یادگیری ماشین با پایتون

مقدمه‌ای عملی

۱۴۰۰ / جادی برای مکتب‌خونه

یادگیری ماشین و پایتون

- بر اساس دوره IBM در edx با آموزشگری سعید آقابزرگی، دانشمند داده آی.بی.ام.

Welcome

- We are going to talk about ML and how it helps in different areas (loans, segmentations, medicine, recommendations, …)
- We will use python libraries to create models, say building a model to estimate the CO2 emission of the cars using scikit learn or will predict customer churn
  - All codes are provided in notepad / jupyther
  - After this course you will have new skills like regression, classification, clustering, scikit learn, numpy, pandas AND new projects specially if you start working on the datasets you can freely available on the internet.

Intro

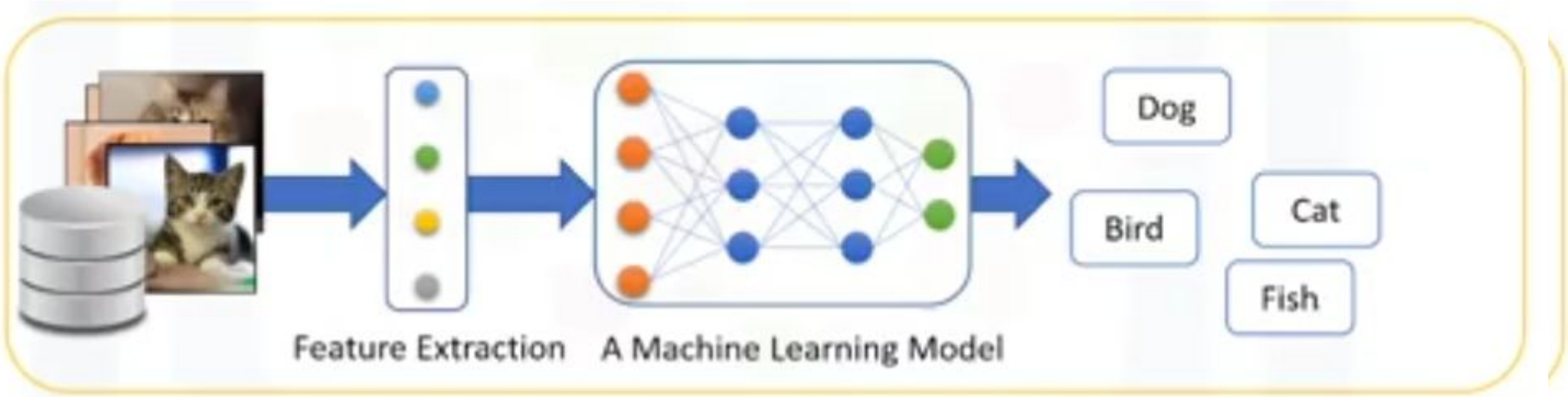
- Say we want to understand if this is a fraud trx or not, is this a malignant or benign cell, what should I show next to this customer, …
- This can be done by ML by looking at some characteristics of data.
  - clean the data
  - select the proper algorithm
  - train the model
  - predict new cases

Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
5	1	1	1	2	1	3	1	1	benign
5	4	4	5	7	10	3	2	1	benign
3	1	1	1	2	2	3	1	1	malignant
6	8	8	1	3	4	3	7	1	benign
4	1	1	3	2	1	3	1	1	benign
8	10	10	8	7	10		7	1	malignant
1	1	1	1	2	10	3	1	1	benign
2	1	2	H	2	1	3	1	1	benign
2	1	1	1	2	1	1	1	5	benign
4	2	1	1	2	1	2	1	1	benign

Intro

Machine Learning is the subfield of computer science that gives “computers the ability to learn without being explicitly programmed”

- Arthur Samuel who coined the phrase in 1959



Intro

Examples:

- CO2 emission (Regression)
- Is this cancer? (Classification)
- Bank Loans (Clustering)
- Anomaly detection (credit card fraud)
- Netflix recommendations (recommenders)

Intro

AI (mimics human I)

- Computer vision
- Language Processing
- Creativity

ML (Subset of AI, more statistical)

- Classification
- Clustering
- Neural Network

Revolution in ML (Special field of ML)

- Deep Learning

Intro

Python

- You should know the basics
- It is easy
- Libraries like Numpy & Pandas + SciKit Learn are used with a quick intro
- We are python dependent. You can do with anything else.. but why? :D

Supervised vs. Unsupervised

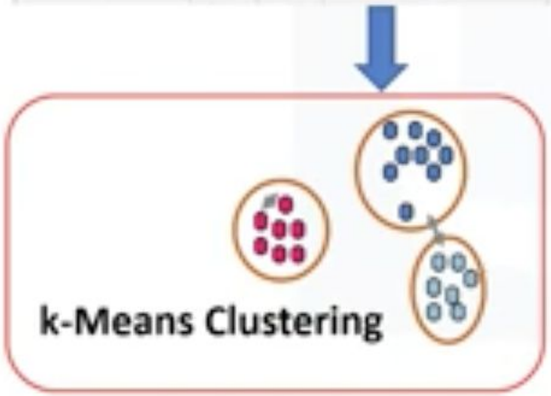
- Supervised: we “teach the model” by labeling & just then, the model can predict the unknown or future instances

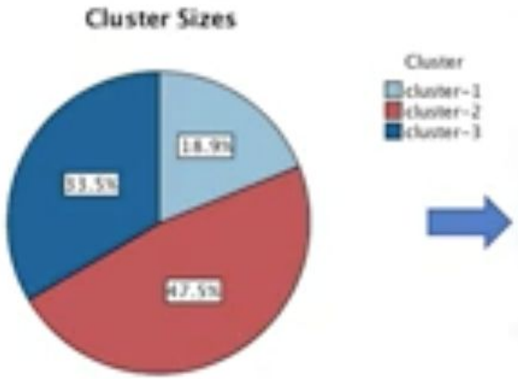
Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
5	1	1	1	2	1	3	1	1	benign
5	4	4	5	7	10	3	2	1	benign
3	1	1	1	2	2	3	1	1	malignant
6	8	8	1	3	4	3	7	1	benign
4	1	1	3	2	1	3	1	1	benign
8	10	10	8	7	10		7	1	malignant
1	1	1	1	2	10	3	1	1	benign
2	1	2	H	2	1	3	1	1	benign
2	1	1	1	2	1	1	1	5	benign
4	2	1	1	2	1	2	1	1	benign

Supervised vs. Unsupervised

- UnSupervised: Model works on its own to discover information.

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

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

Supervised vs. Unsupervised

Supervised Learning

- **Classification:**
  Classifies labeled data
- **Regression:**
  Predicts trends using previous labeled data
- Has more evaluation methods than unsupervised learning
- Controlled environment

Unsupervised Learning

- **Clustering:**
  Finds patterns and groupings from unlabeled data
- Has fewer evaluation methods than supervised learning
- Less controlled environment

رگرسیون

Regression Intro

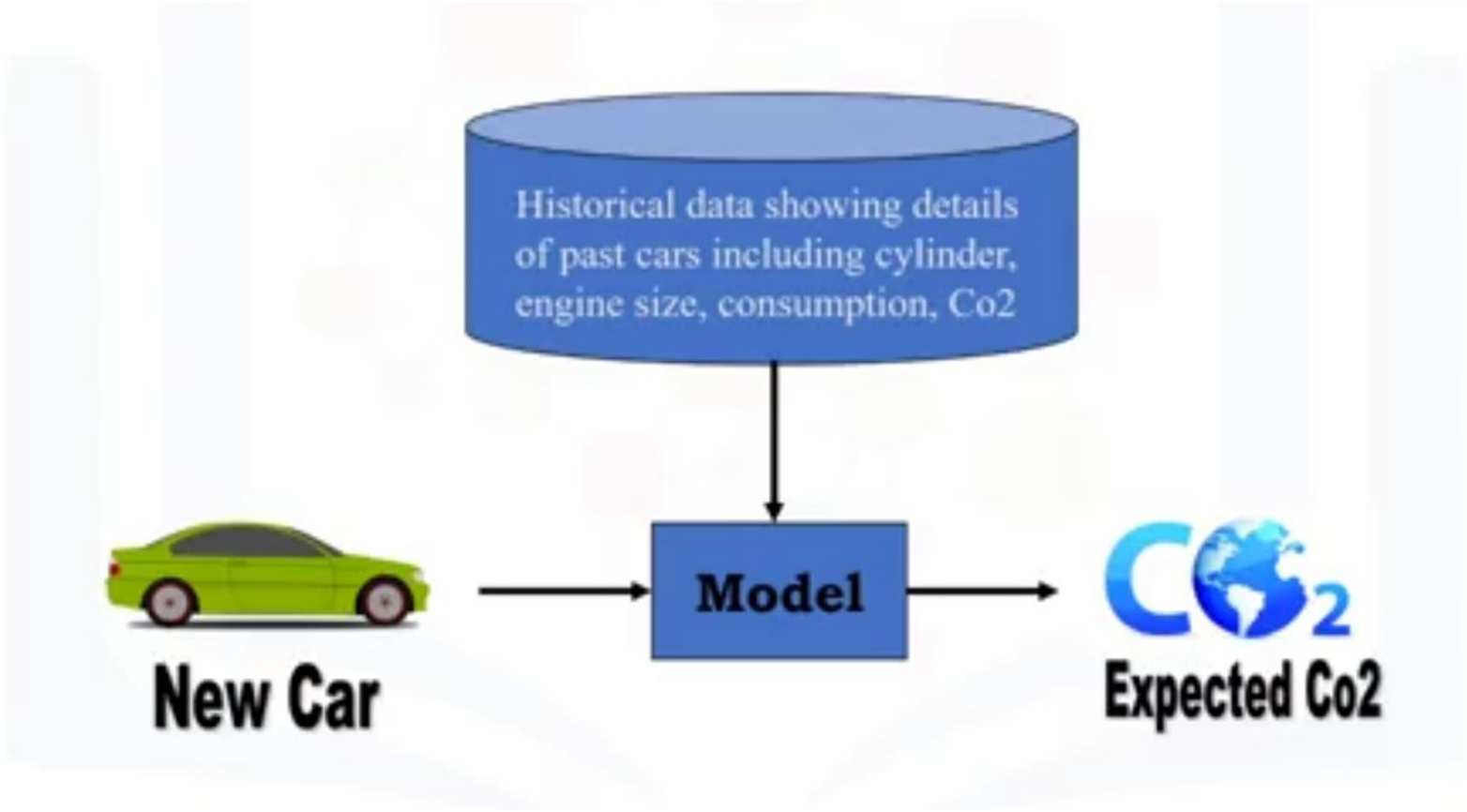
- regression is the process of predicting a continuous value
- Independent (x, desc, ...) vs Dependent (y, goal, prediction, ...) variables
- y is continuous

[5]:

	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

Regression Intro

Model



Regression Intro

Types

- Simple (only one independent)
  - Linear
  - Non-Linear
- Multiple (multiple independent)
  - Linear
  - Non-Linear

Regression Intro

Samples

- Household Price
- Customer Satisfaction
- Sales Forecast
- Employment Income

Regression Intro

Algorithms

- Ordinal
- Poisson
- Fast Forest quantile
- Linear, Polynominal, Lasso, Stepwise, Ridge
- Bayesian Linear
- Nerural Network
- Decision Forest
- Boosted decision tree
- K-nearest neighbors

Simple Linear Regression

- Can we predict co2 emission from one of the independents (this is why we call it Simple)
- Lets try engine size...

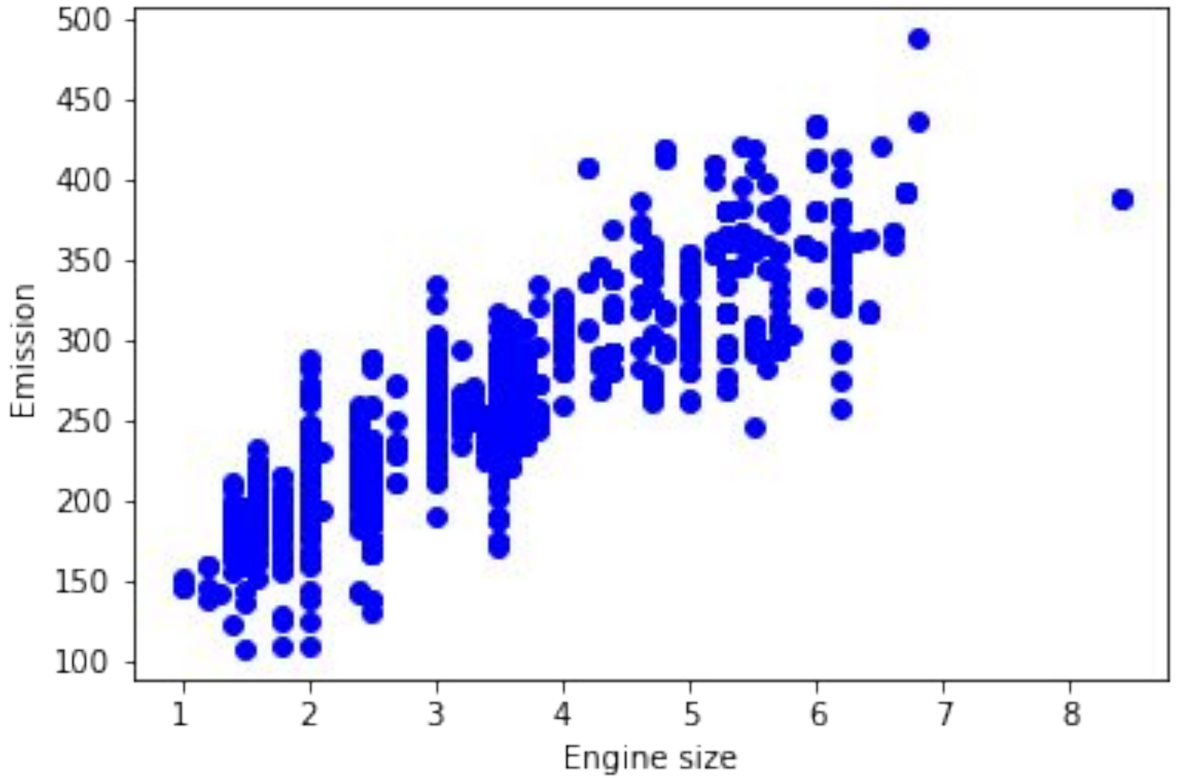
[5]:

	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

Simple Linear Regression

$\hat{y} = \theta_0 + \theta_1 x_1$ ship is obvious

- There is a line, we assume a straight line
- we can predict an emission for say, a car with 2.4
- y hat is the dependent variable of the predicted value.
- x1 is the independent variable.
- Theta 0 and theta 1 are the parameters of the line
- Theta 1 is known as the slope or gradient of the fitting line and theta 0 is known as the intercept.
- Theta 0 and theta 1 are also called the coefficients of the linear equation.

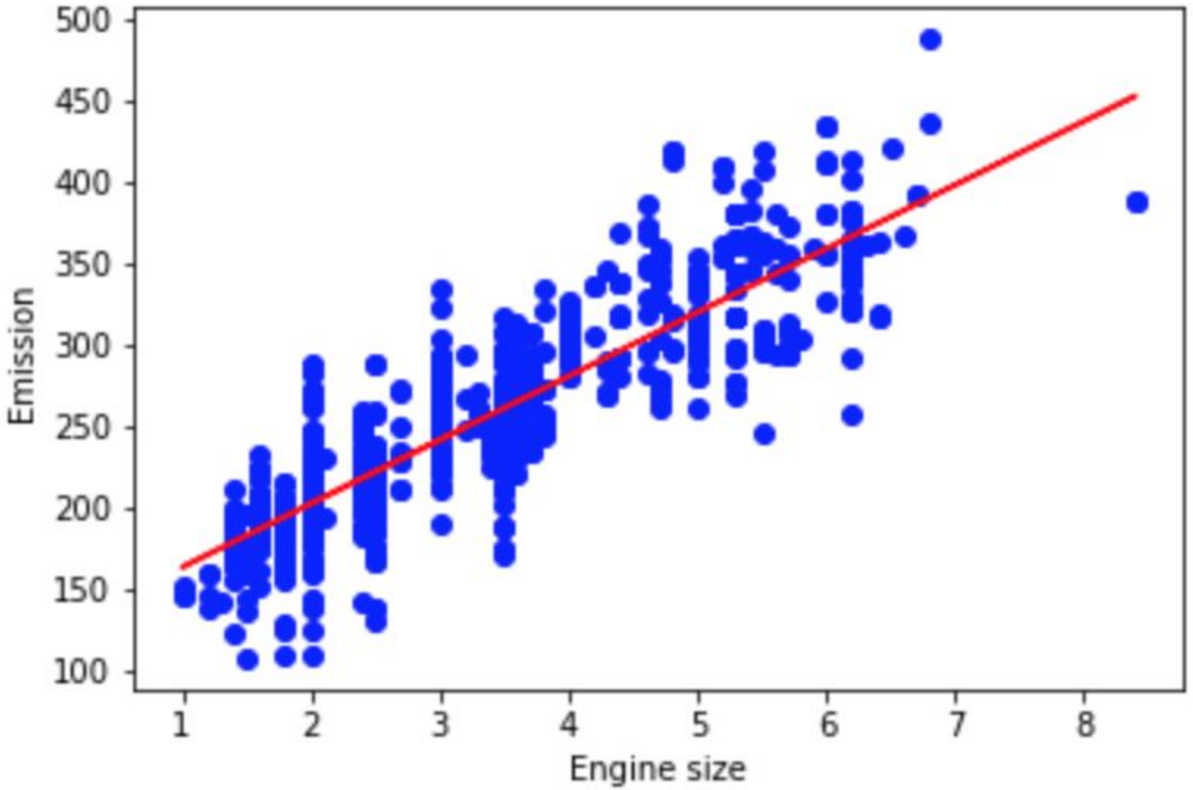


Simple Linear Regression

MSE

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

or each point is the prediction from the actual point. 3) Mean Square Error (MSE should be minimized)



- Minimum MSE can be achieved with two methods: Math or Optimization

Simple Linear Regression

MSE (Math)

	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

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

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

$\bar{x} = (2.0 + 2.4 + 1.5 + \dots)/9 = 3.03$

$\bar{y} = (196 + 221 + 136 + \dots)/9 = 226.22$

$$\theta_1 = \frac{(2.0 - 3.03)(196 - 226.22) + (2.4 - 3.03)(221 - 226.22) + \dots}{(2.0 - 3.03)^2 + (2.4 - 3.03)^2 + \dots}$$

