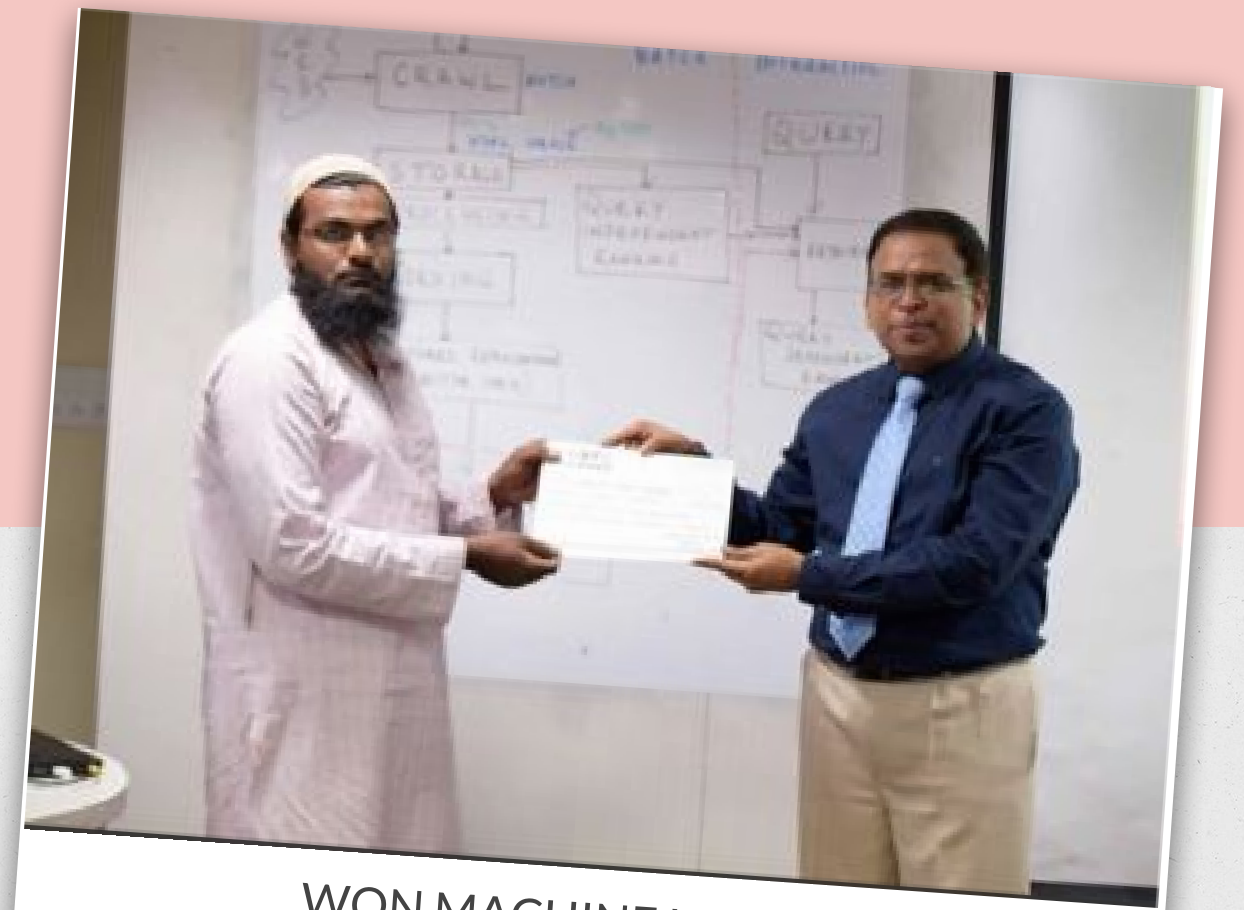
Mohamed Noordeen Alaudeen

Senior Data Scientist - Logitech

About me

TIGER ANALYTICS

logitech



WON MACHINE LEARNING CONTEST

WON BEST PROJECT AWARD

International School of Engineering

awards

Certificate in Engineering Excellence

in Big Data Analytics and Optimization

to

Mohamed Noordeen Alaudeen

on successful completion of all the requirements of the 352-hour program

conducted between December 10, 2016 and June 04, 2017 followed by a project defense.

This program is certified for quality of content, assessment and pedagogy by the Language Technologies Institute (LTI) of Carnegie Mellon University (CMU). LTI also provided assistance in curriculum development for this program.

INSOFE

Dated this second day of August, two thousand and seventeen.

lti

Dr. Dakshinamurthy V Kolluru
President

Dr. Sridhar Pappu
Executive VP - Academics

BEST STUDENT OF THE BATCH

GUEST LECTURES

NOT THE FIRST TIME FOR US

MY WORK

https://github.com/nursnaaz
https://www.linkedin.com/in/nursnaaz/
http://www.technaaz.com/

Data Preprocessing

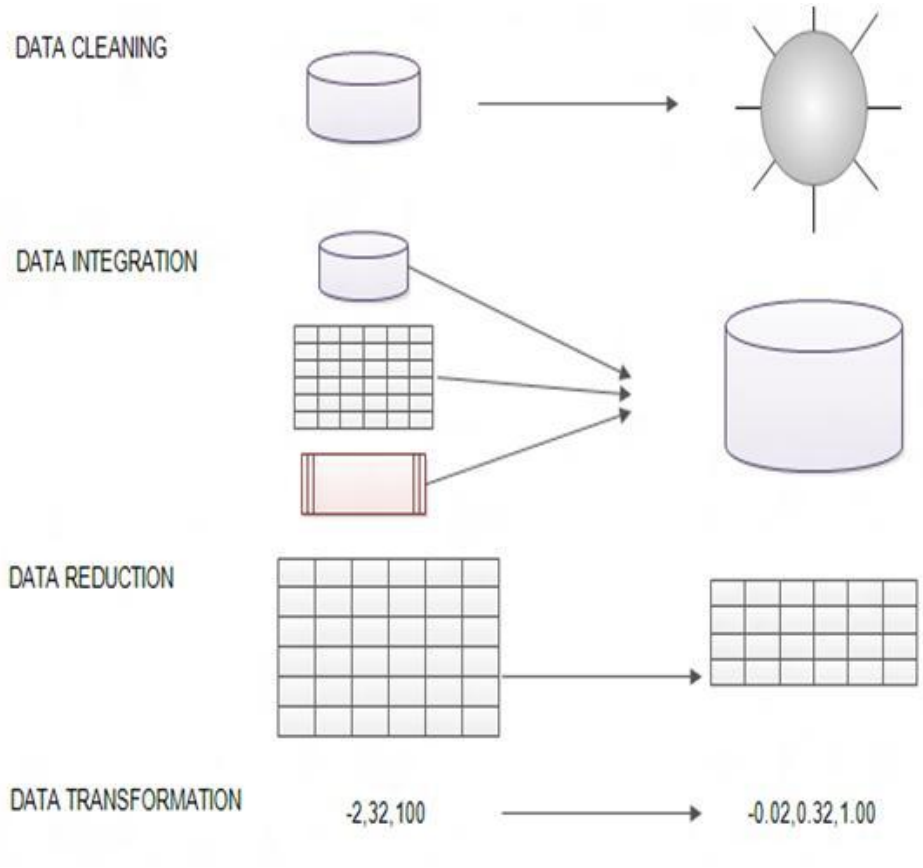
Real World Data

Any Problem?

