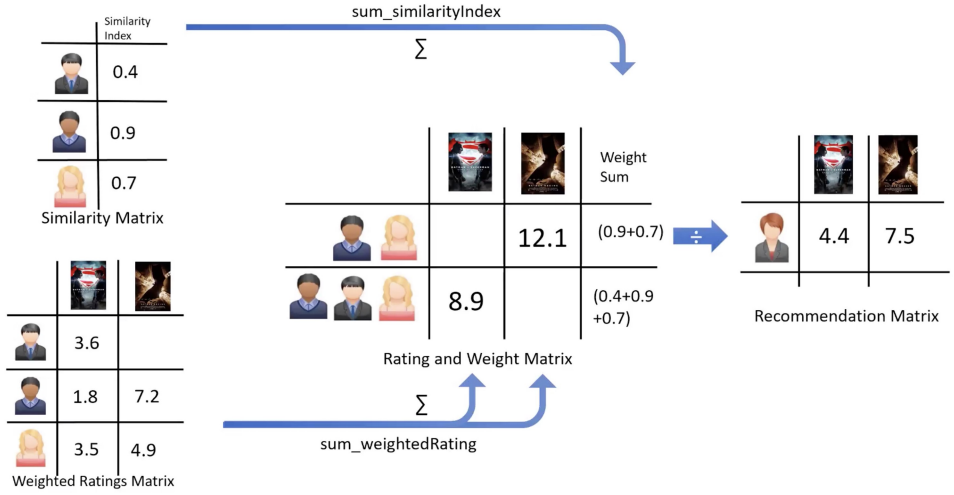
  Similarity Between users could be found using Euclidian Distances, person Correlation , cosine Similarity. Assume we get similarity between them by some means as:

- **Step 3: Create Weighted Ratings Matrix**

  Multiplying candidate Rating Matrix(movies rates by other users which are not watched by active user) with Similarity Matrix



- **Step 4: Creating the Recommendation Matrix**



Challenges of Collaborative Filtering

- Data Sparsity
    - Data is large but people rate very little about the items making it hard to recommend
- Cold Start
    - Difficulty to recommend a new user or a new Item as it will have null ratings to find similarity with some others
- Scalability
    - When there are so many users or items, there will be decrease in performance as there will be so much similarity between items or users.

How to overcome this?

Hybrid based Recommend System

Important codes

Wednesday, 4 September 2019 21:14

Methods to Normalize

- To scale in standard curve

```
from sklearn import preprocessing
x = preprocessing.StandardScaler().fit(x).transform(x)
```

- To scale bw 0 to 1

```
from sklearn import preprocessing
x = preprocessing.min_max_scaler().fit(x).transfrom(x)
```

```
data = data[pd.to_numeric(data['BareNuc'], errors='coerce').notnull()]  # If 'coerce', then invalid parsing will be set as NaN
data['BareNuc'] = data['BareNuc'].astype(int)
print('BareNuc', np.unique(data['BareNuc']))
```