$\theta_1 = 39$

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

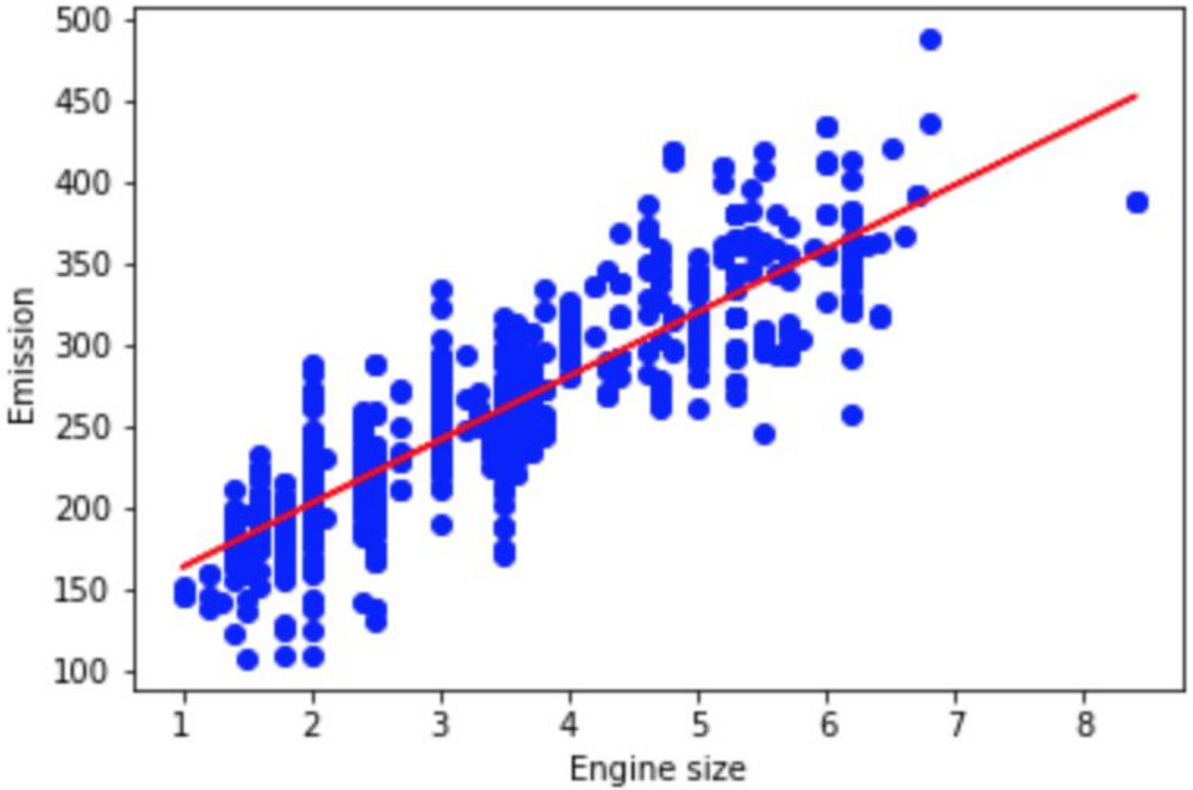
$\theta_0 = 226.22 - 39 * 3.03$

$\theta_0 = 125.74$

Simple Linear Regression

Pros

- Very Fast
- Easy to understand and interpret
- No need for parameter tuning (say like in KNN)



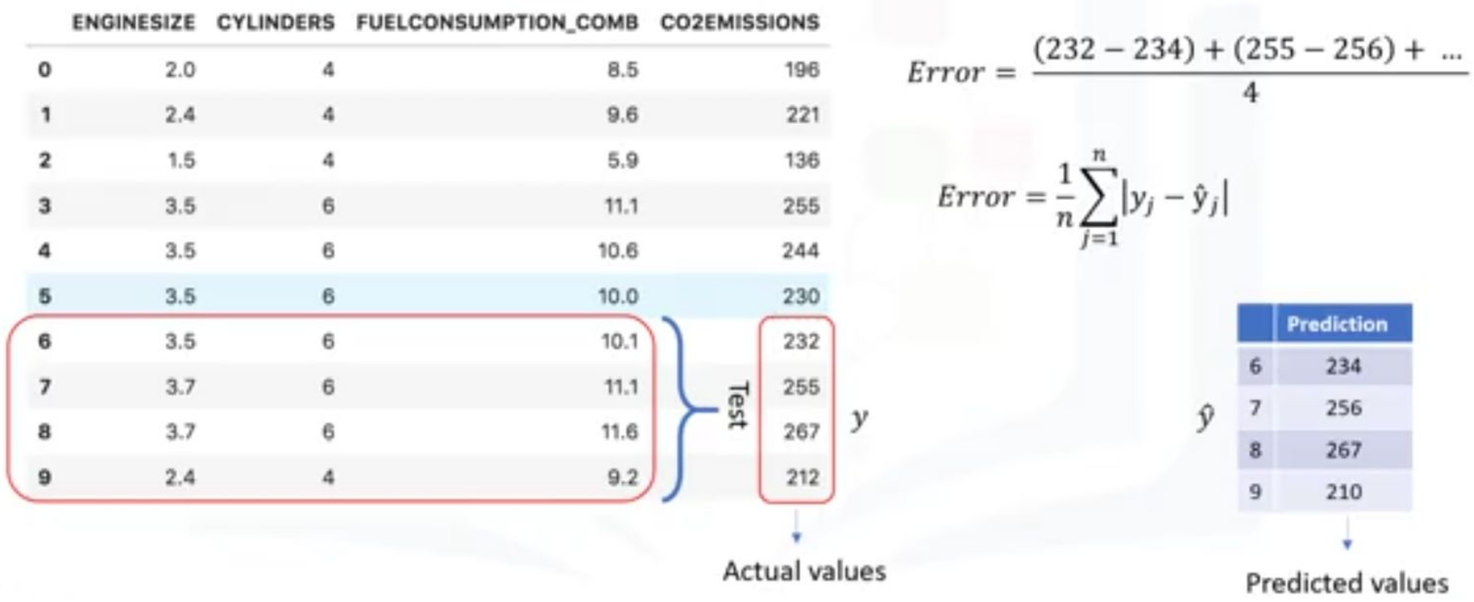
Model Evaluation

- goal is to build a model to accurately predict an unknown case.
- You need to evaluate to see how much you can trust your model/prediction
- Two main methods:
  - Train and Test on Same data
  - Train / Test split
- Regression Evaluation Metrics

Model Evaluation

Train and Test on Same data

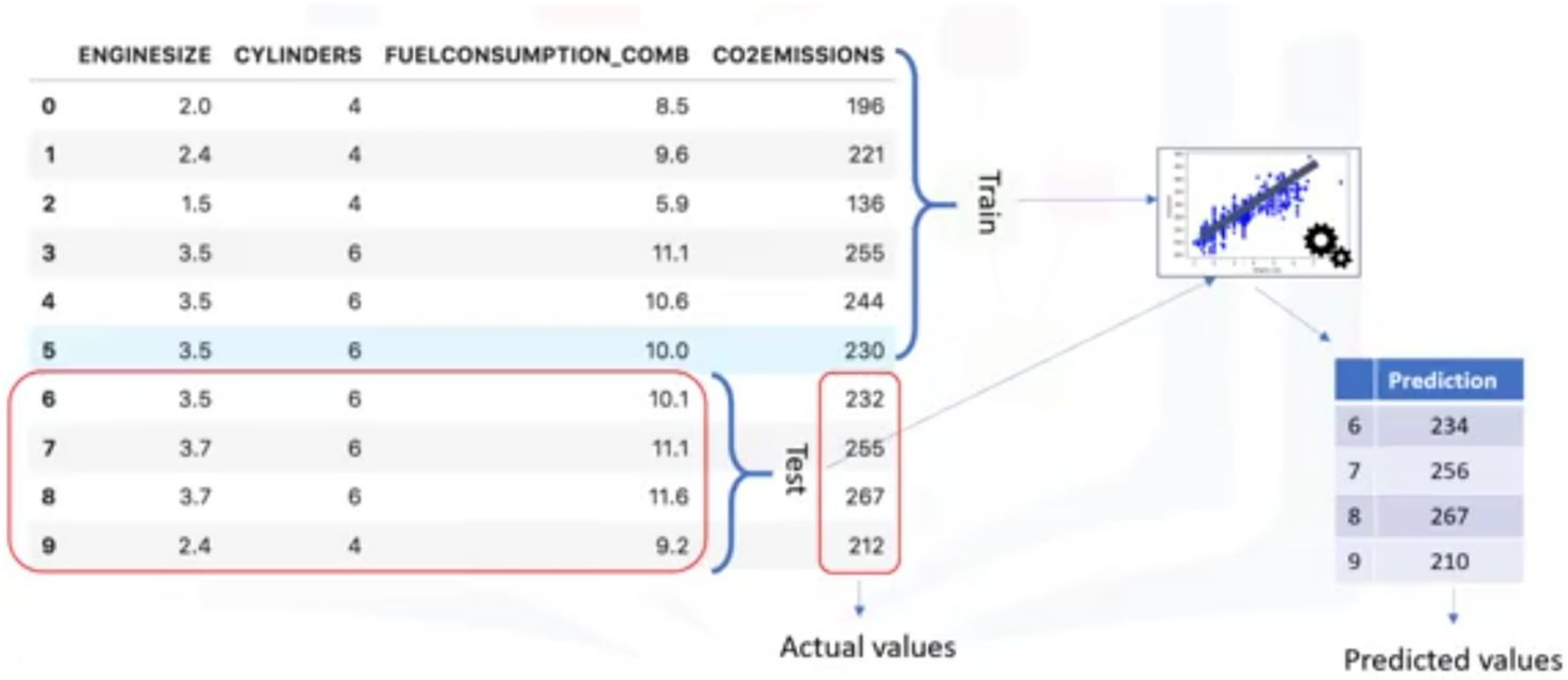
- High "training accuracy"
  - not always good
  - overfitting the data (say capture noise and produce non generalized model)
- Low "out of sample accuracy"
  - Important to have



Model Evaluation

Train/Test split

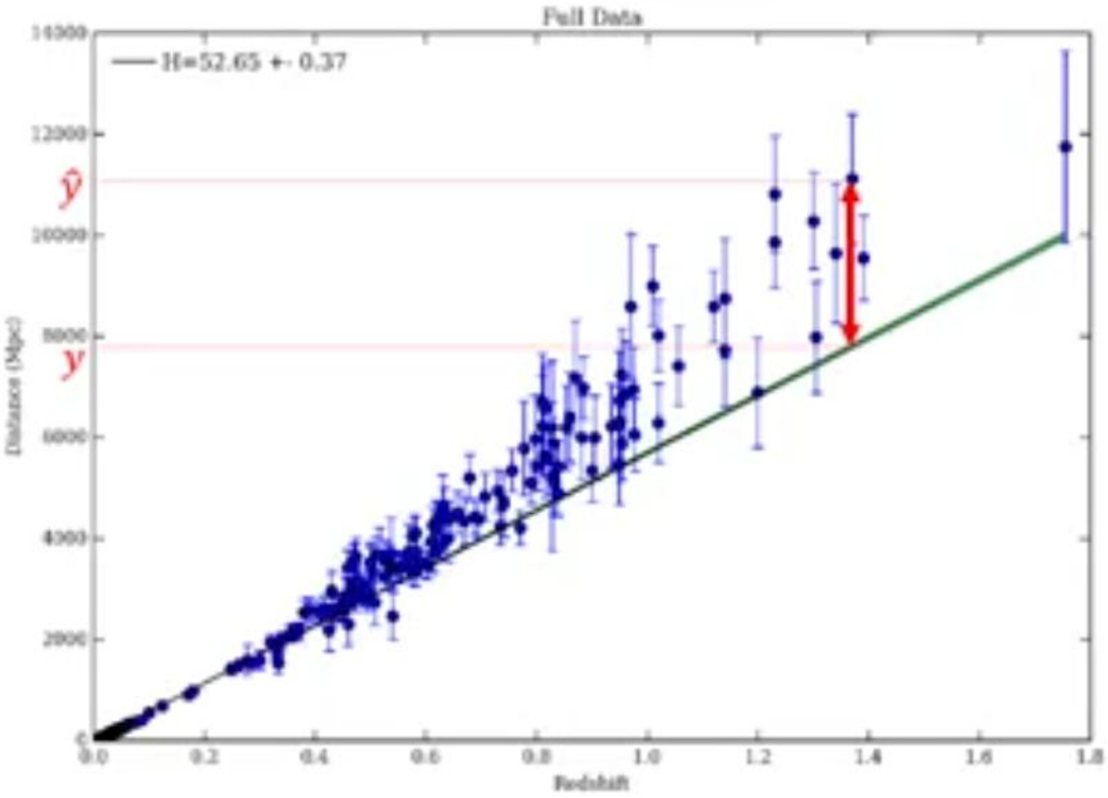
- Mutually exclusive split
- More accurate on out-of-sample
- ensure that you train your model with the testing set afterwards, as you don't want to lose potentially valuable data.
- Dependent on which datasets the data is trained and tested



Model Evaluation

Evaluation Matrix

- used to explain the performance of a model
- say comparing actual with predicted
- error of the model is the difference between the data points and the trend line generated by the algorithm
- There different metrics (next slide) but the choice is based on the model, data type, domain, ...



Model Evaluation

Errors

- mean absolute error (MAE)
- mean squared error (MSE)
- root mean squared error (RMSE); interpretable in the same units as the response vector or y units
- Relative absolute error, also known as residual sum of square (RAE)
- Relative squared error (RSE)
- R2; Popular metric for the accuracy of your model. represents how close the data values are to the fitted regression line. The higher the better

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

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

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

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

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

$$R^2 = 1 - RSE$$

Lets see some libraries!

- Notebook
- Numpy
- Matplotlib
- pandas

Lab: Simple Linear Regression

- ML0101EN-Reg-Simple-Linear-Regression-Co2.ipynb

Multiple Linear Regression

- Simple / Multiple
- kind of same as simple
- usages:
  - find the strength of each independent variable
  - predict the impact of the change on one of the independent variables

Multiple Linear Regression

Formula

$$Co2\ Em = \theta_0 + \theta_1 Engine\ size + \theta_2 Cylinders + \ ...$$

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ ... + \theta_n x_n$$

$$\hat{y} = \theta^T X$$

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

X: Independent variable

Y: Dependent variable

	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	?

Multiple Linear Regression

Finding parameters

- Again we can find the MSE
- the best model is the one with he minimized MSE
- The method is called Ordinary Least square
  - linear algebra
  - slow! for less than 10K samples
- Optimization Algorithms
  - Gradient Descent (Starts with random, then changes in multiple iterations)

Multiple Linear Regression

Some notes

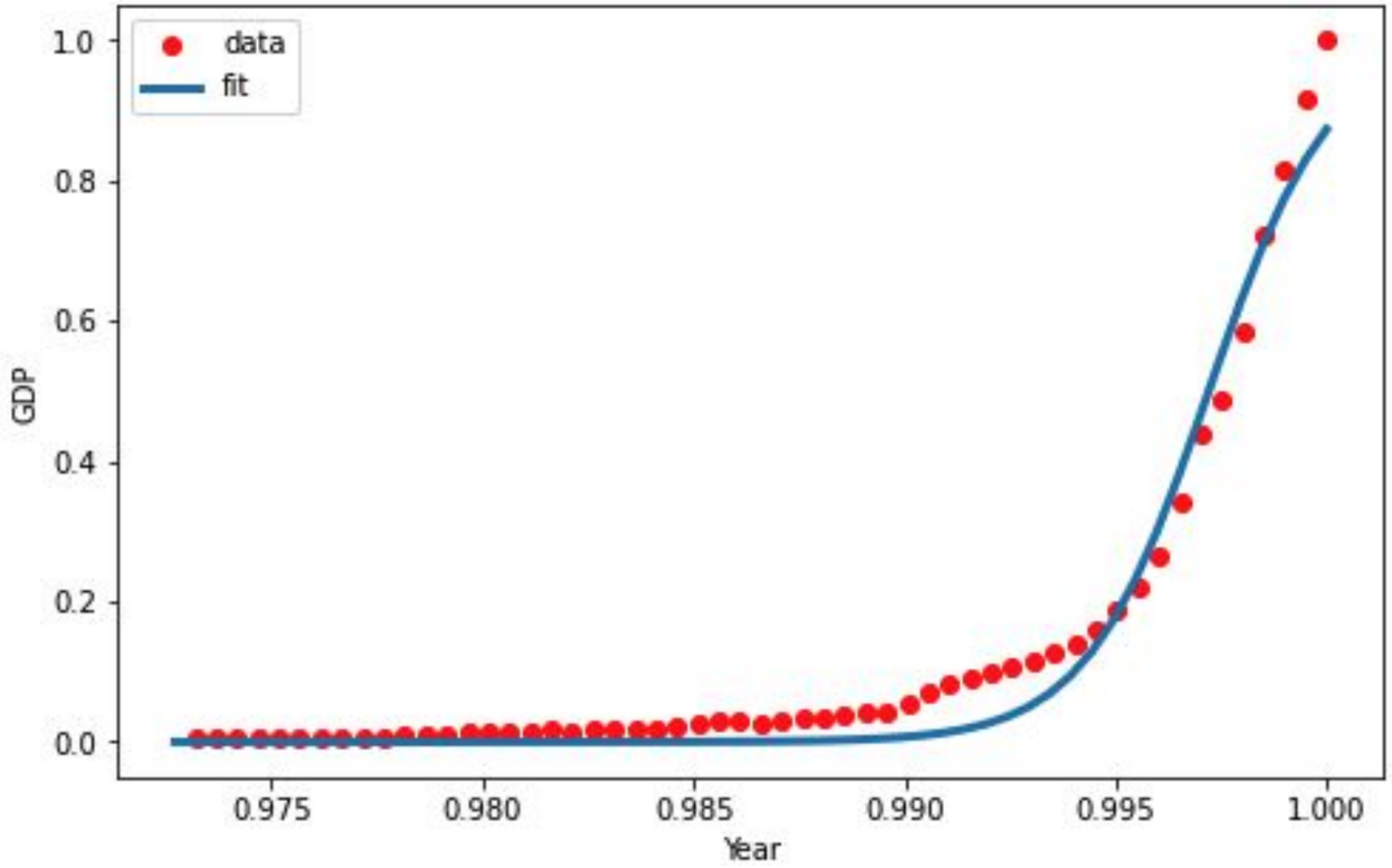
- Try to have theoretical defense when choosing the independent variables. too many Xs might result in over fitting
- Xs do not need to be continues. If they are not try to assign values (like 1 and 2) to categories
- there needs to be a linear relationship. Test your Xs with scatter plots or use your logic. If the relationship displayed in your scatter plot is not linear, then you need to use non-linear regression.