S.No	Credit_rating	Age	Income	Credit_cards
1	0.00	21	10000	y
2	1.0		2500	n
3	2.0	62	-500	y
4	100.012	42		n
5	yes	200	1	y
6	30	0	Seventy thousand	No

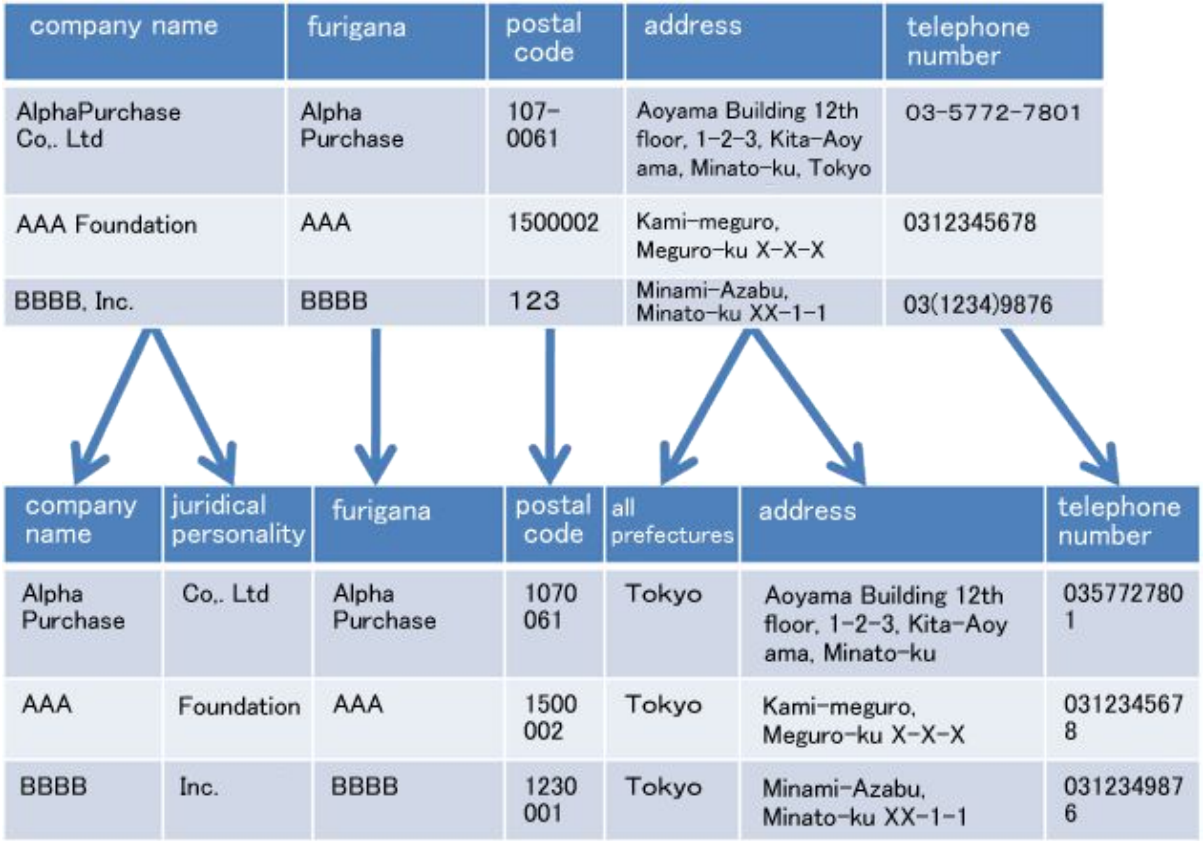
Data Preprocessing

- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation



Data Cleaning

1. Missing Data
   - Central Imputation
   - KNN Imputation
2. Noisy Data
   - Smoothing
   - Clustering
3. Outlier Removal
   - Using Boxplot



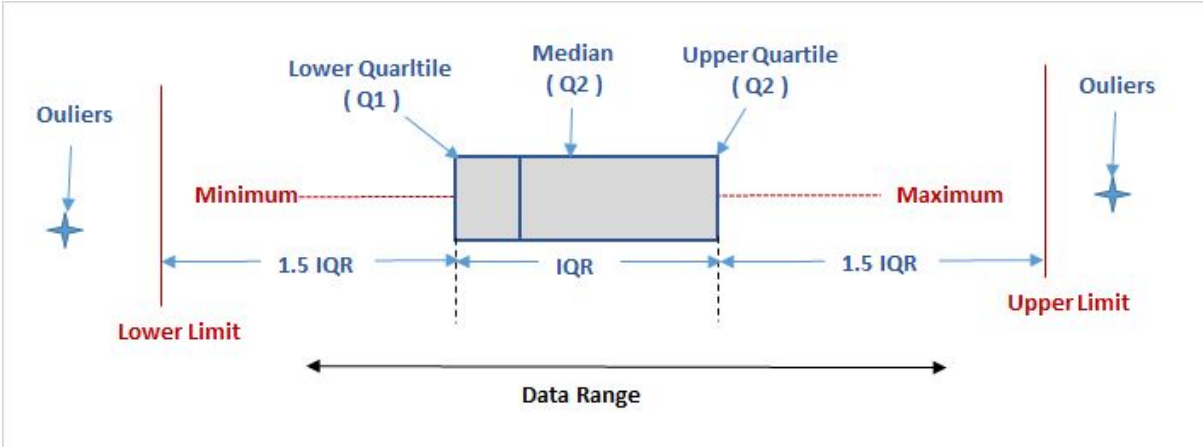
Imputation

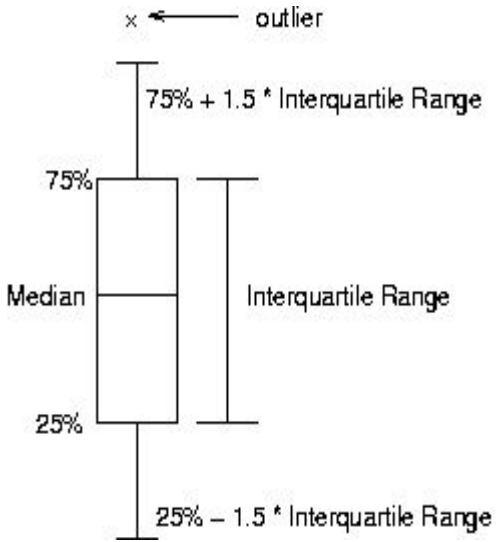
- Replace with mean or a median
- When to use mean?
- Replace with nearest neighbour
- How much nearest to see?

S.No	Qualification	Age	Income
1	B.Tech	25	30k
2	M.Tech	30	50k
3	B.Tech	26	32k
4	B.Tech	25	?
5	M.Tech	29	60k
6	B.Tech	?	30k

Outlier

- BoxPlot





Standard Deviation

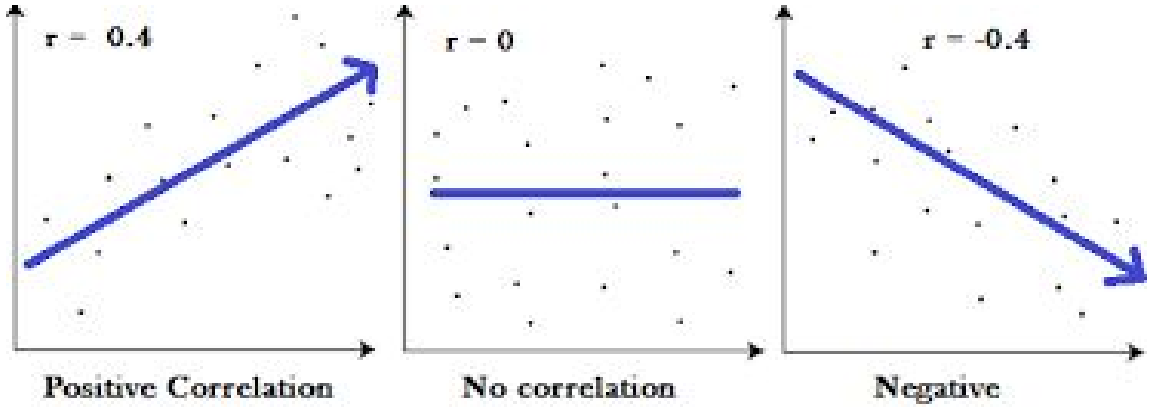
$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{n}}$$

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n - 1}}$$

S.No	x	x^2	x-mean	Abs(x-mean)	(x-mean)^2
1	5	25	0.3333333333	0.3333333333	0.1111111111
2	7	49	2.333333333	2.333333333	5.444444444
3	4	16	-0.6666666667	0.6666666667	0.4444444444
4	2	4	-2.666666667	2.666666667	7.111111111
5	6	36	1.333333333	1.333333333	1.777777778
6	2	4	-2.666666667	2.666666667	7.111111111
7	8	64	3.333333333	3.333333333	11.11111111
8	5	25	0.3333333333	0.3333333333	0.1111111111
9	3	9	-1.666666667	1.666666667	2.777777778
SUM	42	232	0	15.33333333	36
Average	4.666666667	25.77777778			

Data Integration

- Check for correlation
- Remove uncorrelated data



$$r = \frac{\sum (x - \overline{x})(y - \overline{y})}{\sqrt{\sum (x - \overline{x})^2 \sum (y - \overline{y})^2}}$$

Data Transformation

- Normalization

Min-max normalization

1. Min Max Normalization
2. Z - Score Normalization
3. Decimal scaling

Decimal scaling

v= v/10^j

Normalization: Example II

- **Min-Max normalization on an employee database**
  - max distance for salary: 100000-19000 = 81000
  - max distance for age: 52-27 = 25
  - New min for age and salary = 0; new max for age and salary = 1

$$x'_i = \frac{x_i - \min x_i}{\max x_i - \min x_i}(new\max - new\min) + new\min$$

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

ID	Gender	Age	Salary
1	1	0.00	0.00
2	0	0.96	0.56
3	0	1.00	1.00
4	1	0.24	0.44
5	0	0.72	0.32

Normalization: Example

- **z-score normalization: v' = (v - Mean) / Stdev**
- **Example: normalizing the "Humidity" attribute:**

Humidity
85
90
78
96
80
70
65
95
70
80
70
90
75
80

Mean = 80.3
Stdev = 9.84

Humidity
0.48
0.99
-0.23
1.60
-0.03
-1.05
-1.55
1.49
-1.05
-0.03
-1.05
0.99
-0.54
-0.03

Data Reduction

- Data Cube Aggregation

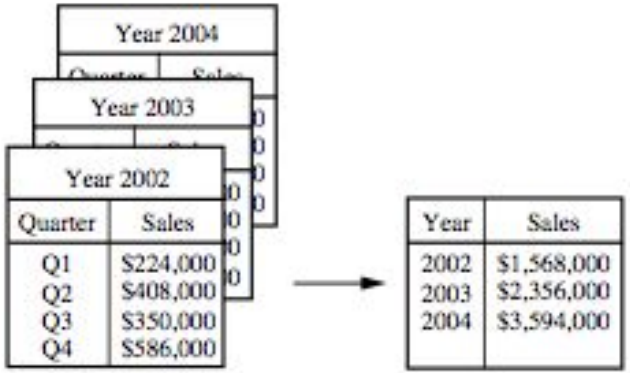


Figure 2.13 Sales data for a given branch of *AllElectronics* for the years 2002 to 2004. On the left, the sales are shown per quarter. On the right, the data are aggregated to provide the annual sales.

Find the Relationship

x	Y
2	6
6	14
4	10
3	8
7	16
4	10
2	6
5	12

Relationship

$$Y = 2 + 2(X)$$

x	Y
2	6
6	14
4	10
3	8
7	16
4	10
2	6
5	12

What is 2 here?

$$Y = 2 + 2(X)$$

x	Y
2	6
6	14
4	10
3	8
7	16
4	10
2	6
5	12

Find the Y in ?

Y = 2 + 2(X)

x	Y
2	6
6	14
4	10
3	8
7	16
4	10
2	6
5	12
10	?
1	?

Terminology

$$Y = 2 + 2(X)$$

Y = Model

2 = Intercept

2 in 2x = Slope

X = input

x	Y
2	6
6	14
4	10
3	8
7	16
4	10
2	6
5	12
10	**?**
1	?

Formula for a line

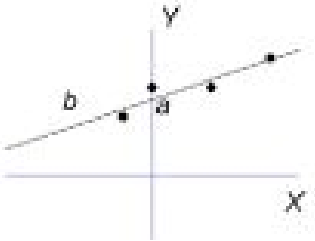
Linear regression equation
(without error)

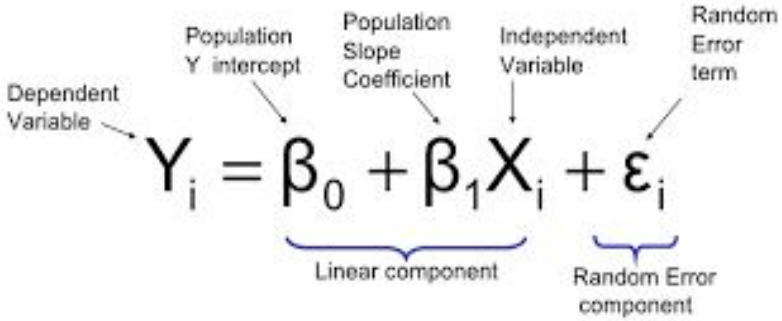
$$\hat{Y} = bX + a$$

predicted values of Y

b = slope = rate of predicted ↑/↓ for Y scores for each unit increase in X

Y-intercept = level of Y when X is 0





Linear Regression

Welcome to the world of data science

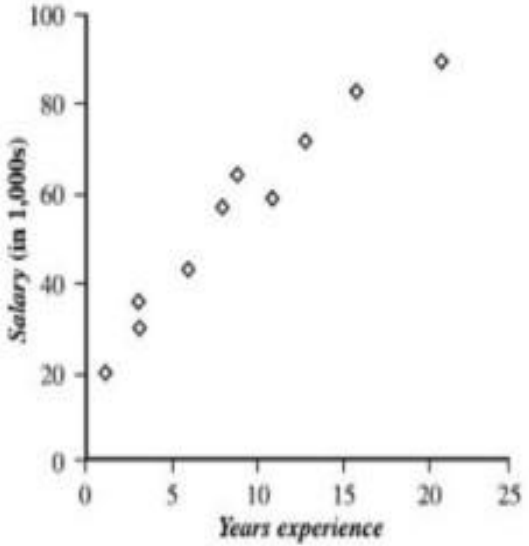
What is linear?