Lab: Multiple Linear Regression

- ML0101EN-Reg-Mulitple-Linear-Regression-Co2.ipynb

Non-Linear Regression

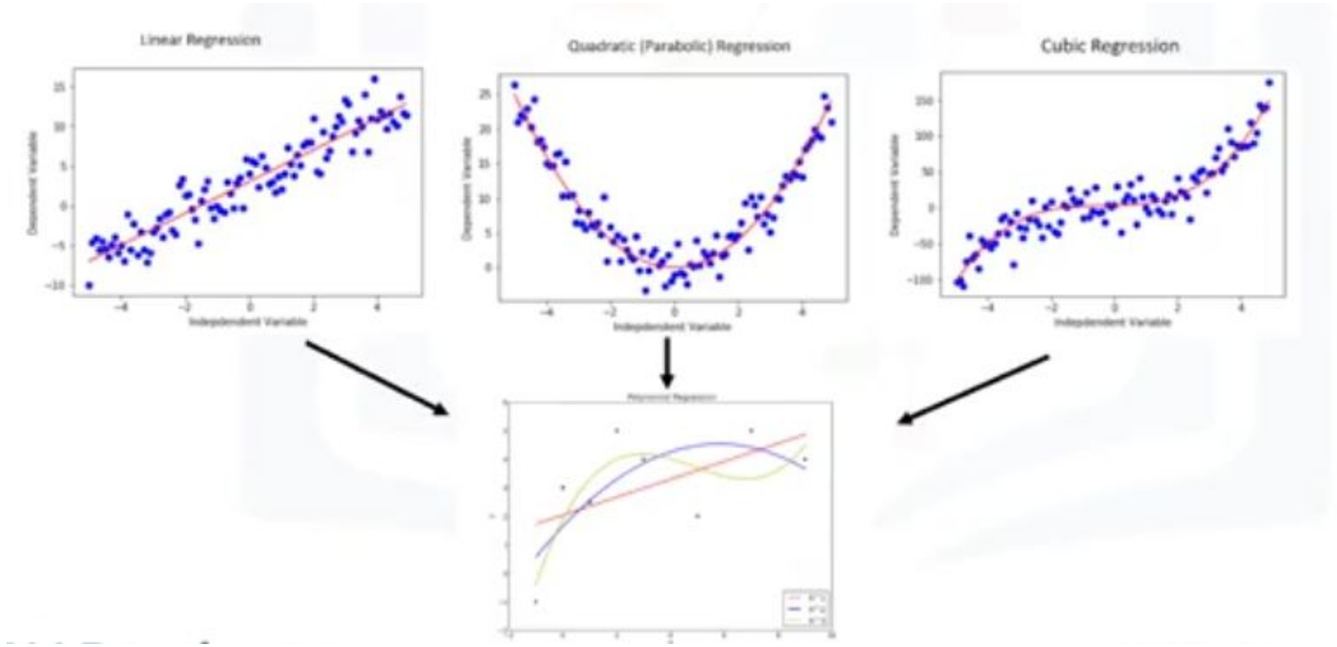
	Year	Value
0	1960	5.918412e+10
1	1961	4.955705e+10
2	1962	4.668518e+10
3	1963	5.009730e+10
4	1964	5.906225e+10
5	1965	6.970915e+10
6	1966	7.587943e+10
7	1967	7.205703e+10
8	1968	6.999350e+10
9	1969	7.871882e+10



Non Linear regression

Polynomial

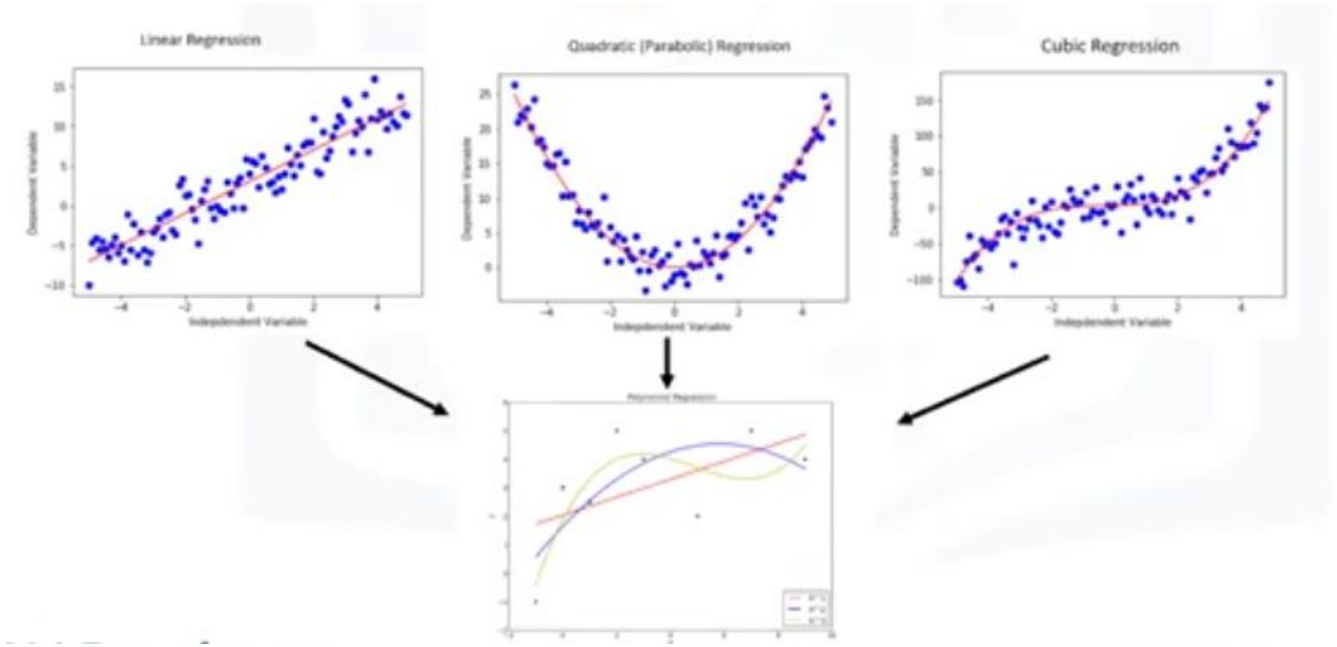
- Deferent types
- if you have X^2, it is possible to define a new X as X^2. So it can be represented as a special case of multiple linear regression. This is called polynomial



Non Linear regression

Non Linear

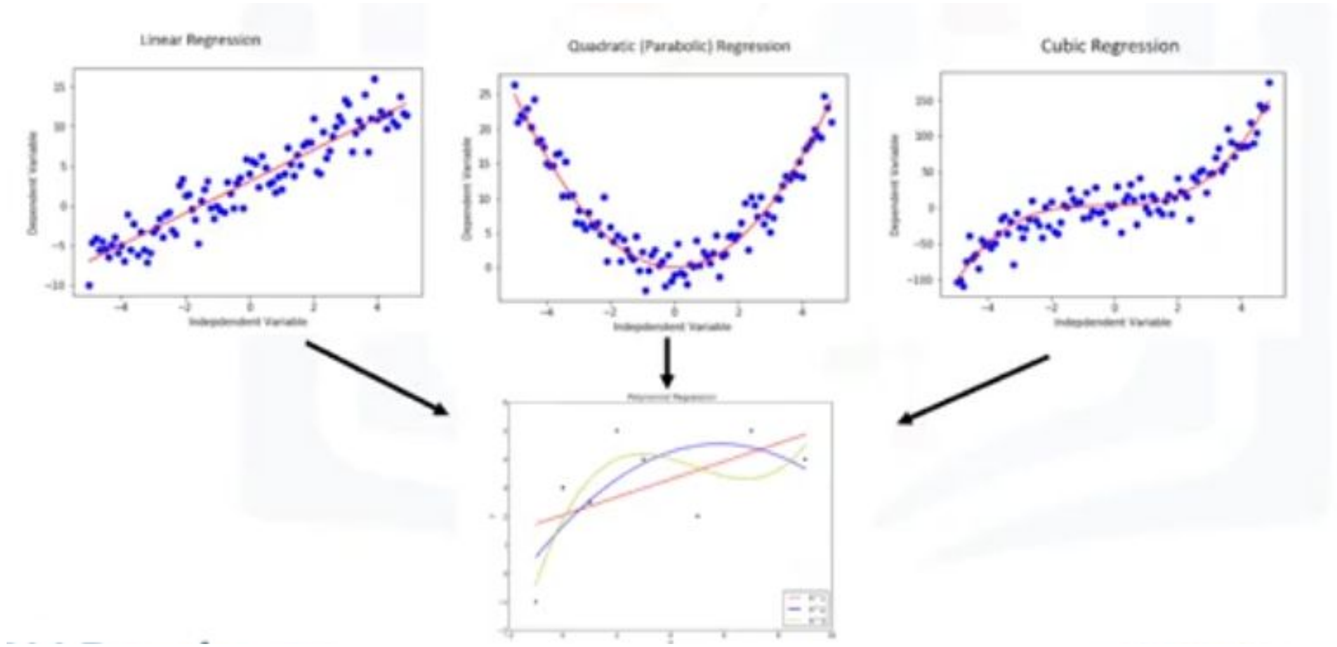
- Models a non linear relationship between Xs and Y
- Y is non linear function of parameters Theta



Non Linear regression

Notes

- How to say if nonlinear:
  - plot and see. Y based on each X / calculate coefficient
  - use non-linear if you can not solve by linear
- How to model data of its non-linear
  - polynomial
  - non-linear
  - transform data!



Lab: Polynomial Regression

- ML0101EN-Reg-Polynomial-Regression-Co2.ipynb

Lab: Non Linear Linear Regression

- ML0101EN-Reg-NoneLinearRegression.ipynb

بخش سوم
Classification

Classification

Intro

- Understand Classification
- Understand different methods such as KNN, Decision Trees, Logistic Regression and SVM
- Apply on datasets
- Evaluate

Classification

Intro

- Supervised
- Categorizing unknown items in classes
- Target is categorical with discrete values (called classifier)
- Binary: 2 values vs Multi Class

Classification

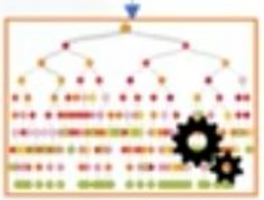
Intro

Age	Sex	BP	Cholesterol	Na	K	Drug
23	F	HIGH	HIGH	0.793	0.031	drugY
47	M	LOW	HIGH	0.739	0.056	drugC
47	M	LOW	HIGH	0.697	0.069	drugC
28	F	NORMAL	HIGH	0.564	0.072	drugX
61	F	LOW	HIGH	0.559	0.031	drugY
22	F	NORMAL	HIGH	0.677	0.079	drugX
49	F	NORMAL	HIGH	0.79	0.049	drugY
41	M	LOW	HIGH	0.767	0.069	drugC
60	M	NORMAL	HIGH	0.777	0.051	drugY
43	M	LOW	NORMAL	0.526	0.027	drugY

Age	Sex	BP	Cholesterol	Na	K	Drug
36	F	LOW	HIGH	0.697	0.069	

Categorical Variable

Modeling



Classifier

Classification

Intro

- Loan (age, income, load size, previous records, ...)
- Churn (age, address, income, equip, data usage, calls, ...)
- Spam / Important email
- Handwriting/Speech recognition
- Biometric identification

Classification

Intro

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

Classification

KNN

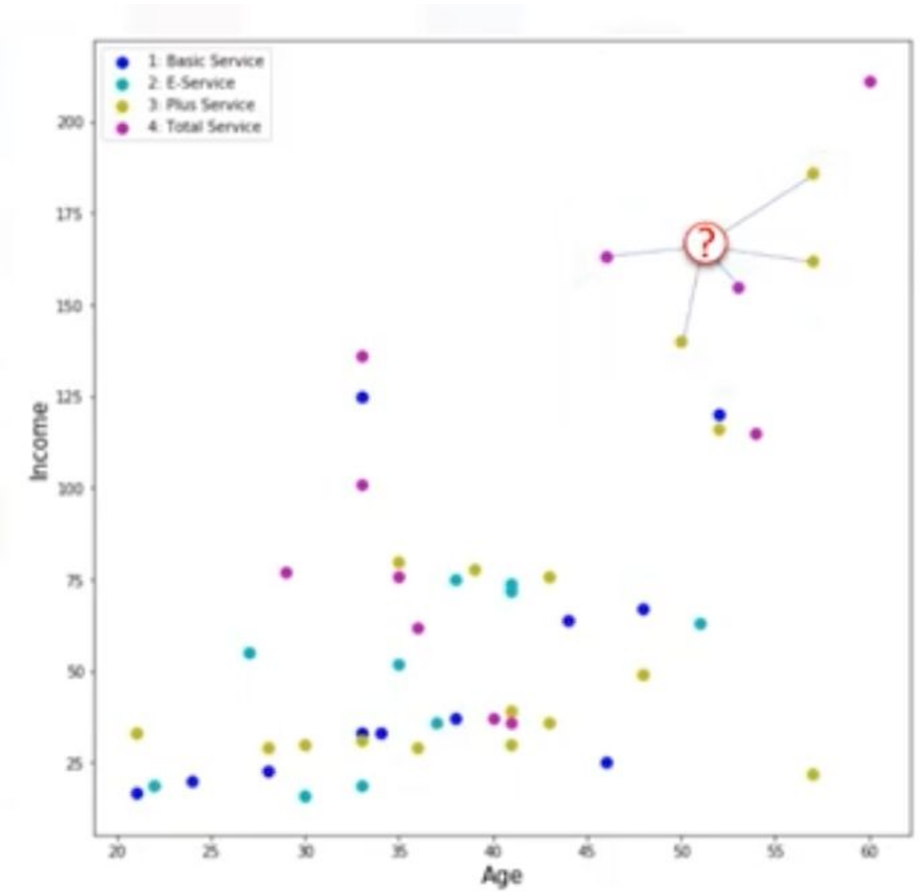
-

	region	tenure	age	marital	address	income	ed	employ	retire	gender	reside	custcat
0	2	13	44	1	9	64.0	4	5	0.0	0	2	1
1	3	11	33	1	7	136.0	5	5	0.0	0	6	4
2	3	68	52	1	24	116.0	1	29	0.0	1	2	3
3	2	33	33	0	12	33.0	2	0	0.0	1	1	1
4	2	23	30	1	9	30.0	1	2	0.0	0	4	3
5	2	41	39	0	17	78.0	2	16	0.0	1	1	3
6	3	45	22	1	2	19.0	2	4	0.0	1	5	2
7	2	38	35	0	5	76.0	2	10	0.0	0	3	4
8	3	45	59	1	7	166.0	4	31	0.0	0	5	3

Classification

KNN

- pick K
- calculate of the unknown points distance from all cases
- predict based on the K nearest points

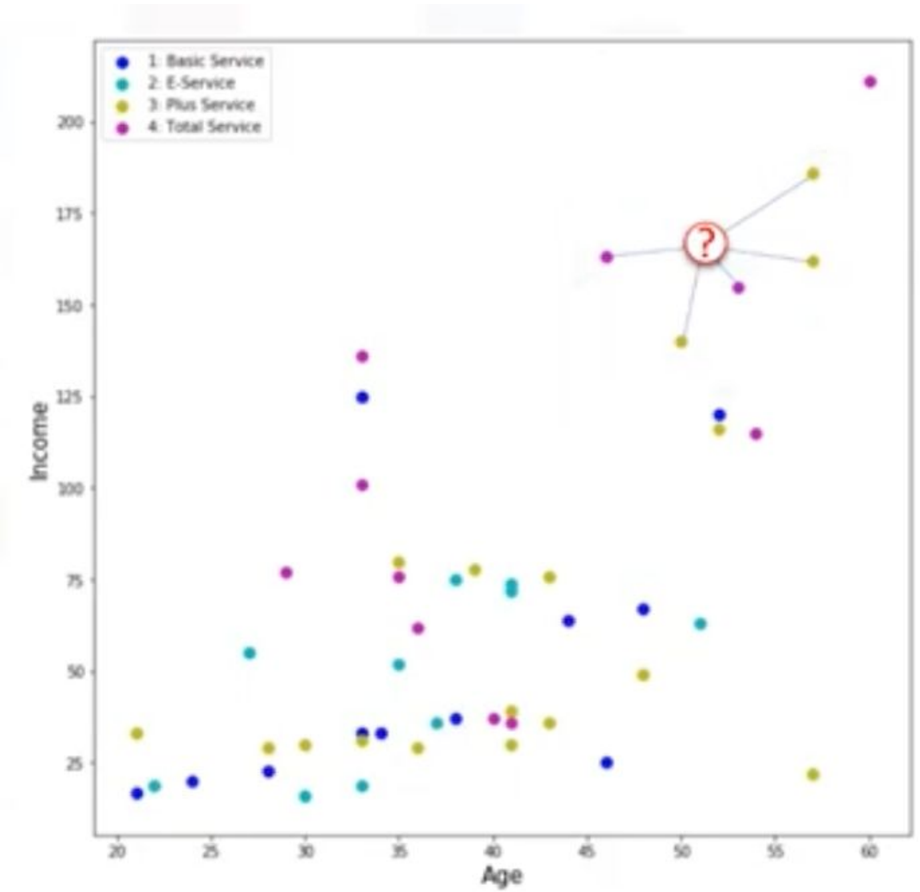


- How to find "distance" (Euclidean can be one way)
- How to choose K (low -> noise & overfit; high -> too general). Use the different Ks with test set and see which K is good.

Classification

KNN

- KNN can be used to compute a continuous target (regression)
- Say find 3 of the closest cases and find the median



Classification

KNN Evaluation

- Evaluation explains the performance of our model
- On test data we have y and $\hat{y}$
- There are different model evaluation metrics: Jaccard index, F1-score, and Log Loss.

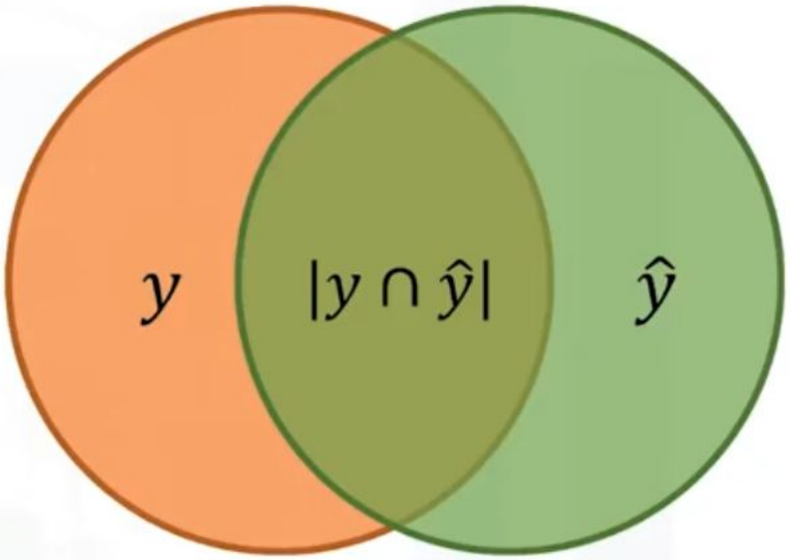
Classification

KNN Evaluation / Jaccard Index

y: Actual labels

$\hat{y}$: Predicted labels

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



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

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

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

Classification

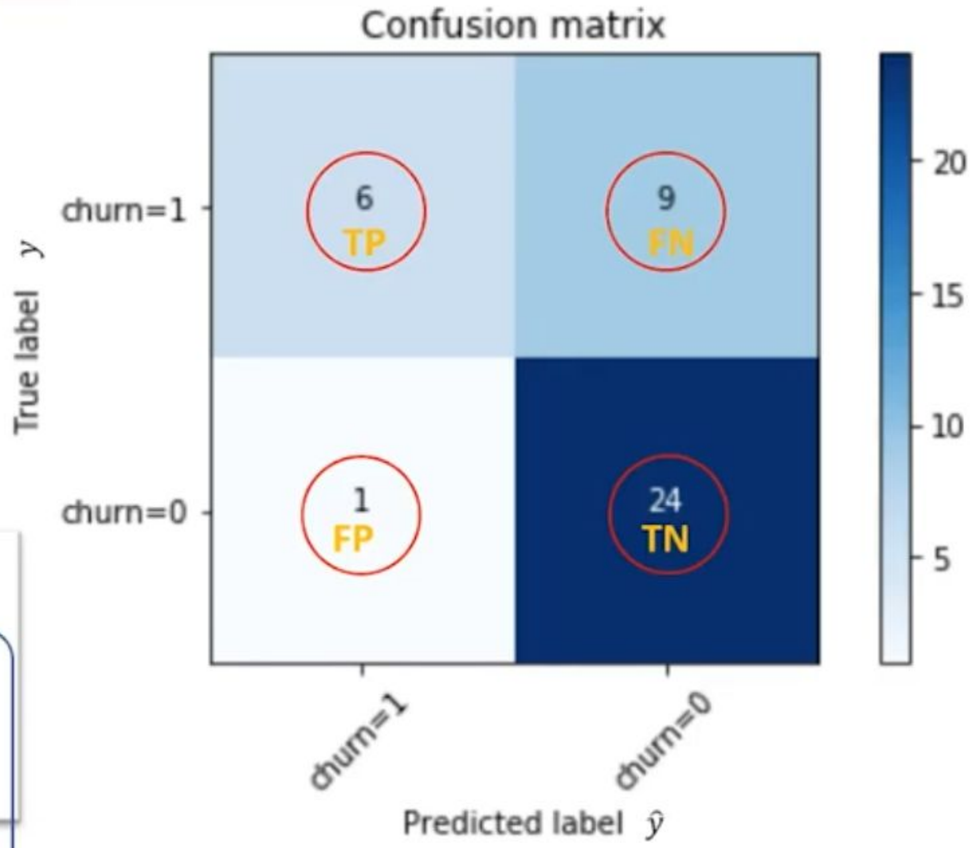
KNN Evaluation / F1-Score

- 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

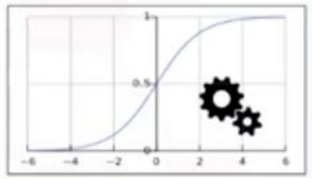
	precision	recall	f1-score
Churn = 0	0.73	0.96	0.83
Churn = 1	0.86	0.40	0.55
		Avg Accuracy =	0.72



Classification

KNN Evaluation / LogLoss

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

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

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

Higher Accuracy

Lab: KNN

-

Classification

Decision Trees / Intro

-

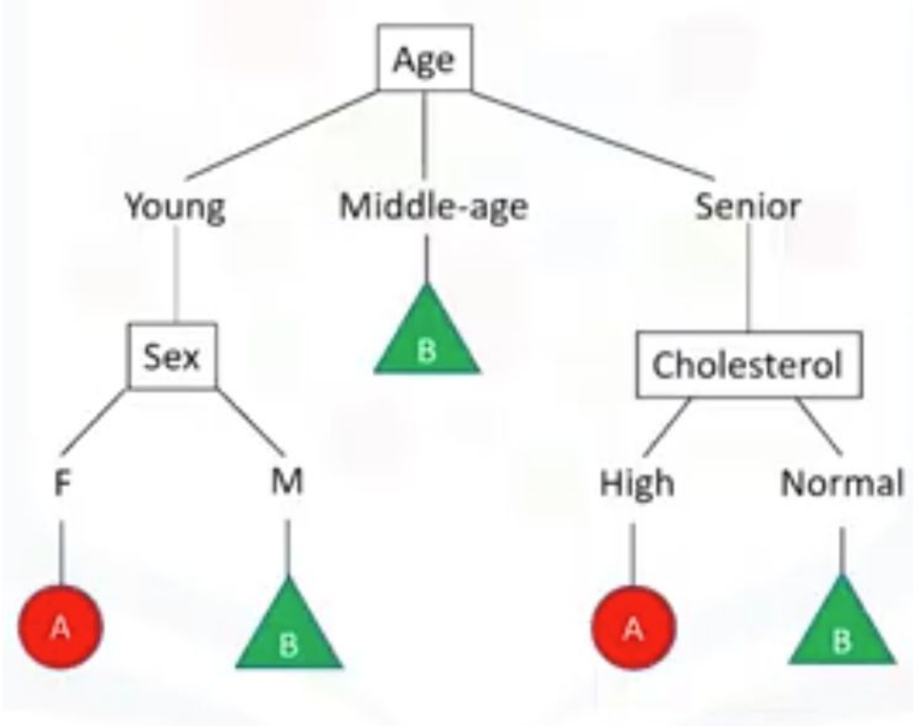
Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	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
p15	Middle-age	F	Low	Normal	?

Classification

Decision Trees / Intro

Internal Node (test), branch (result of test) & leaf (class)

1. Choose attribute from dataset
2. Calculate the significance of the attribute in the splitting of data
3. split data based on value of the best attribute
4. replete !

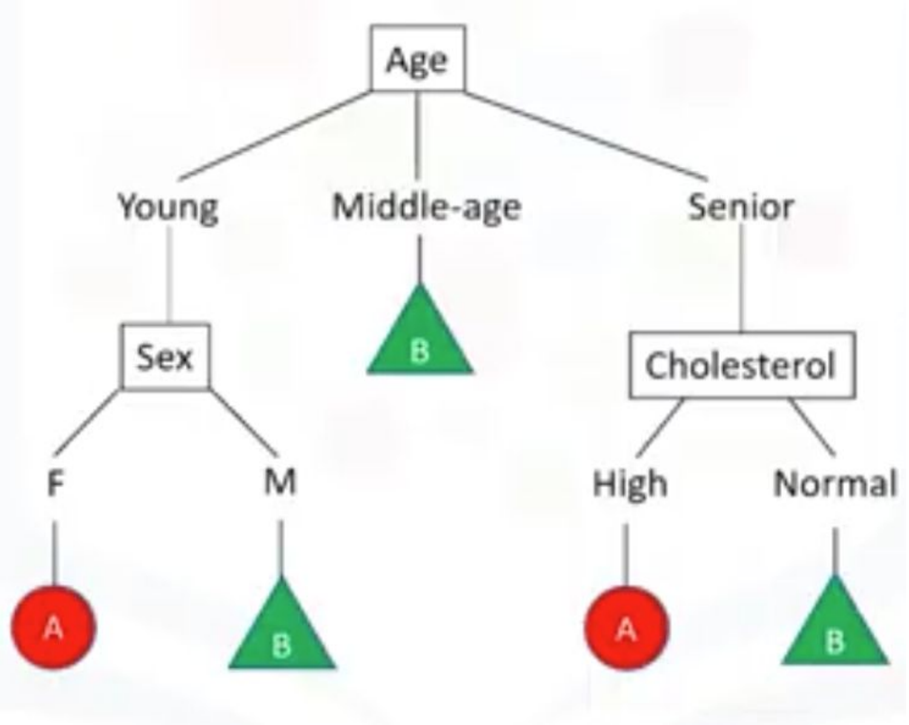


Classification

Decision Trees / Building

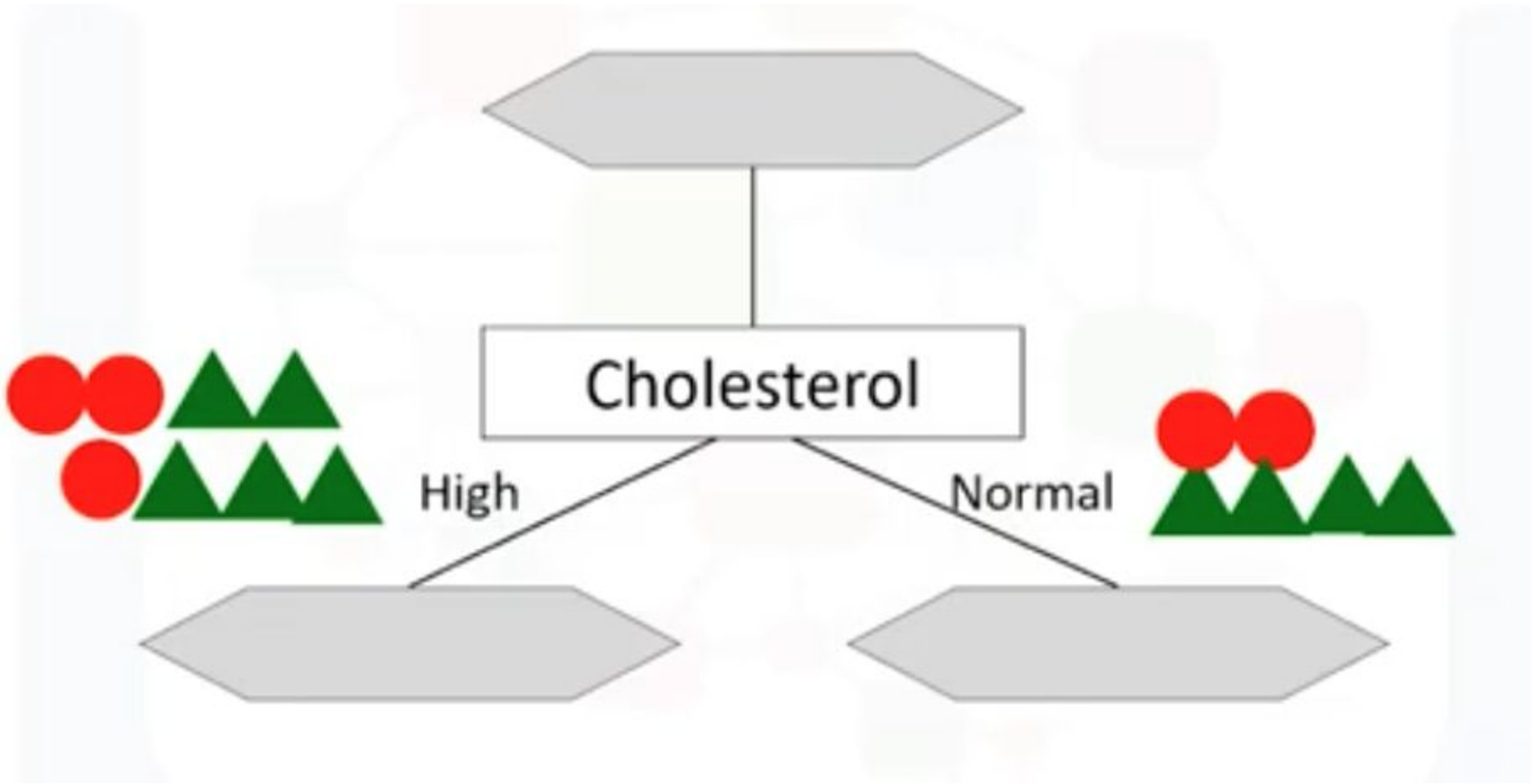
Decision trees are built using recursive partitioning to classify the data.
Cholesterol? Sex? ...

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
p15	Middle-age	F	Low	Normal	?



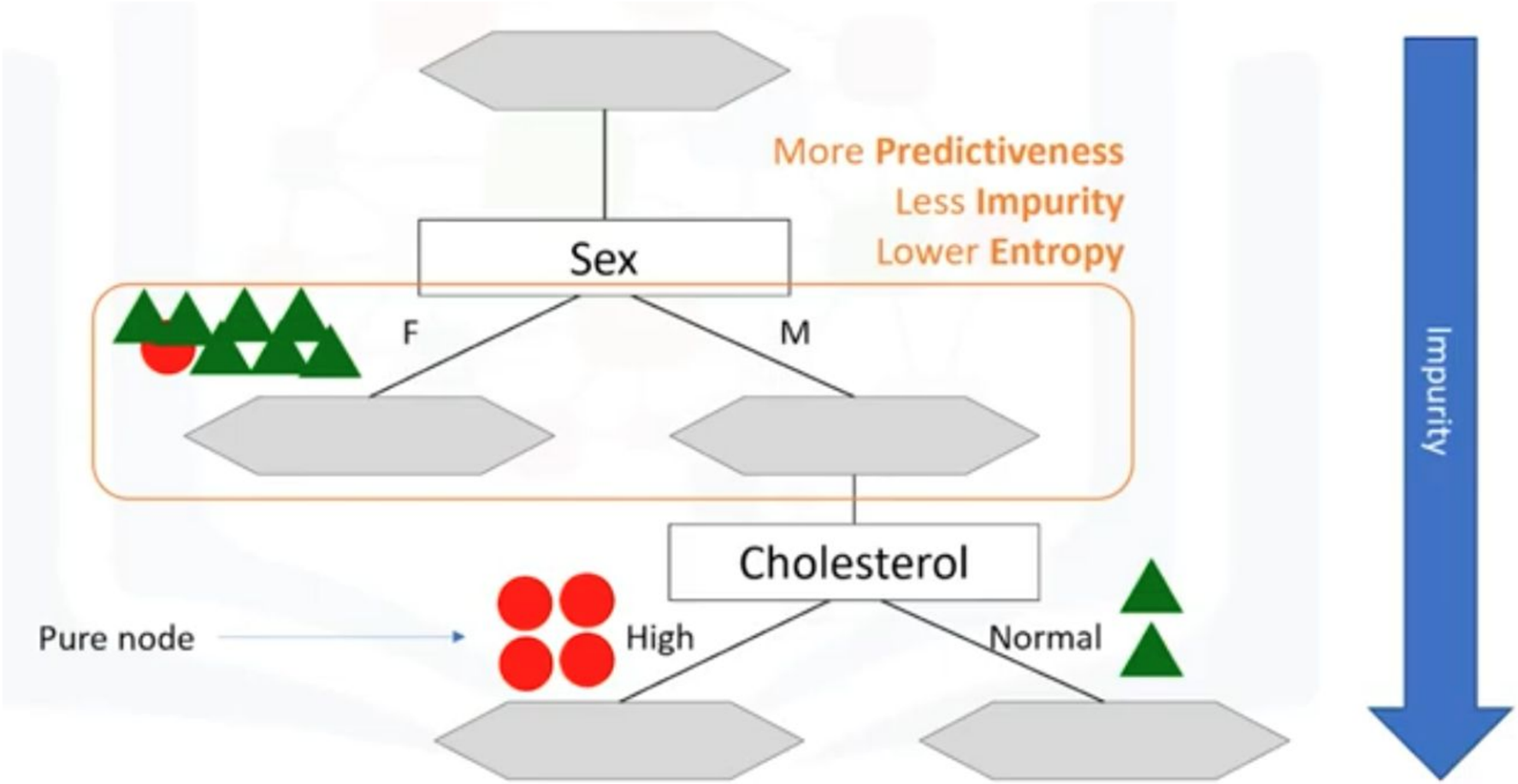
Classification

Decision Trees / Building



Classification

Decision Trees / Building

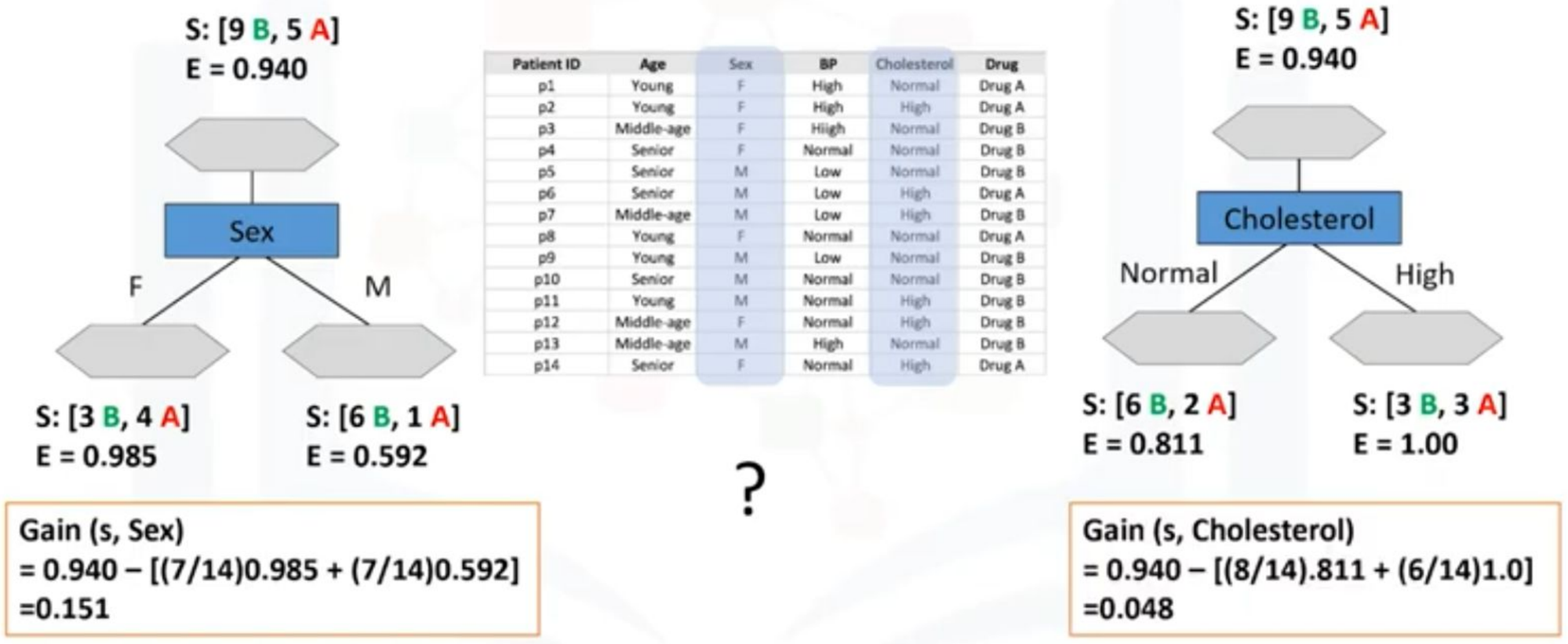


Classification

Decision Trees / Building

- The best? the one with the most information gain
- Information gain is the information that can increase the level of certainty after splitting.