What is Regression?

Help me in finding the relationship?

Salary data.

x years experience	*y* salary (in $1000s)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83



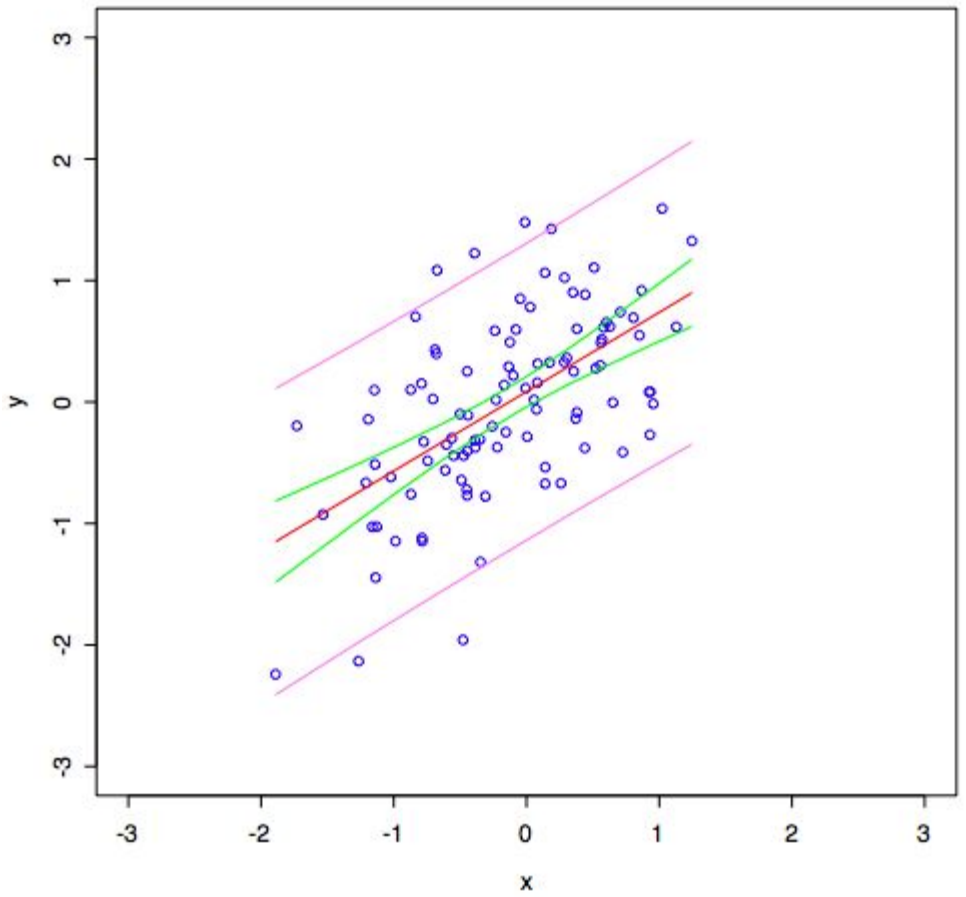
Gradient Descent

Finding the optimum relationship where the error is minimal.

Finding the intercept and coefficients value.

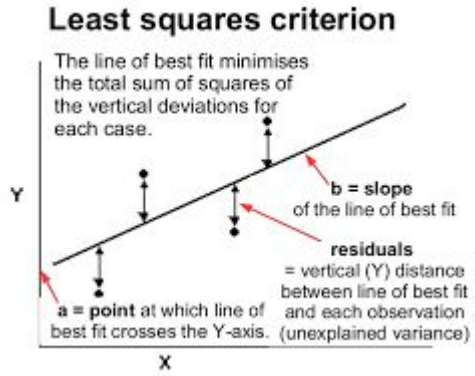
Find the solution?

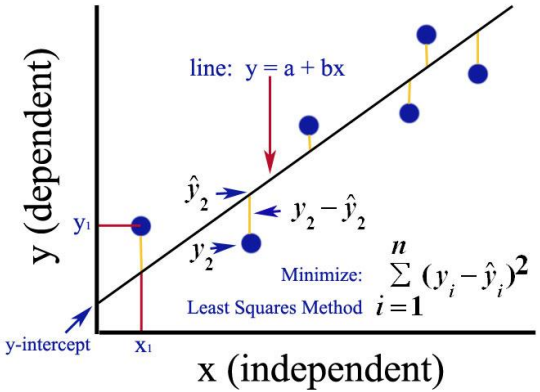
Any Suggestions?