- IG = Entropy before split - Weighted entropy after split.



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

Lab: Decision Trees

-

Classification

Logistic Regression/ Intro

- who is leaving and why
  - Close to Regression but here, Y is a categorical (here binary) value
  - All Xs should be continues, or converted to “continues”

	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

Classification

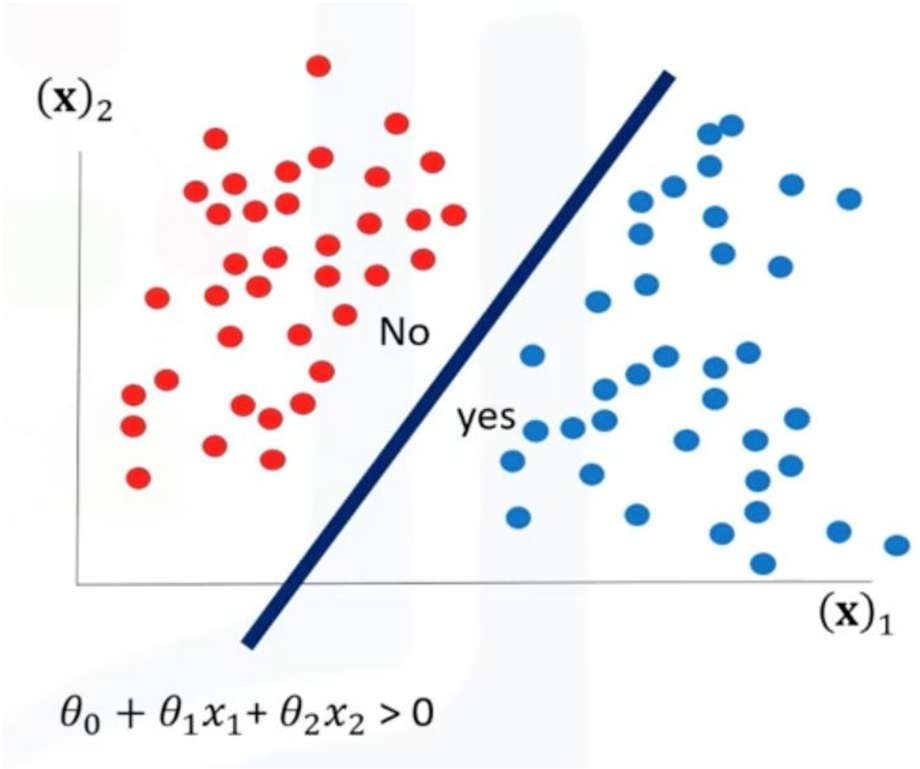
Logistic Regression/ Intro

- Predicting a disease
  - chance of mortality based on a situation
  - halting a subscription
  - purchase
  - failure of a product
  - ...

Classification

Logistic Regression/ Intro

- Target should be category (or better, binary)
  - We need the probability of prediction
  - we need a linear decision boundary (line or even polynomial)
  - We need to understand the impact of features (Theta is closer to 0 or is high)



Classification

Logistic Regression vs Linear 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	1.0
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	1.0
2	23.0	30.0	9.0	30.0	1.0	2.0	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	0.0

$X \in \mathbb{R}^{m \times n}$

$y \in \{0,1\}$

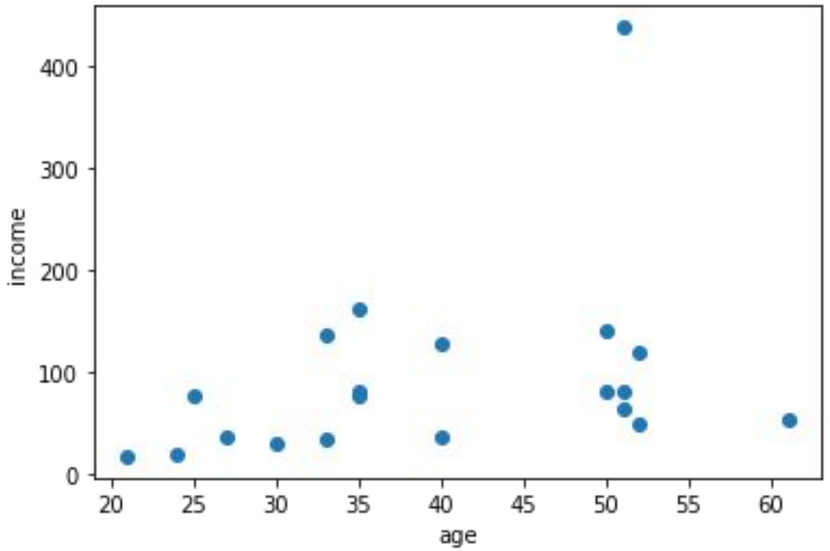
$\hat{y} = \text{P(y=1|x)}$

P(y=0|x) = 1- P(y=1|x)

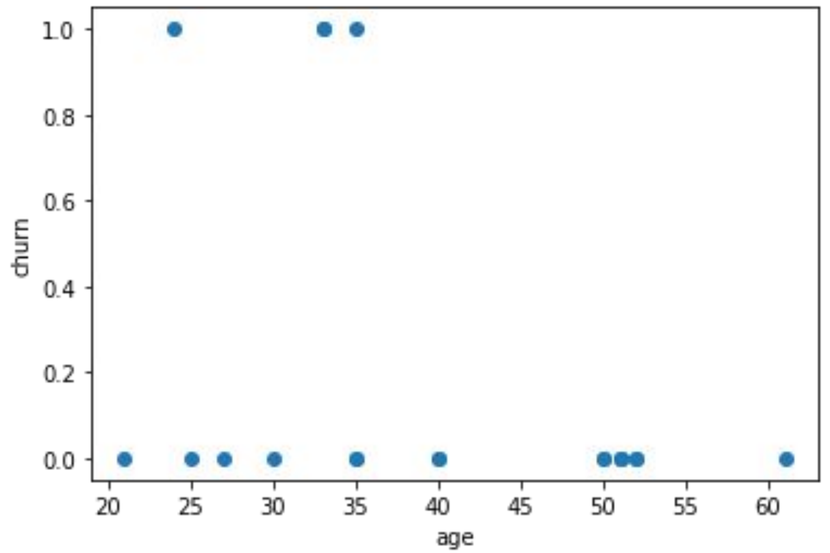
Classification

Logistic Regression vs Linear Regression

- On Previous data, try linear with age vs income



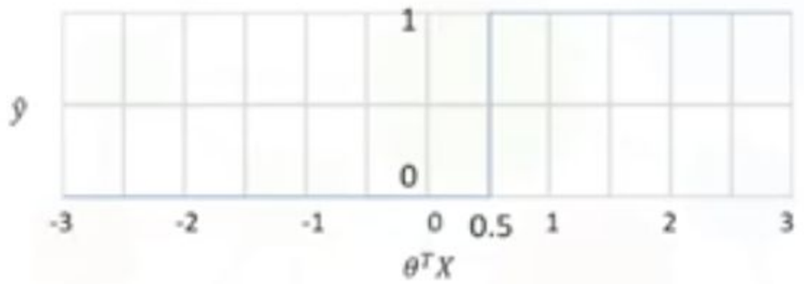
- now repeat, trying with age vs churn: funny and we should have a step function as threshold



Classification

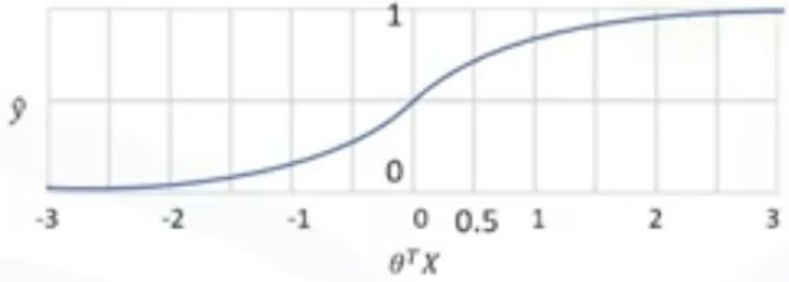
Logistic Regression vs Linear Regression / Sigmoid

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



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

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



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

P(y=1|x)

Classification

Logistic Regression Training

$\sigma(\theta^T X)$ → 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}

Classification

Logistic Regression Training

- Cost Function
- we have to minimize the Cost
- Can be done via derivative but its difficult

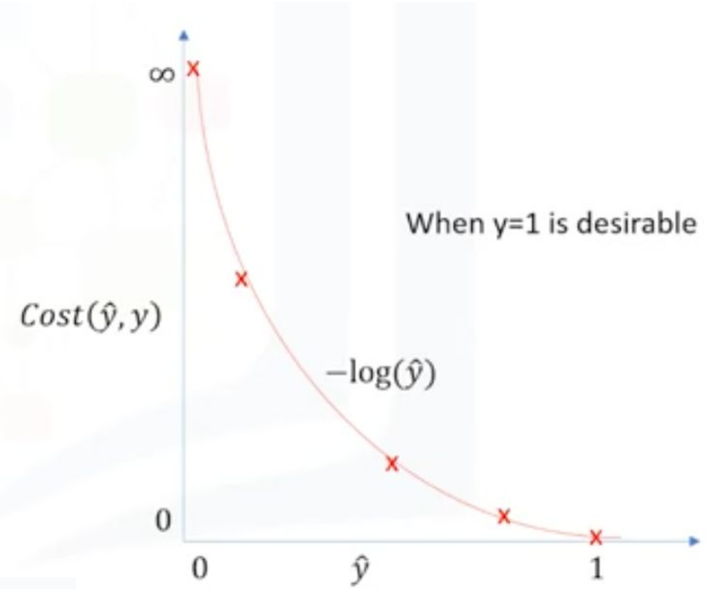
$$Cost(\hat{y}, y) = \frac{1}{2}(\sigma(\theta^T X) - y)^2$$

$$J(\theta) = \frac{1}{m}\sum_{i-1}^{m} Cost(\hat{y}, y)$$

Classification

Logistic Regression Training

- We can define a new Cost function!
- here there are more approaches to minimize the function; say Gradient Descent (iterative technique)



$$Cost(\hat{y}, y) = \frac{1}{2}\left(\sigma(\theta^T X) - y\right)^2$$

$$Cost(\hat{y}, y) = \begin{cases} -\log(\hat{y}) & if\ y = 1 \\ -\log(1 - \hat{y}) & if\ y = 0 \end{cases}$$

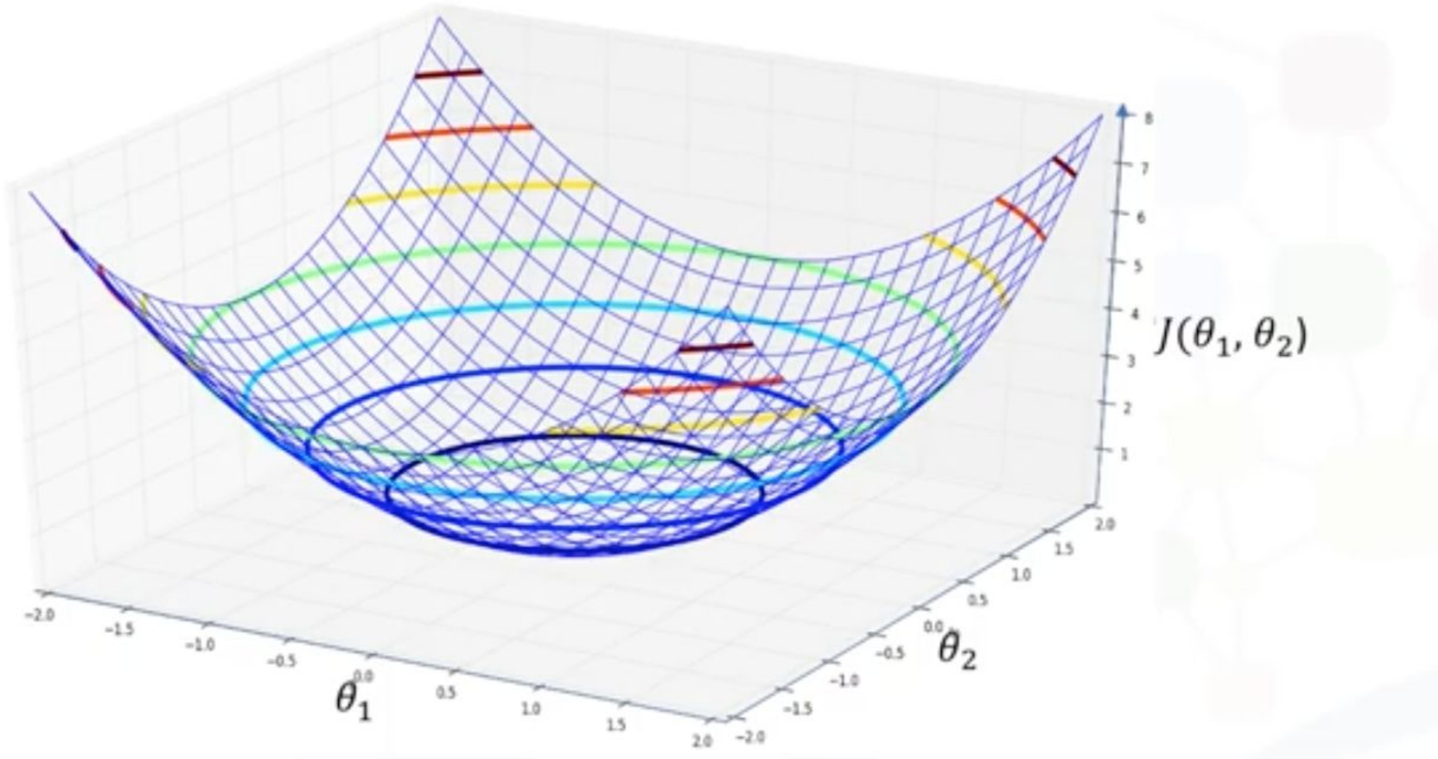
$$J(\theta) = \frac{1}{m}\sum_{i-1}^{m} Cost(\hat{y}, y)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

Classification

Logistic Regression Training

- Gradient descent is an iterative approach to finding the minimum of a function. It uses the derivative of a cost function to change the parameter values to minimize the cost or error.



$$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

Classification

Logistic Regression Training

1. initialize the parameters randomly.
2. Feed the cost function with training set, and calculate the error.
3. Calculate the gradient of cost function.
4. Update weights with new values.
5. Go to step 2 until cost is small enough.
6. Predict the new customer X.

$$\theta^T = [\theta_0, \theta_1, \theta_2, ...]$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

$$\nabla J = [\frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2}, \frac{\partial J}{\partial \theta_3}, \cdots, \frac{\partial J}{\partial \theta_k}]$$

$$\theta_{new} = \theta_{prv} - \eta \nabla J$$

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

Lab: Logistic Regression

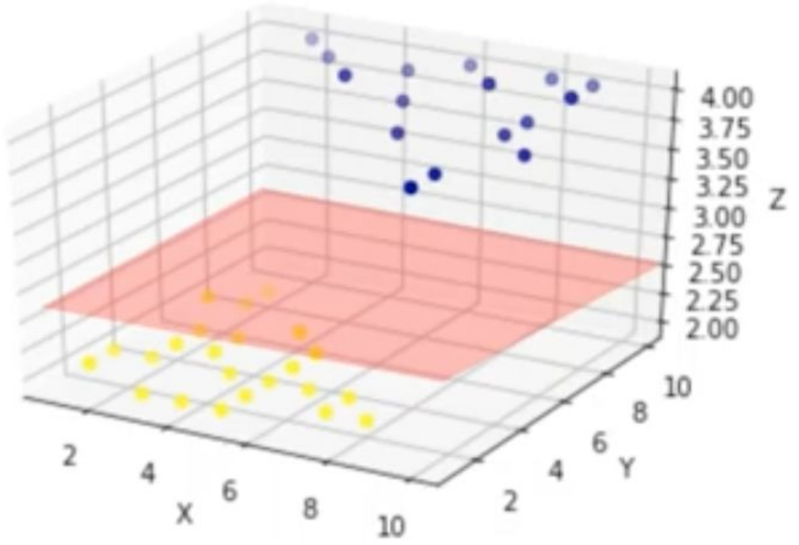
-

Classification

Support Vector Machines

- supervised
- classifier based on separator
- mapping data to high-dimensional so a hyperplane separator can be drawn
- Lots of real world datas are Linearly non separable , but what if we go to a higher dimension? ;)

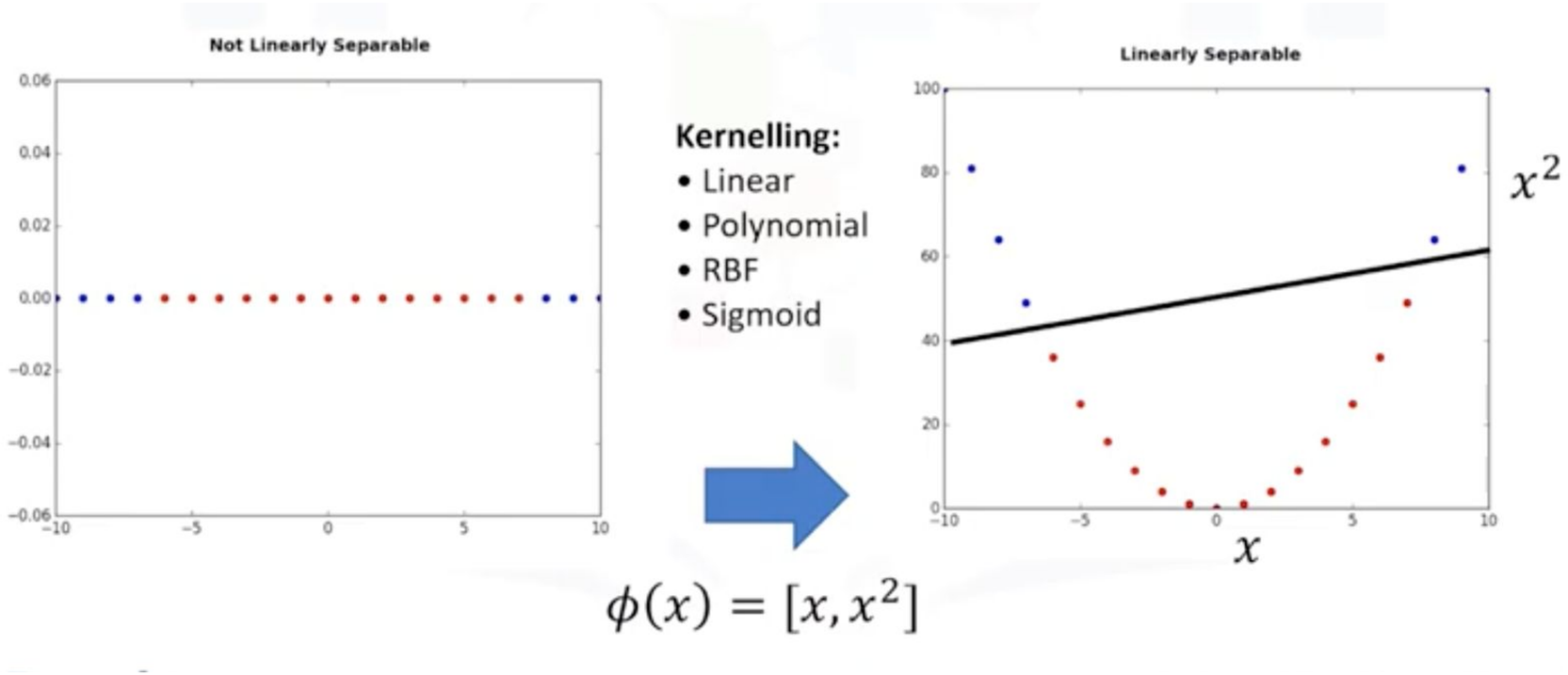
Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
5	1	1	1	2	1	3	1	1	benign
5	4	4	5	7	10	3	2	1	benign
3	1	1	1	2	2	3	1	1	malignant
6	8	8	1	3	4	3	7	1	benign
4	1	1	3	2	1	3	1	1	benign
8	10	10	8	7	10		7	1	malignant
1	1	1	1	2	10	3	1	1	benign
2	1	2	H	2	1	3	1	1	benign
2	1	1	1	2	1	1	1	5	benign
4	2	1	1	2	1	2	1	1	benign



Classification

Support Vector Machines

- but… how to move to n-dimention?
- there are different kernel functions
- our libraries will do, we will just compare
- How to find the hyperplane?

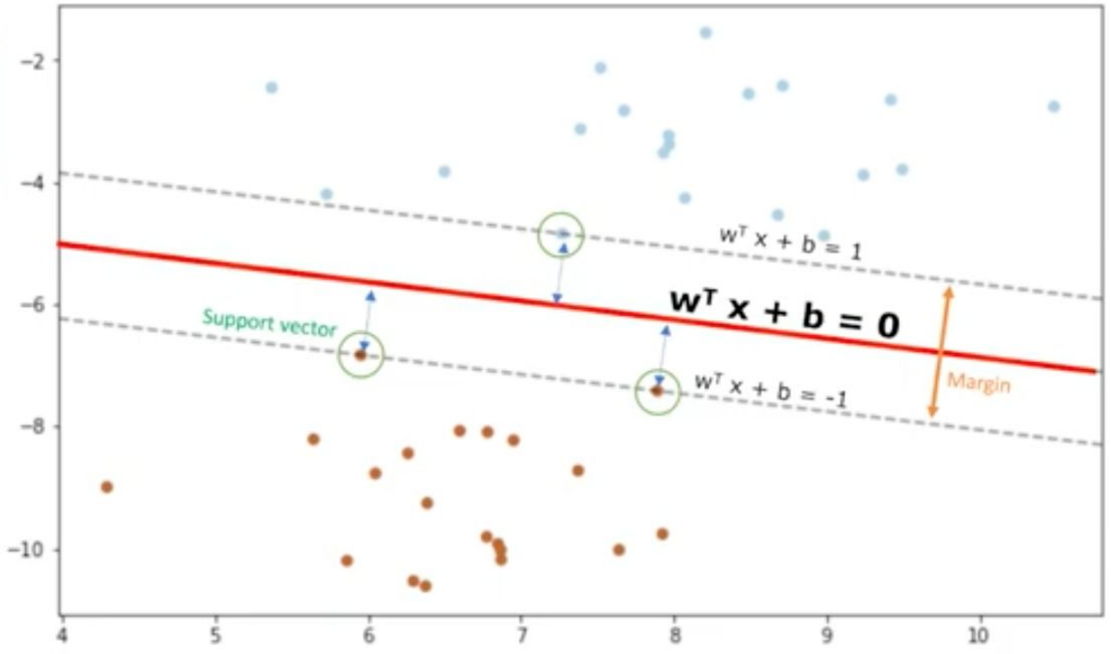


Classification

Support Vector Machines

- to find the hyperplane, we are looking for largest margins from support vectors
- can also be solved using gradient descent
- when learned, we can just check the data and see if its above the line or below it and decide

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



Classification

Support Vector Machines

- Pros
  - accurate in high dimensional spaces
  - memory efficient
- Cons
  - Prone to over-fitting if we have lots of features
  - No probability estimation
  - Not computationally efficient for large dataset (n>1000)

Classification

Support Vector Machines

- Image recognition
- Text Category Assignment
  - spam
  - category
  - sentiment analysis
- Gene Expression Classification
- Outlier detection and clustering

Lab: SVM

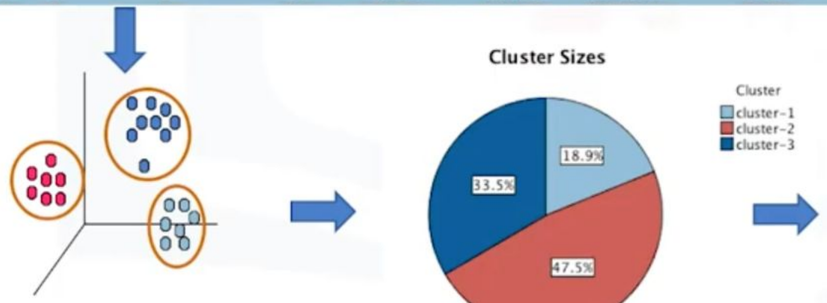
-

Clustering

Intro

- Partitioning a customer base into groups of individuals based on characteristics
- Allows a business to target different groups (high profit&low risk, …)
- we can cross-reference the groups with their purchases

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



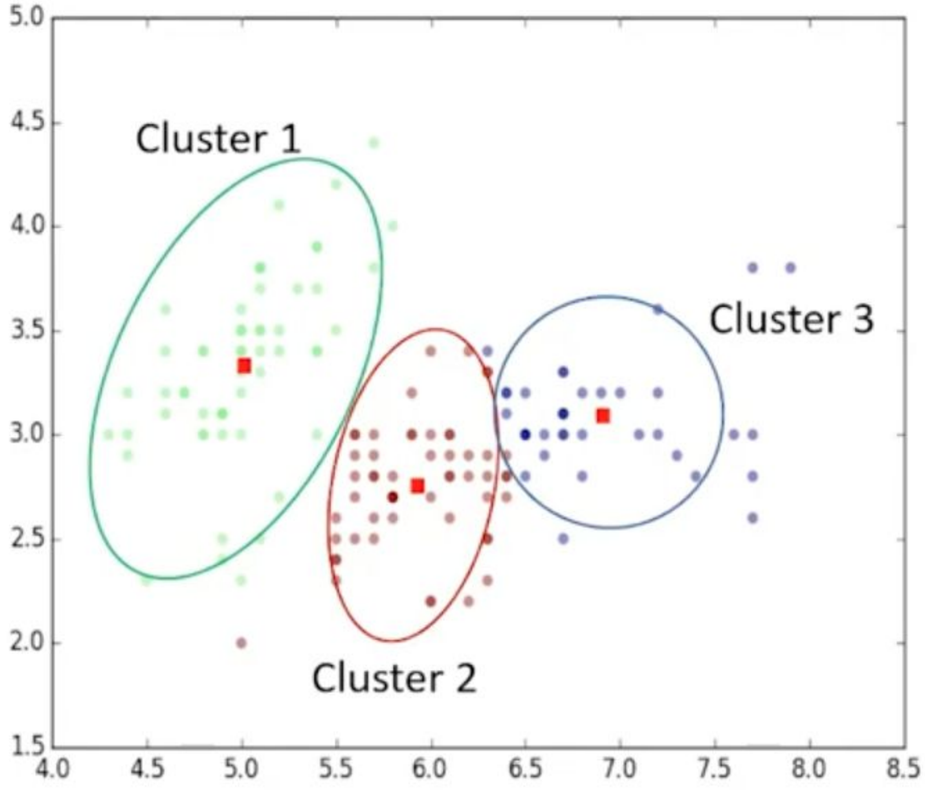
Customer ID	Segment
1	YOUNG AND LOW INCOME
2	AFFULUENT AND MIDDLE AGED
3	AFFULUENT AND MIDDLE AGED
4	YOUNG AND LOW INCOME
5	AFFULUENT AND MIDDLE AGED
6	AFFULUENT AND MIDDLE AGED
7	YOUNG AND LOW INCOME
8	YOUNG AND LOW INCOME
9	AFFULUENT AND MIDDLE AGED

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

Clustering

Intro

- finding “clusters” in datasets, unsupervised
- Cluster: a group of data points or objects in a dataset that are similar to other objects in the group, and dissimilar to datapoints in other clusters.
- Different than classficiation:
  - no need to be labeled
  - Prediction is not the goal



Clustering

Intro / Samples

- Retail & Marketing: identify buying patterns / recommendation systems
- Banking: Fraud detection / identify clusters (loyal, churn, …)
- Insurance: Fraud detection / Risk
- Publication: auto-categorize / recommend
- Medicine: characterize behaviour
- Biology: group genes / cluster genetic markers (family ties)

Clustering

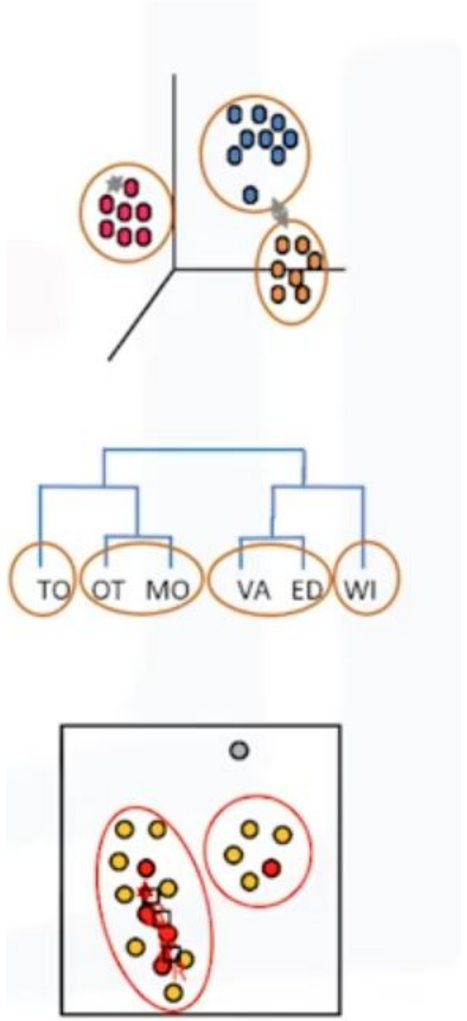
Intro / Where

- Exploratory data analysis
- summary generation
- outlier detection
- finding duplicates
- pre-processing step

Clustering

Intro / algorithms

- Partitioned-based (K-means, K-Median, Fuzzy c-means, …): sphere like clusters / Medium or large data
- Hierarchical (Agglomerative, Divisive): Trees of clusters / small size datasets
- Density-based (DBSCAN): arbitrary shaped / good for special clusters or noisy data



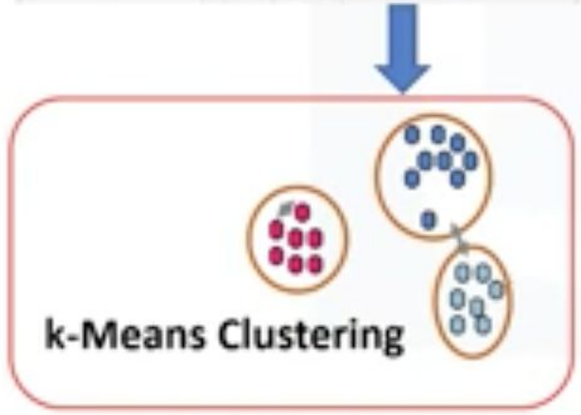
Clustering

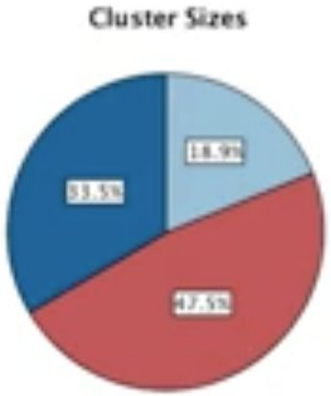
K Means

- Unsupervised, Divides data into K non-overlapping subset/cluster without any cluster internal structure

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





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

Clustering

K Means

- We need to understand the similarity and dissimilarity.
- Golad: minimize intra-cluster distances (Dis(x1, x2)) and maximize inter-cluster distances (Dis(c1, c2))
- It is always good to Normalize!
- different formulas: Euclidean, Cosine, Average distance, … so first understand the domain knowledge

Customer 1		
Age	Income	education
54	190	3

Customer 2		
Age	Income	education
50	200	8

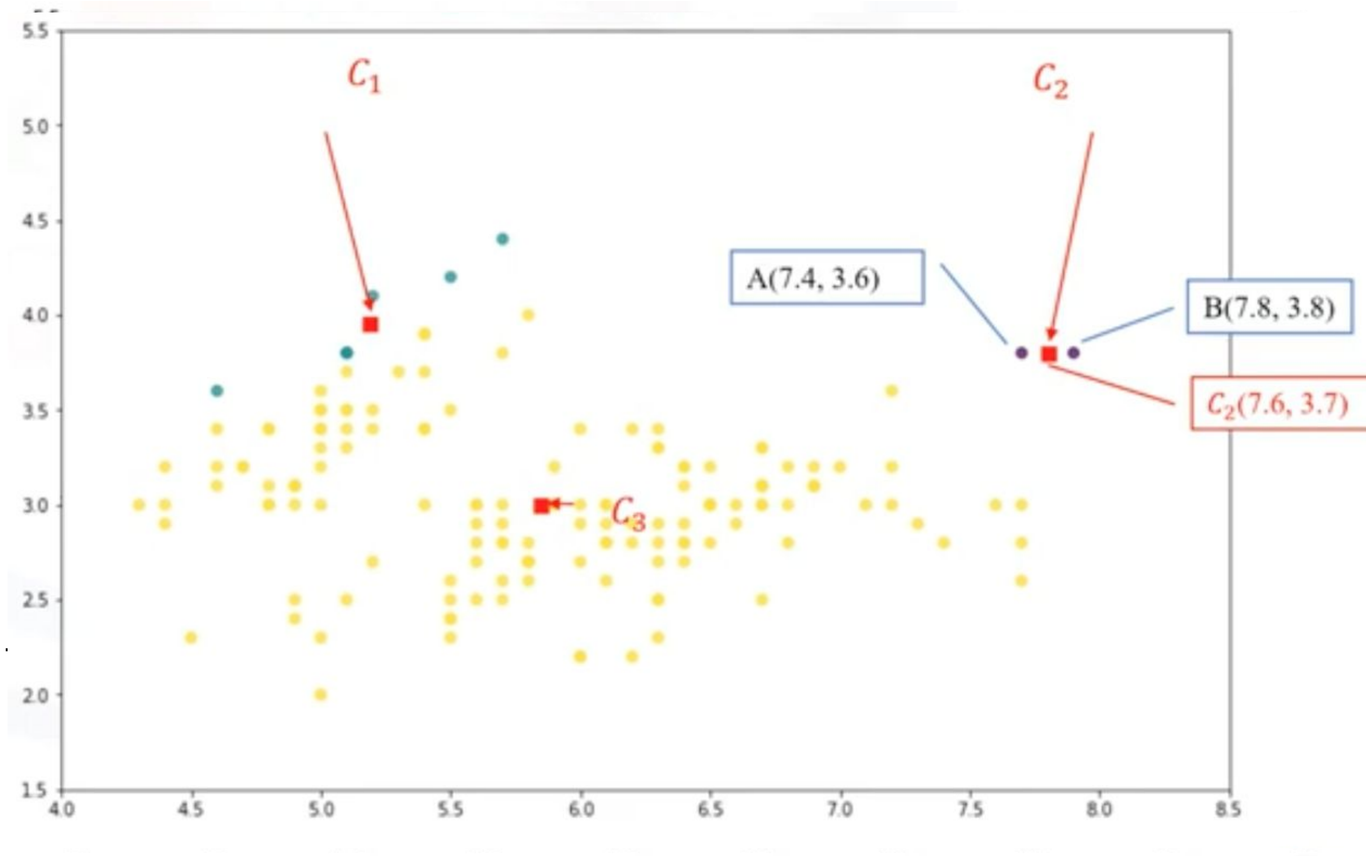
$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

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

Clustering

K Means

- decide the number of cluster (K)
- init K “centroids” by:
  - random points from the dataset
  - random points
- assign each customer to the closest centroid and create distance matrix
- $SSE = \sum_{1}^{n} (x_i - C_j)^2$
- update the centroid to the mean of its datapoints
- continue till the centroids stop moving
- Notes:
  - iterative
  - does not guarantee the best result. may catch a local optimum; but its fast so we can run it many times!