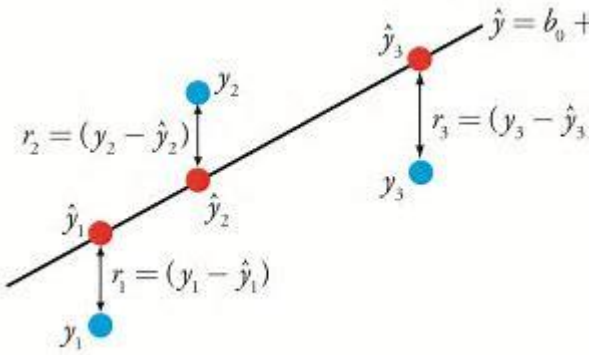


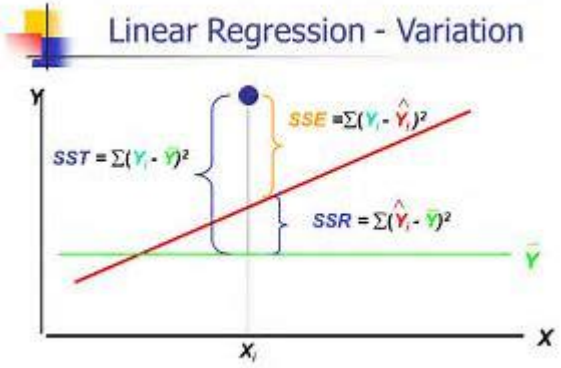
Line of best fit

Ordinary least square line









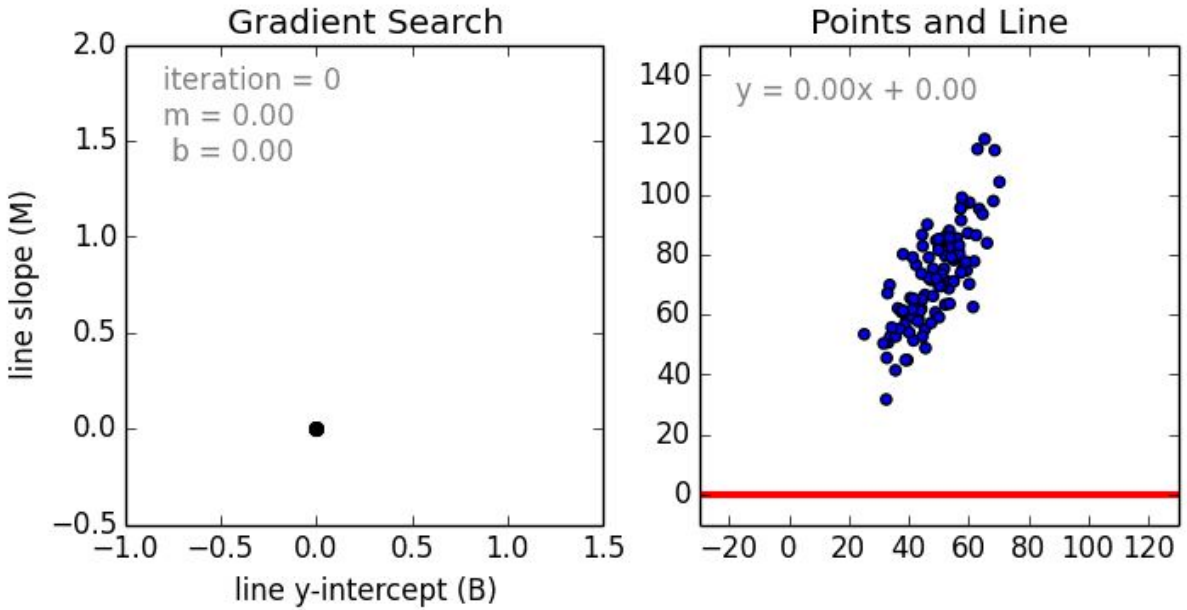
Cost Function

$$\text{Error}_{(\text{m,b})} = \frac{1}{N}\sum_{i=1}^{N}(y_i - (mx_i + b))^2$$

Gradient Descent

Learning Rate

Momentum



Partial Derivative

Finding the direction of coefficient and slope moves in.

$$\frac{\partial}{\partial \mathrm{m}} = \frac{2}{N}\sum_{i=1}^{N} - x_i(y_i - (mx_i + b))$$

$$\frac{\partial}{\partial \mathrm{b}} = \frac{2}{N}\sum_{i=1}^{N} - (y_i - (mx_i + b))$$

Advantage of Linear Regression

- Linear regression implements a statistical model that, when relationships between the independent variables and the dependent variable are almost linear, shows optimal results.
- Best place to understand the data analysis
- Easily Explicable

Disadvantages

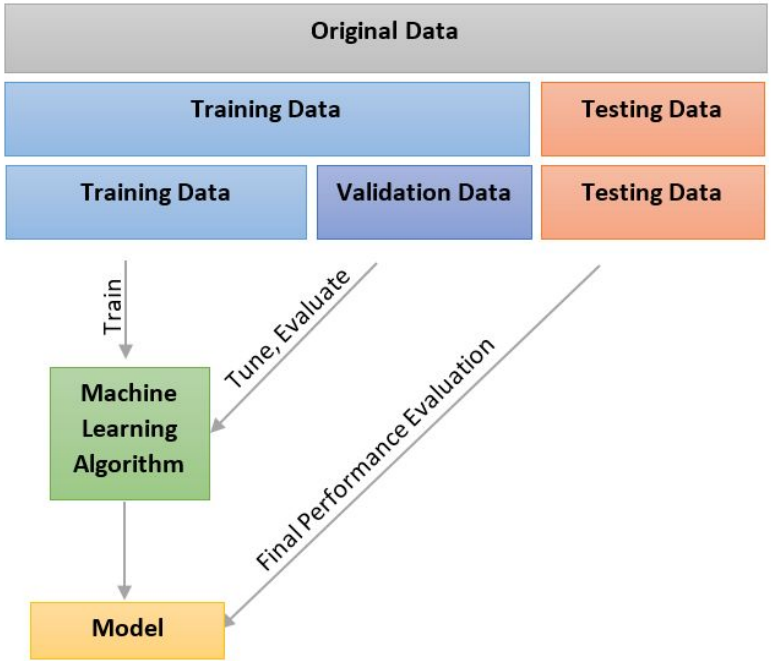
- Linear regression is often inappropriately used to model non-linear relationships.
- Linear regression is limited to predicting numeric output.
- A lack of explanation about what has been learned can be a problem.
- Prone to bias variance problem

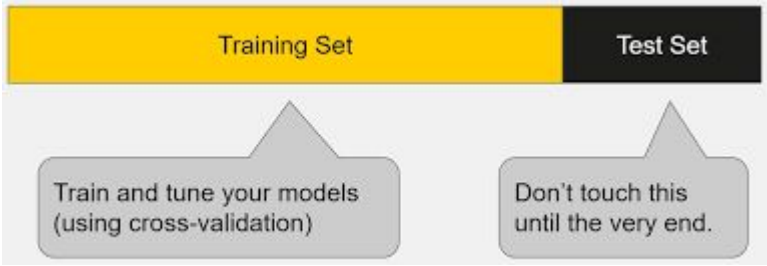
Error Metrics for Regression

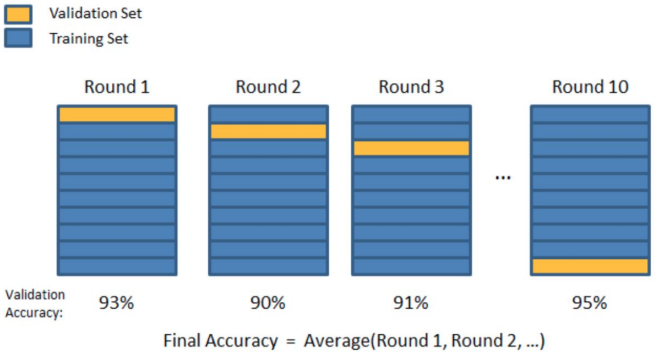
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

$$r^2 = 1 - \frac{\text{SS Error}}{\text{SS Total}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

How to evaluate our model?







Overfitting vs Underfitting

Training Data(Less Error)

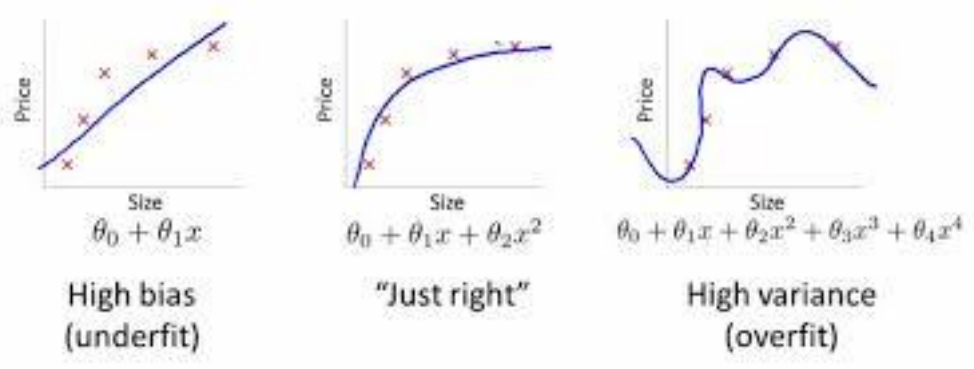
Testing Data (More Error)

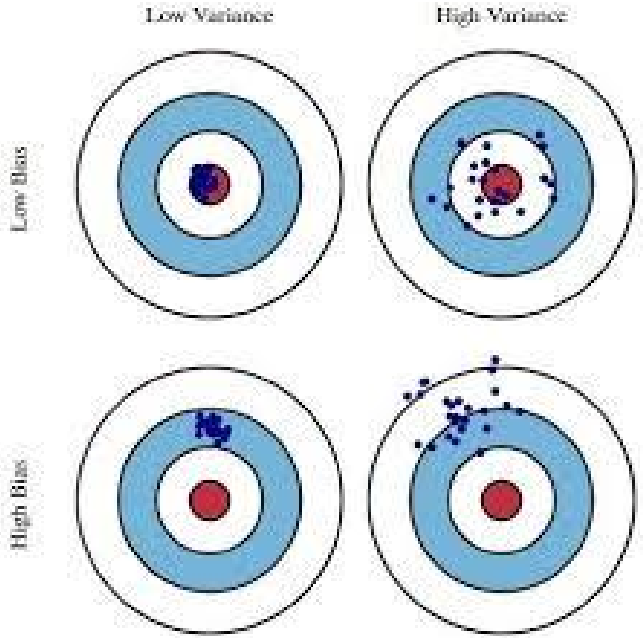
Overfitting vs Underfitting

Training Data (More Error)

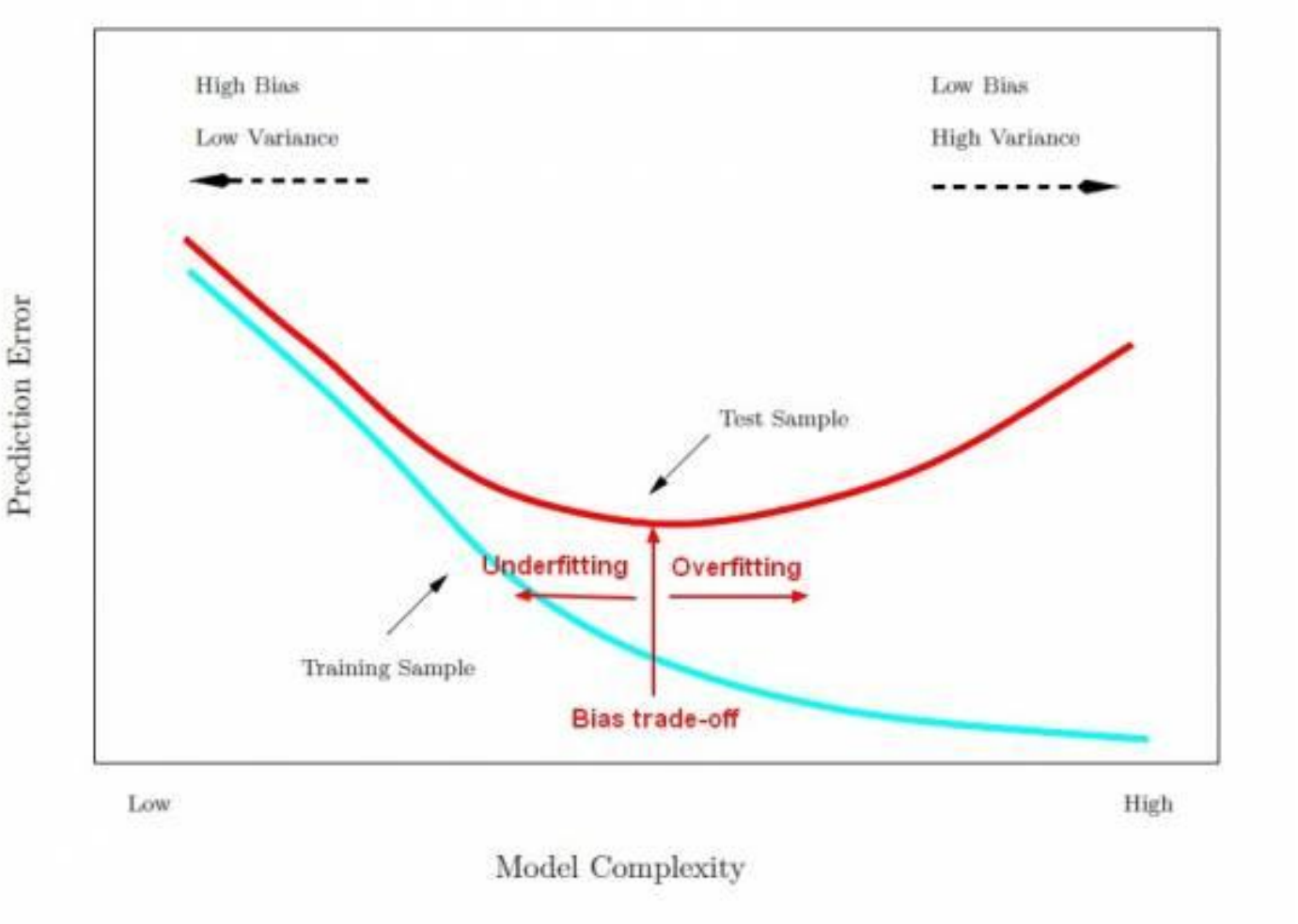
Testing (Still More Error)

Variance and Bias Trade off





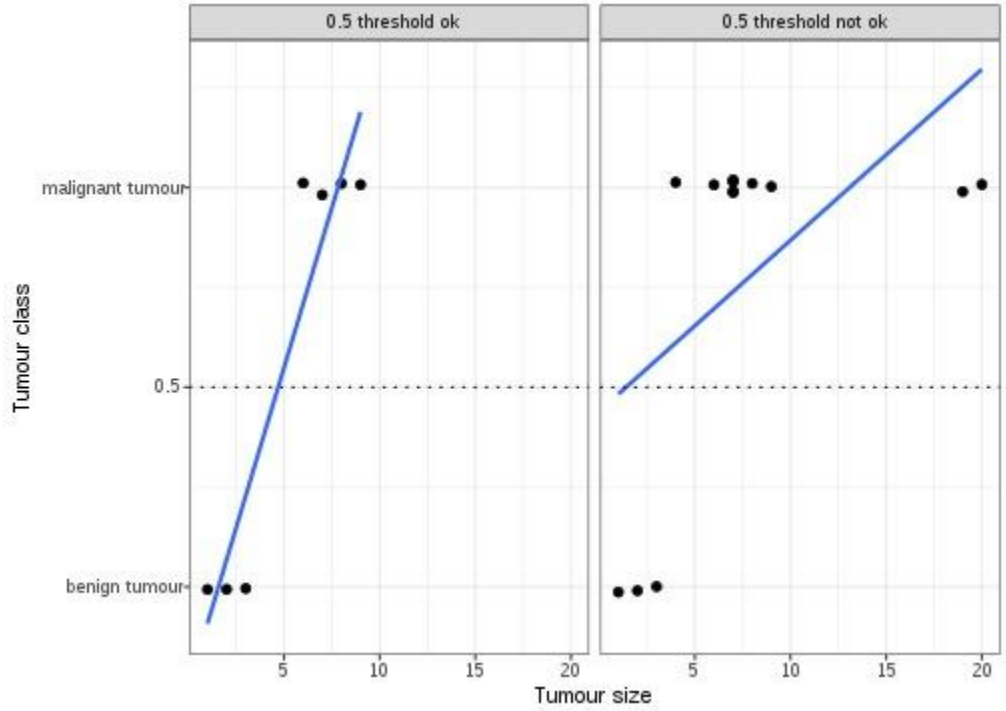
Ideal Model should have Low varinance and Low Bias