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

Clustering

K Means / More Points

- Review the algorithm
- but how can we evaluate?
  - External: compare with truth
  - Internal: Average distance between datapoints within a cluster or the distance between clusters
  - Choosing K is difficult so we run with different Ks and check the accuracy (say mean mean distance inside a cluster) BUT decreasing K will always reduces this. So we do the elbow method

Clustering

K Means / More Points

- Partition based
- unsupervised
- medium and large datasets (relatively efficient)
- sphere like clusters
- K should be known / guessed

Clustering

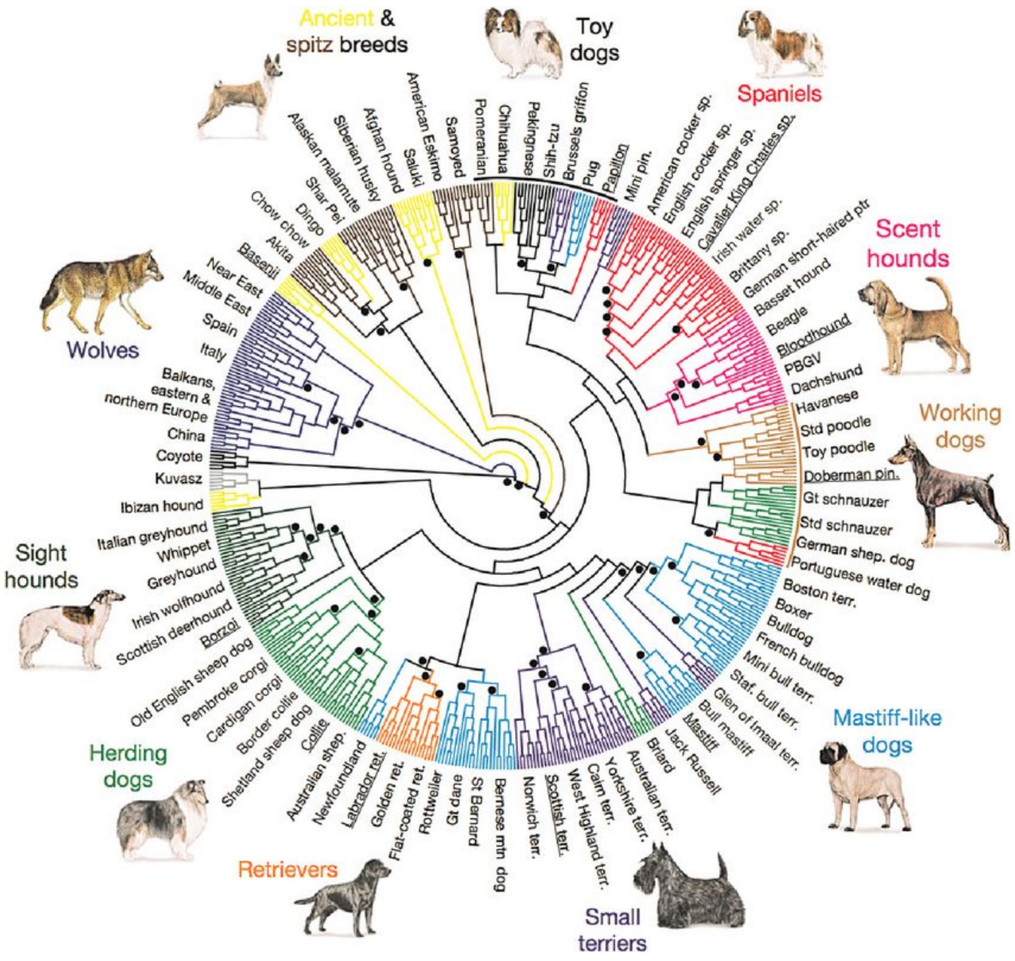
K Means / LAB

-

Clustering

Hierarchical / Intro

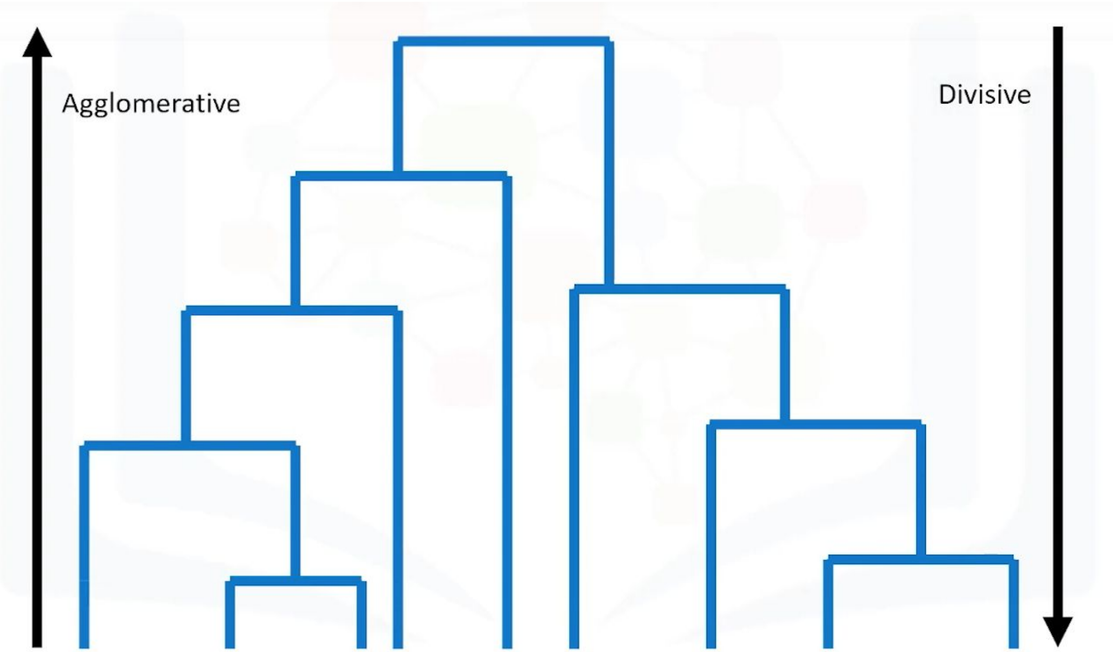
- 48000 genetic markers makes this chart from similarity
- Hierarchy of clusters where each node is a cluster consisting of the clusters of its daughter nodes



Clustering

Hierarchical / Intro

- Divisive is top down, so you start with all observations in a large cluster and break it down into smaller pieces.
- Agglomerative is the opposite of divisive. So it is bottom up, where each observation starts in its own cluster and pairs of clusters are merged together as they move up the hierarchy.



Clustering

Hierarchical / Intro

- Finding Similarity of city locations in Canada
- Dendrogram Y is the similarity
- We can cut Y somewhere to have N number of clusters (say 3)

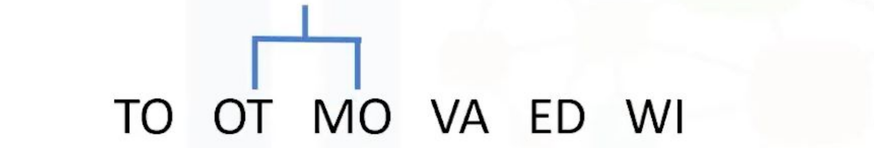
TO OT MO VA ED WI

	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						

Clustering

Hierarchical / Intro

- Finding Similarity of city locations in Canada
- Dendrogram Y is the similarity
- We can cut Y somewhere to have N number of clusters (say 3)



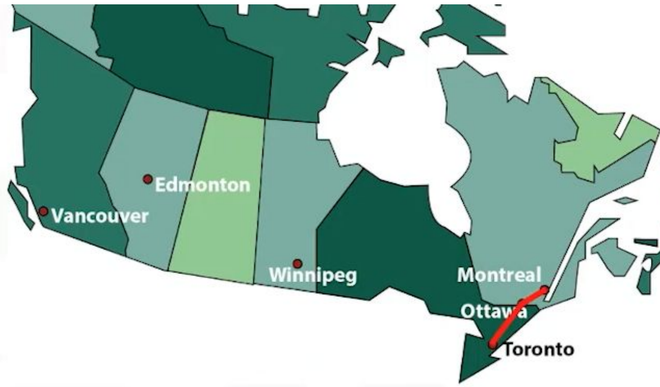
	TO	OT/MO	VA	WI	ED
TO		351	3363	1510	2699
OT/MO			3543	1676	2840
VA				1867	819
WI					1195
ED					

Clustering

Hierarchical / Intro

- Finding Similarity of city locations in Canada
- Dendrogram Y is the similarity
- We can cut Y somewhere to have N number of clusters (say 3)

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



Clustering

Hierarchical / Intro

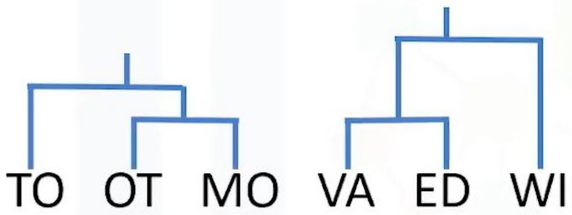
- Finding Similarity of city locations in Canada
- Dendrogram Y is the similarity
- We can cut Y somewhere to have N number of clusters (say 3)

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

Clustering

Hierarchical / Intro

- Finding Similarity of city locations in Canada
- Dendrogram Y is the similarity
- We can cut Y somewhere to have N number of clusters (say 3)

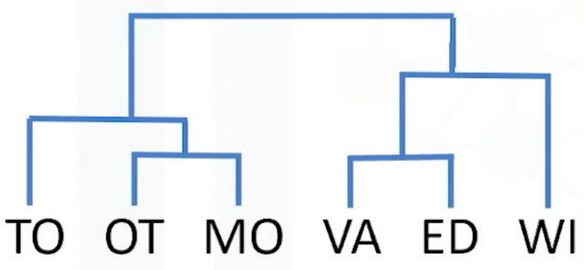


	TO/OT/MO	VA/ED	WI
TO/OT/MO		2840	1676
VA/ED			1667
WI			

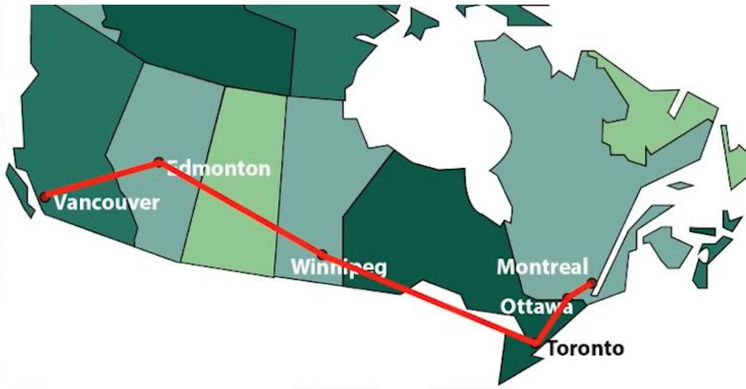
Clustering

Hierarchical / Intro

- Finding Similarity of city locations in Canada
- Dendrogram Y is the similarity
- We can cut Y somewhere to have N number of clusters (say 3)



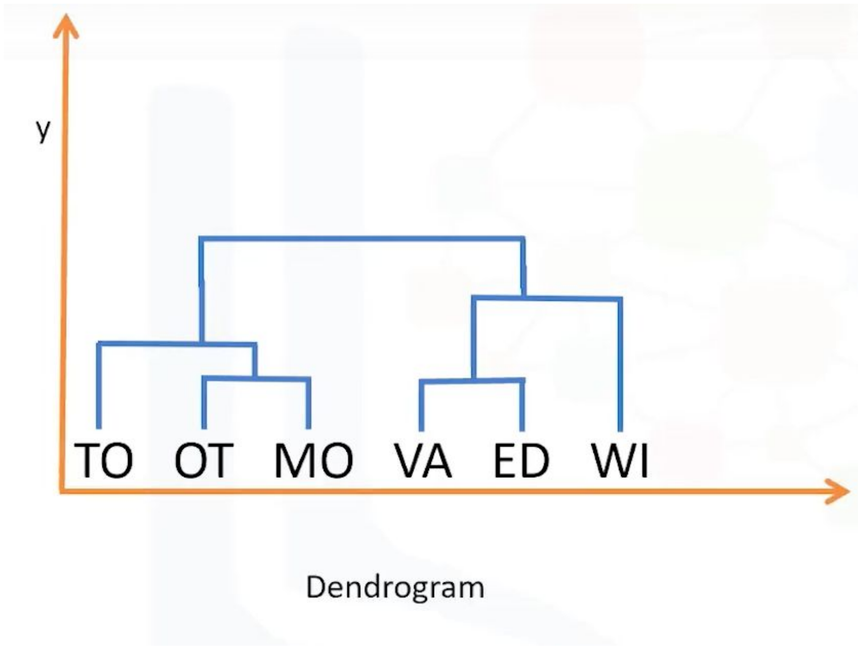
	TO/OT/MO	VA/ED/WI
TO/OT/MO		1676
VA/ED/WI		



Clustering

Hierarchical / Intro

- Finding Similarity of city locations in Canada
- Dendrogram
- We can cut Y somewhere to have N number of clusters (say 3)



Dendrogram

Clustering

Hierarchical / More

1. Create *n* clusters, one for each data point
2. Compute the Proximity Matrix
3. **Repeat**
    i. Merge the two closest clusters
    ii. Update the proximity matrix
4. **Until** only a single cluster remains

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Clustering

Hierarchical / More

- We should be able to calculate distances between data points (again say age, BMI, BP)
- but also need the distance “between” clusters:
  - Single Linkage Clustering: Minimum distance
  - Complete Linkage Clustering: Maximum distance
  - Average Linkage Clustering: average of distances from each point to all other points
  - Centroid Linkage Clustering: centroids of clusters

Clustering

Hierarchical / More

- Pros
  - Works with unknown N
  - Easy to implement
  - Useful dentograms; good for understanding
- Cons
  - Impossible to undo via algorithm
  - long runtimes
  - sometimes difficult to identify the number of clusters (specially for large datasets)

Clustering

Hierarchical / More

- Hierarchical vs K-Means
  - Can be slower
  - Does not require the number of clusters to run
  - Gives more than one partitioning
  - Always generate the same clusters

Clustering

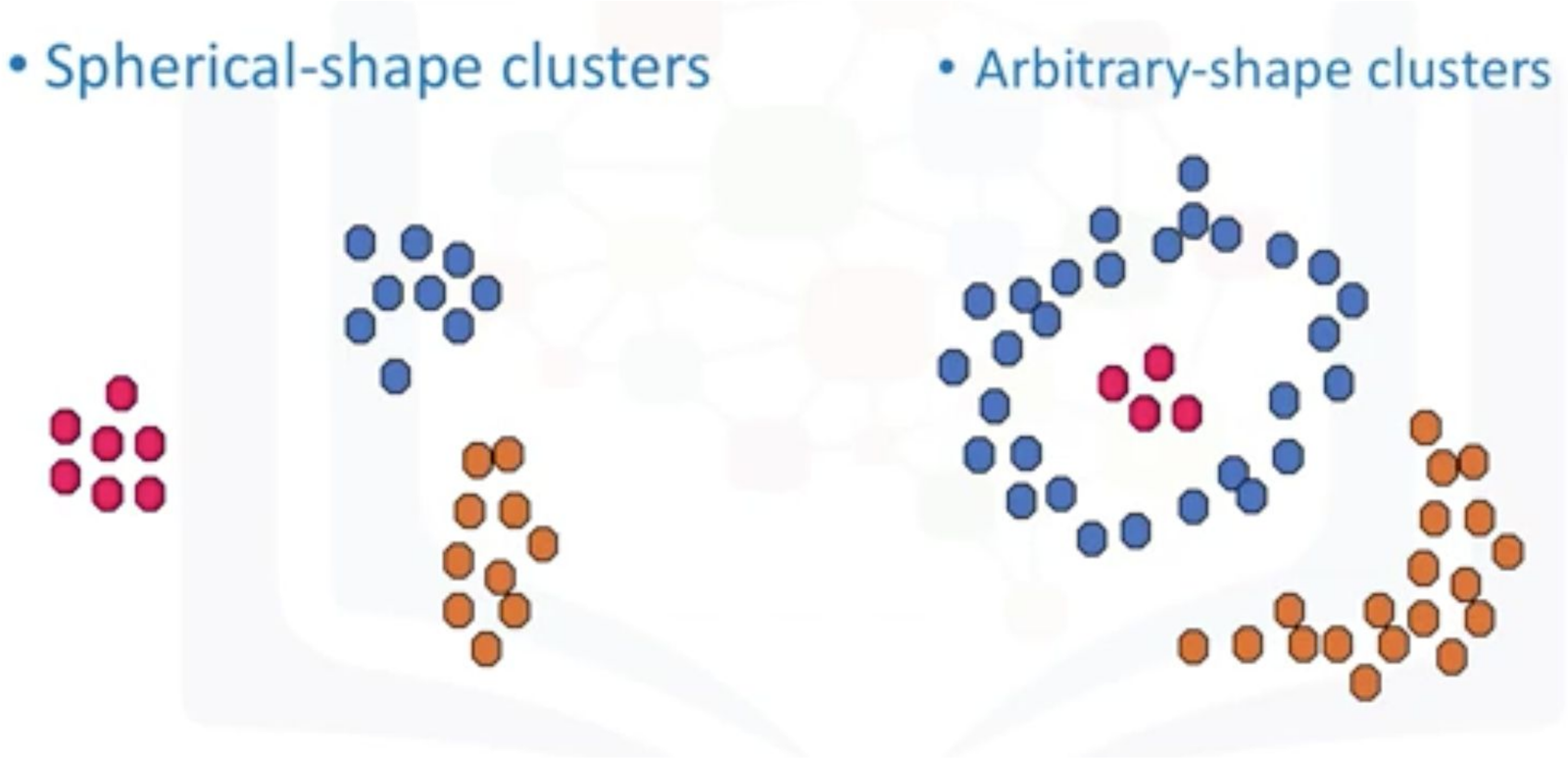
Hierarchical / Lab

-

Clustering

DBSCAN

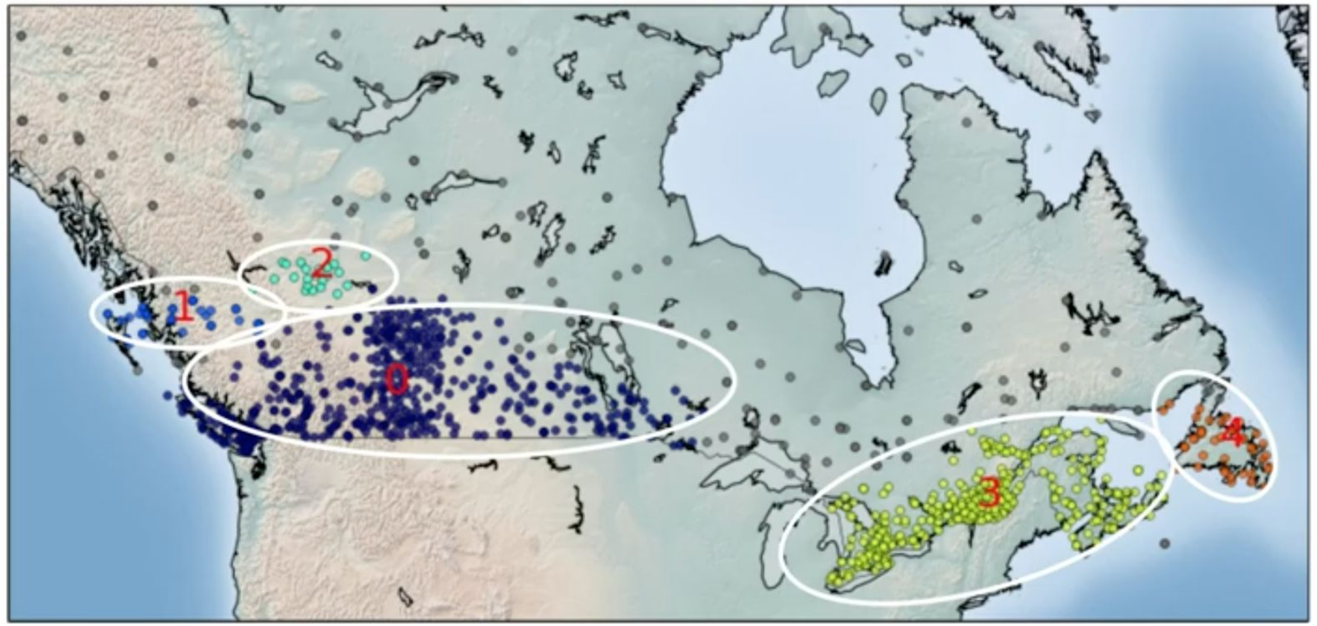
- K-Means will assign every datapoint to a cluster; no outlier
- Density Based clusters will find dense areas and will separate outliers. Good for anomaly detection
- Density: number of points within a radius