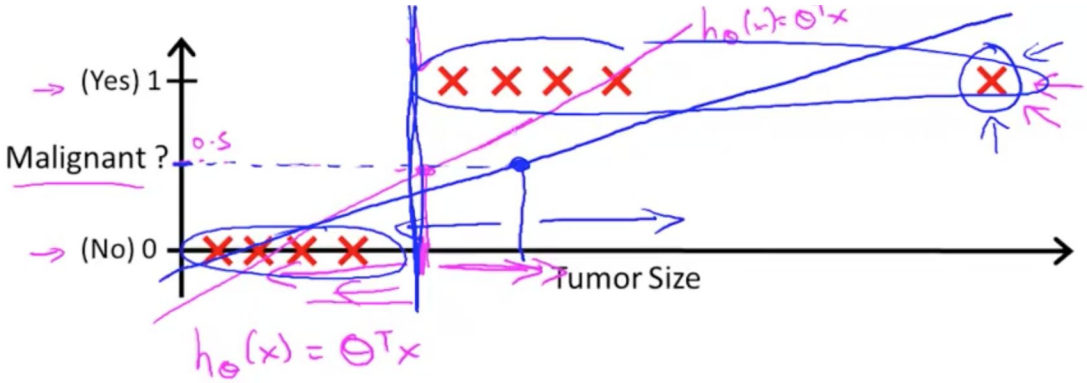

High Bias
Low Variance
Low Bias
High Variance
Prediction Error
Test Sample
Underfitting
Overfitting
Training Sample
Bias trade-off
Low
High
Model Complexity

Logistic Regression

Find who will likely to get Cardiovascular Disease

Cigarettes/Day	BMI	CVD
4	27	**yes**
6	25	**yes**
5	15	**No**
2	21	**No**
1	30	**Yes**
0	28	**No**
10	25	**Yes**


0.5 threshold ok
0.5 threshold not ok
malignant tumour
0.5
benign tumour
Tumour class
5
10
15
20
Tumour size



→ Threshold classifier output $h_\theta(x)$ at 0.5:

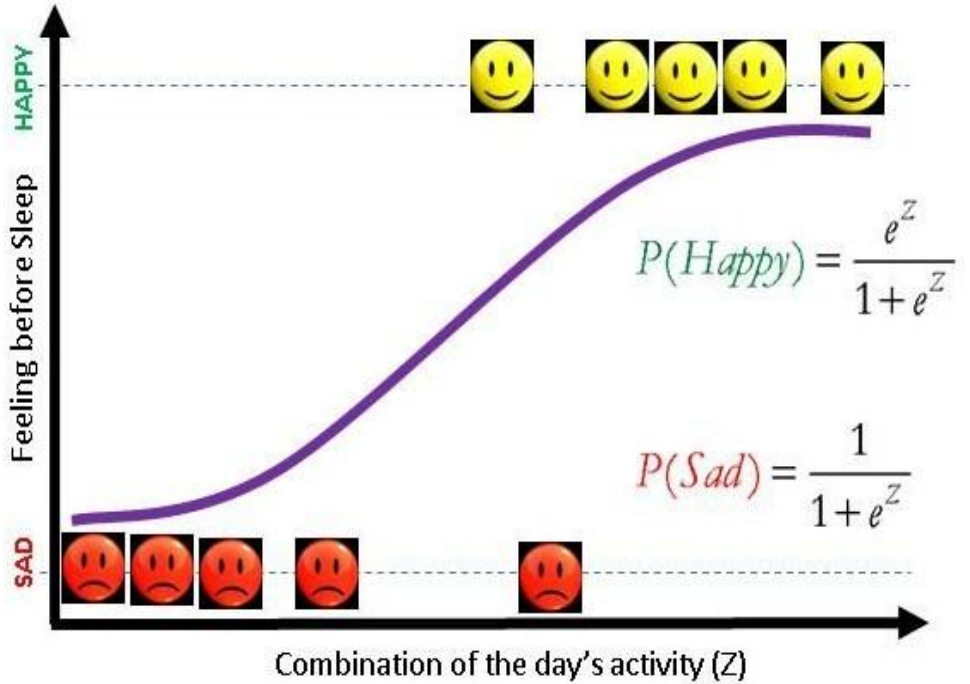
→ If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

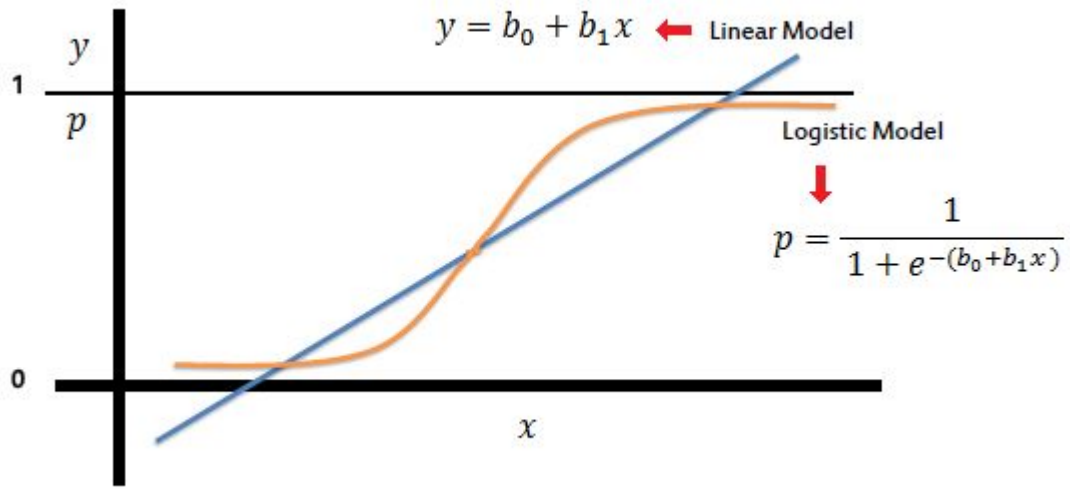
$$\ln\left(\frac{P}{1-P}\right) = a + bX$$

$$\frac{P}{1-P} = e^{a+bX}$$

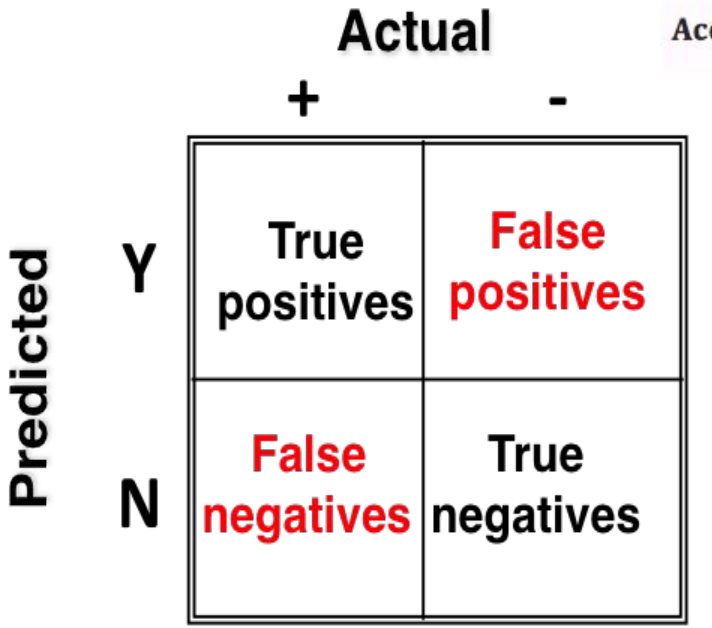
$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$