Clustering

DBSCAN

- DBSCAN algorithms is effective for tasks like class identification
- effective even in presence of noise
- Grouping same weather on dense areas



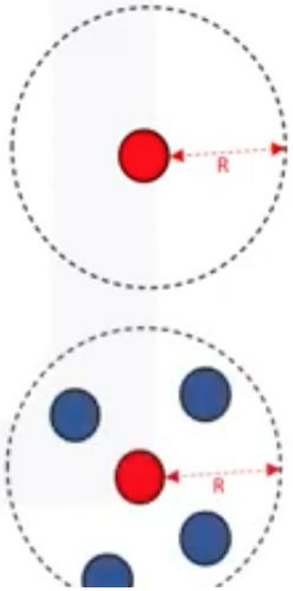
Clustering

DBSCAN

- DBSCAN (**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise)
  - Is one of the most common clustering algorithms
  - Works based on density of objects

- R (**R**adius of neighborhood)
  - Radius (R) that if includes enough number of points within, we call it a dense area

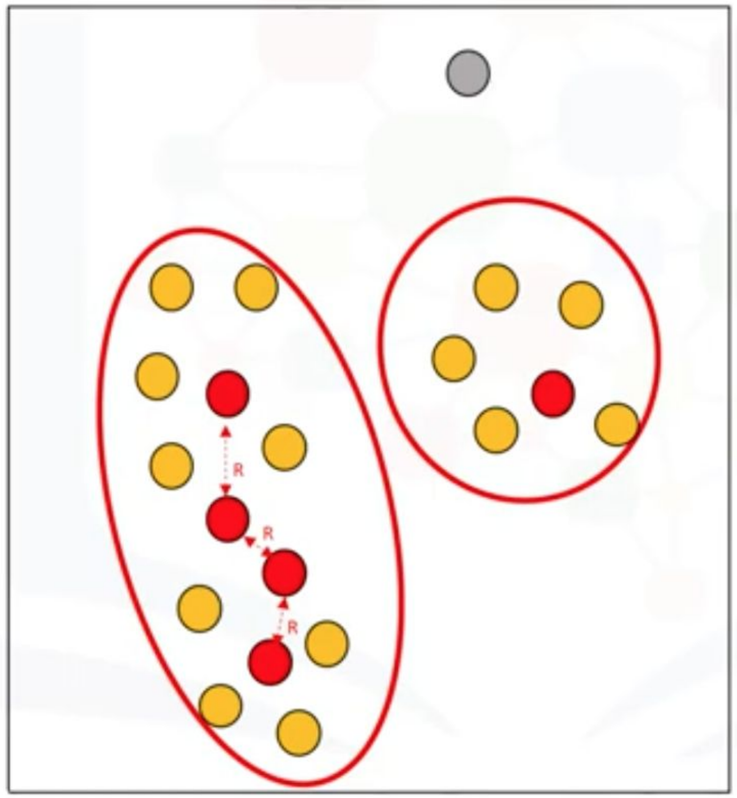
- M (**M**in number of neighbors)
  - The minimum number of data points we want in a neighborhood to define a cluster



Clustering

DBSCAN

- point types:
  - core: within our neighborhood of the point there are at least M points.
  - Border:
    - less than M in neighborhood
    - reachable from a core point
  - outlier is not core neigher a border



R = 2unit , M = 6

Clustering

DBSCAN

- Arbitrarily shaped clusters
- Robust to outliers
- Does not require specification of the number of clusters

Clustering

DBSCAN / Lab

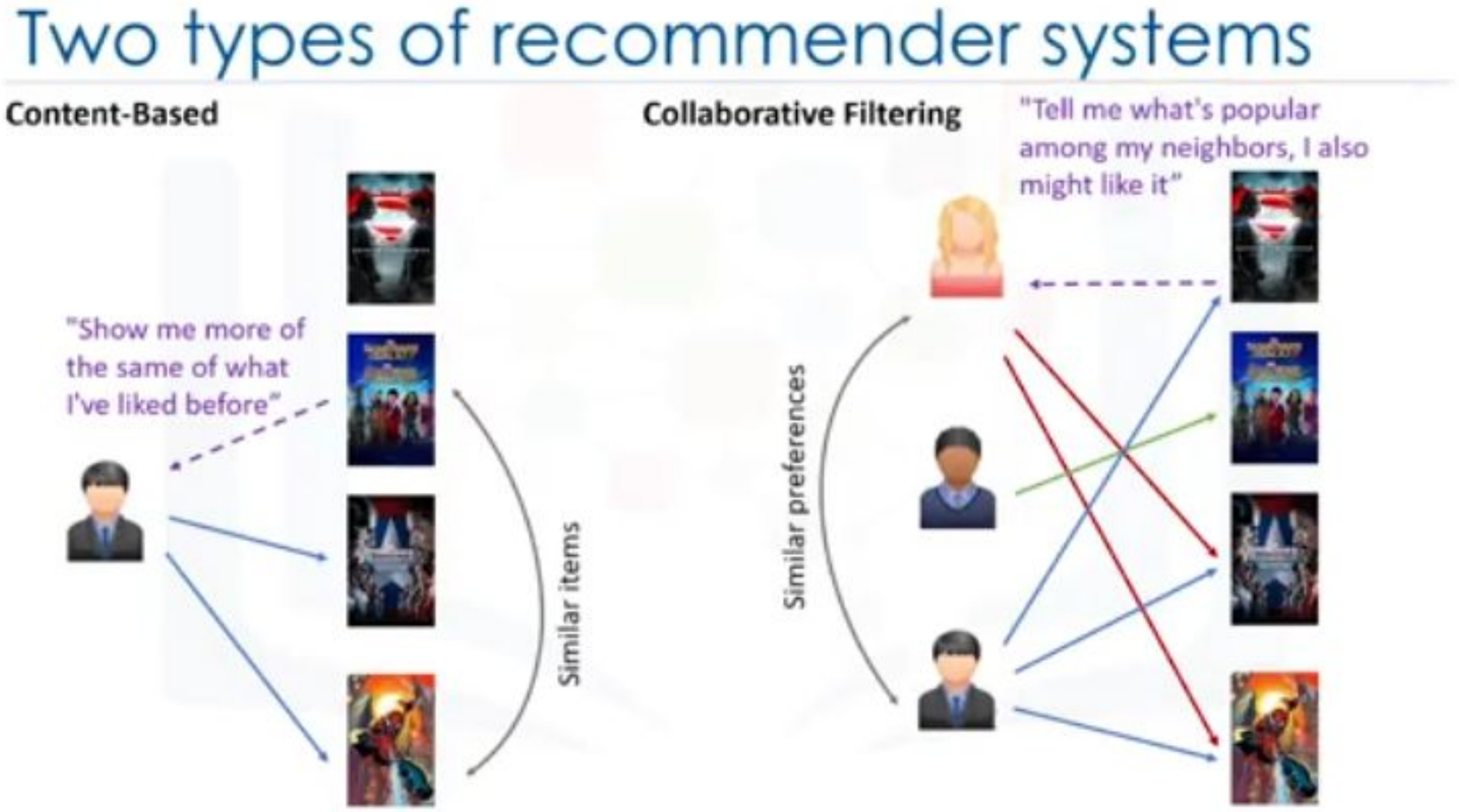
Recommenders

Intro

- Peoples tastes follow patterns (say books)
- Recommender systems capture the pattern of people’s behaviour and use it to predict what else they might want or like
- Many applications. Netflix, Amazon, facebook, twitter, News, Digikala, SnapFood
- Broader Exposure -> More Usage

Recommenders

Intro / types



Recommenders

Intro / types

- Memory Based
  - Uses the entire user-item dataset to generate a recommendation
  - Uses statistical techniques to approximate users or items (Pearson Correlation, Cosine Similarity, Euclidean Distance, …)
- Model Based
  - Develops a model of users in an attempt to learn their preferences
  - Models can be created using ML techniques like regression, clustering, classification, ...

Recommenders

Content Based

- Works based on users profiles
- Works with user ratings (like, view, …) and then finds the similarity between content of those contents (tags, category, genres, …)
-

Recommenders

Content Based

Recommenders

Content Based

Weighing the genres

	2
	10
	8

Input User Ratings

X

	Comedy	Adventure	Super Hero	Sci-Fi
	0	1	1	0
	1	1	1	1
	1	0	1	0

Movies Matrix

=

Weighted Genre Matrix

	Comedy	Adventure	Super Hero	Sci-Fi
	0	2	2	0
	10	10	10	10
	8	0	8	0

User Profile

	Comedy	Adventure	Super Hero	Sci-Fi
	0.3	0.2	0.33	0.16

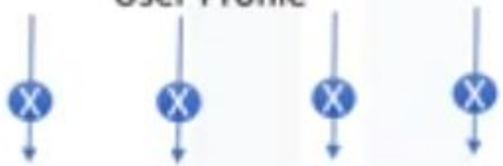
COGNITIVE CLASS

Recommenders

Content Based

	Comedy	Adventure	Super Hero	Sci-Fi
	0.3	0.2	0.33	0.16

User Profile



	Comedy	Adventure	Super Hero	Sci-Fi
	1	1	0	1
	0	0	1	0
	1	0	1	0

Movies Matrix

=

	Comedy	Adventure	Super Hero	Sci-Fi
	0.3	0.2	0	0.16
	0	0	0.33	0
	0.3	0	0.33	0

Weighted Movies Matrix

Weighted Average
0.66
0.33
0.63

Recommendation Matrix

Recommenders

Content Based

- LAB

Recommenders

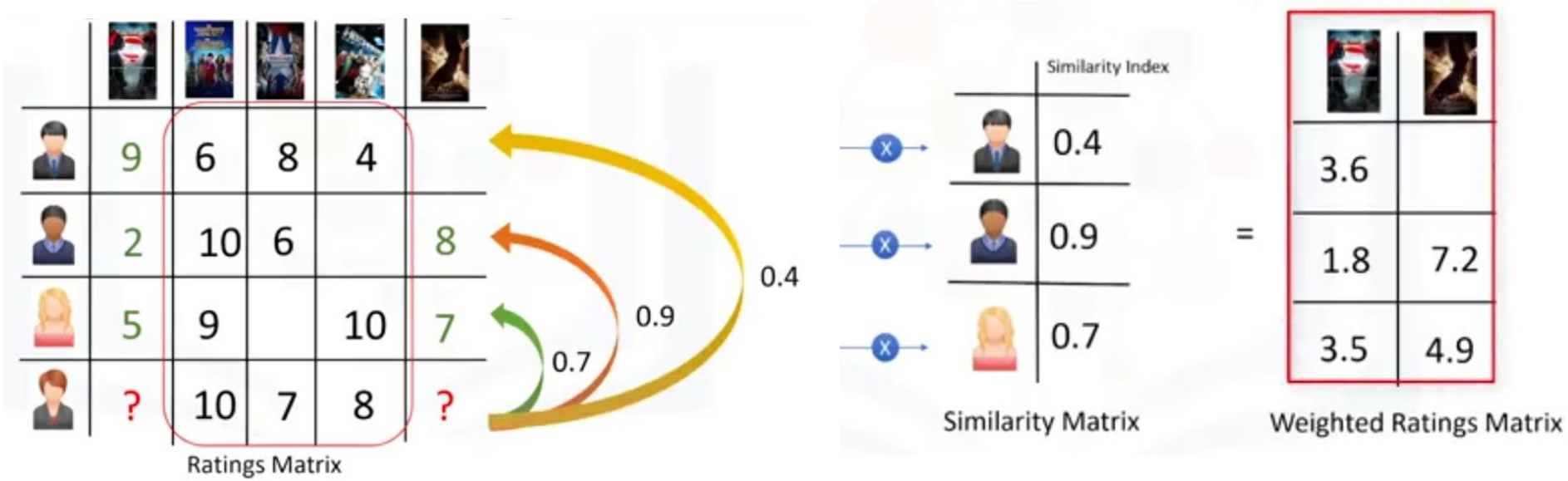
Collaborative Filtering

- User Based
  - Based on the user’s similarity or neighborhoods
  - Finds similarity between users (say likings history)
- Item-Based
  - Based on items similarity

Recommenders

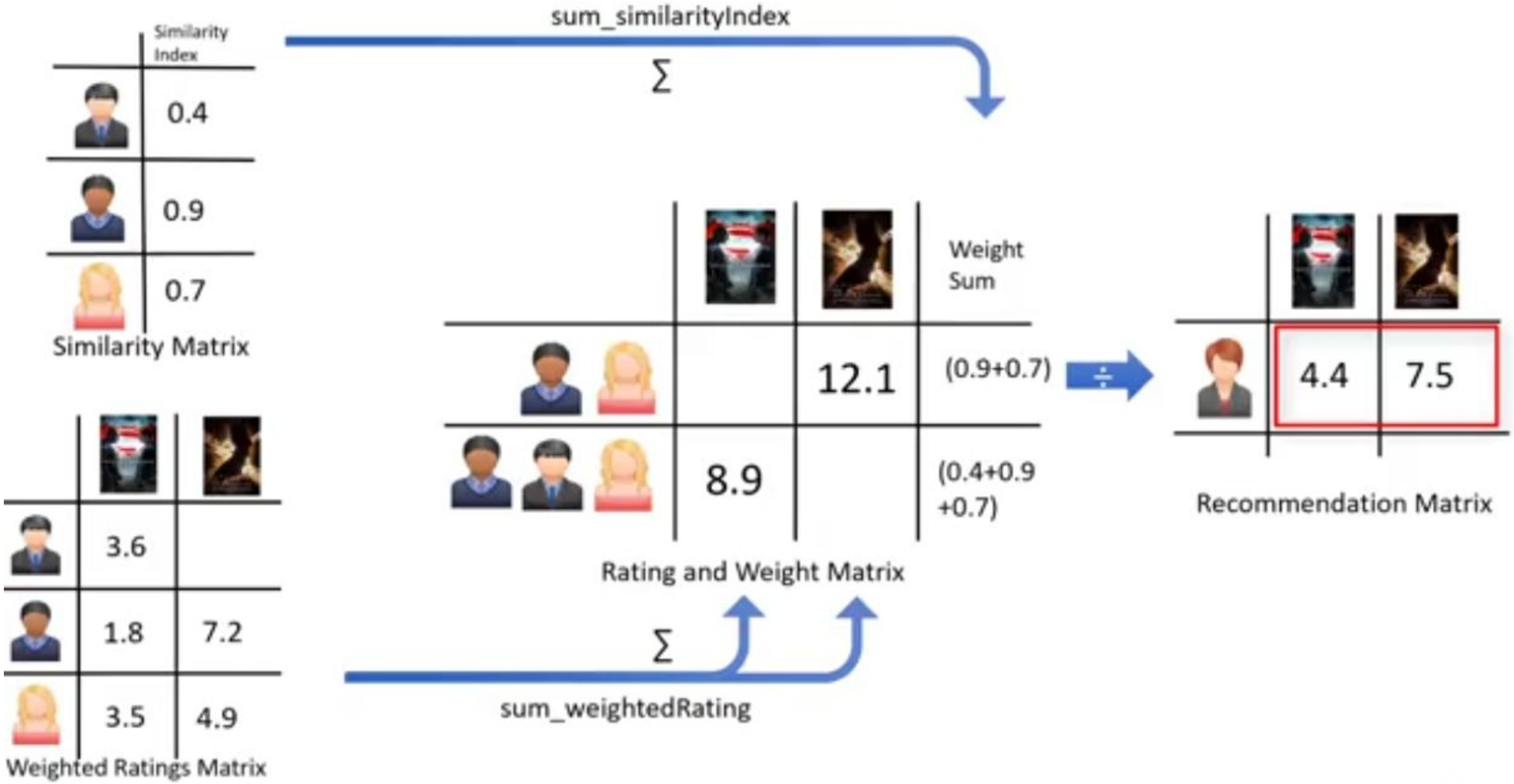
Collaborative Filtering / User Based

•



Recommenders

Collaborative Filtering / User Based



Recommenders

Collaborative Filtering / Item Based

Recommenders

Collaborative Filtering / Challenges

- Data Sparsity
  - Large users but they are rating only a limited number of items
- Cold Start
  - What if a new user joins the system? What if a new Item is added?
- Scalability
  - Drops performance when items/users are increased. Matrix becomes larger and larger
    - There are solutions.. like using hybrid solutions

Recommenders

Collaborative Filtering

- LAB