Logistic Regression Equation



Error Metrics



$$\text{Accuracy} = \frac{\text{True positives} + \text{True negatives}}{\text{True positives} + \text{False positives} + \text{True negatives} + \text{False negatives}}$$

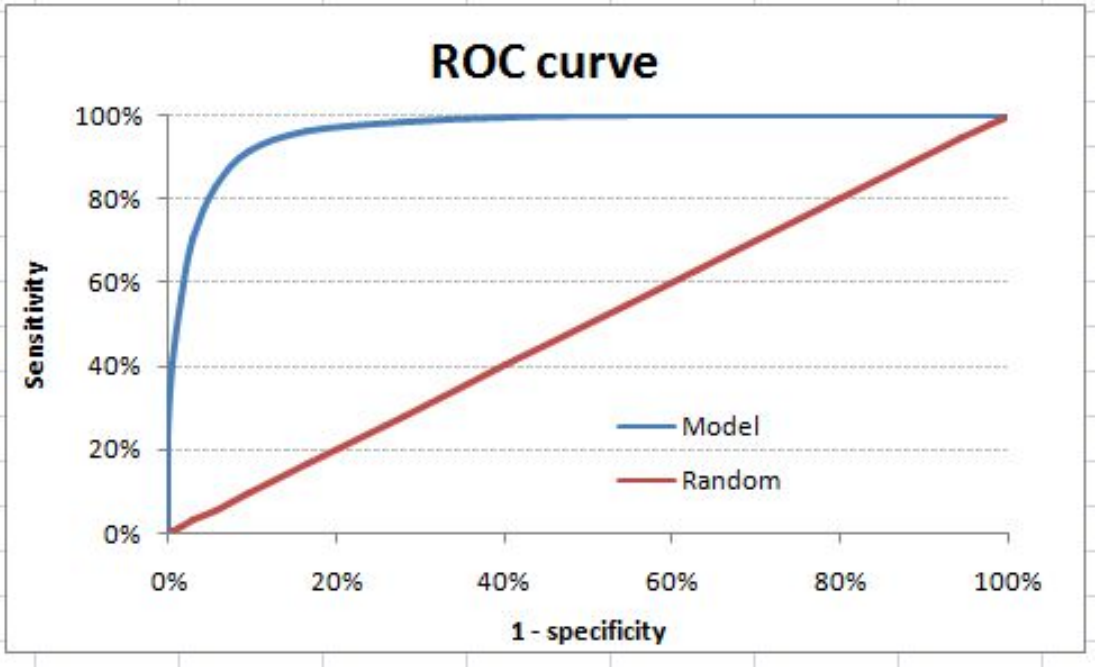
- **true positive rate** = TP/(TP+FN) = 1 − false negative rate
- **false positive rate** = FP/(FP+TN) = 1 − true negative rate
- **sensitivity** = true positive rate
- **specificity** = true negative rate
- **positive predictive value** = TP/(TP+FP)
- **recall** = TP / (TP+FN) = true positive rate
- **precision** = TP / (TP+FP)
- **F-score** is the harmonic mean of precision and recall:
- **G-score** is the geometric mean of precision and recall:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

Count of ID	Target			
Model	**1**	**0**	**Grand Total**	
1	3,834	639	4,473	85.7%
0	16	951	967	1.7%
Grand Total	**3,850**	**1,590**	**5,440**	
	99.6%	40.19%		88.0%

ROC CURVE



- 90-1 = excellent (A)
- .80-.90 = good (B)
- .70-.80 = fair (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)

Logistic Regression Pros

- Convenient probability scores for observations
- Efficient implementations available across tools
- Multicollinearity is not really an issue and can be countered with L2 regularization to an extent
- Wide spread industry comfort for logistic regression solutions [oh that’s important too!]

Logistic Regression Cons

- Doesn’t perform well when feature space is too large
- Doesn’t handle large number of categorical features/variables well
- Relies on transformations for non-linear features
- Relies on entire data [Not a very serious drawback I’d say]

K-Nearest Neighbors

KNN Algorithm

1. To classify document d into class c

2. Define k-neighborhood N as k nearest neighbors
(according to a given distance or similarity measure) of d

3. Count number of documents kc in N that belong to c

4. Estimate P(c|d) as kc/k

5. Choose as class argmaxc P(c|d) [= majority class]

Finding similar rows

Distance functions

Euclidean

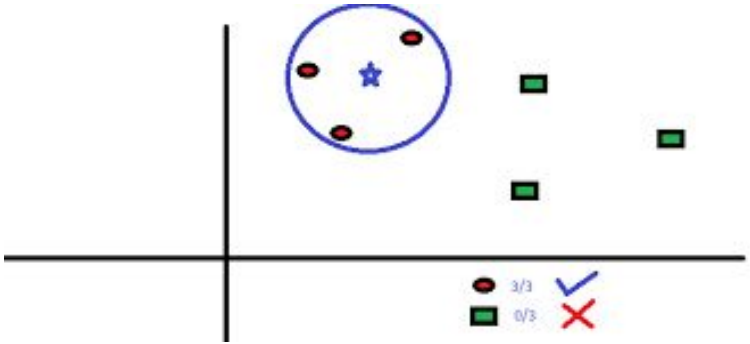
$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan

$$\sum_{i=1}^{k}|x_i - y_i|$$

Minkowski

$$\left(\sum_{i=1}^{k}\left(|x_i - y_i|\right)^q\right)^{1/q}$$



KNN Pros

- Ease to interpret output
- Calculation time
- Predictive Power

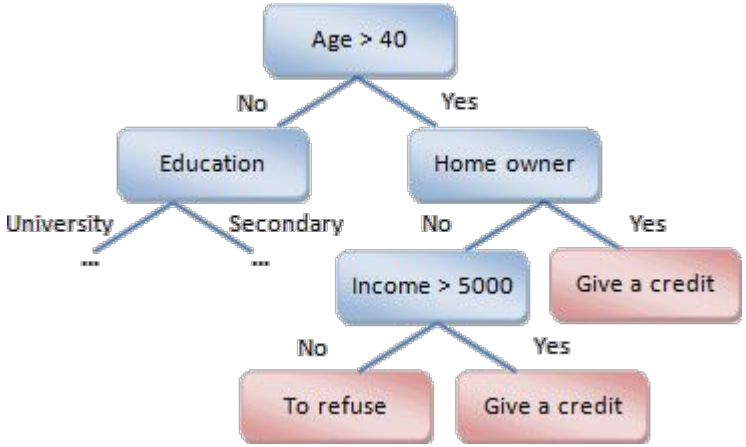
KNN Cons

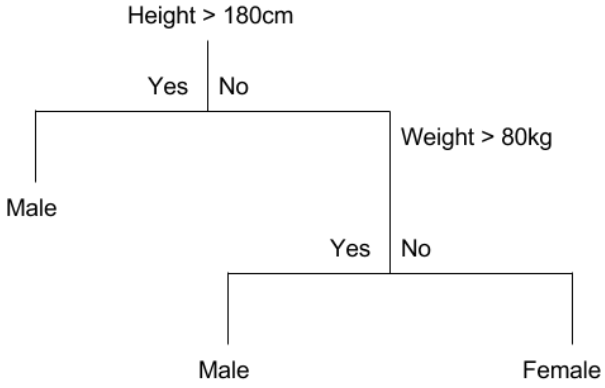
- Large search problem to find nearest neighbours
- Storage of data
- Must know we have a meaningful distance function

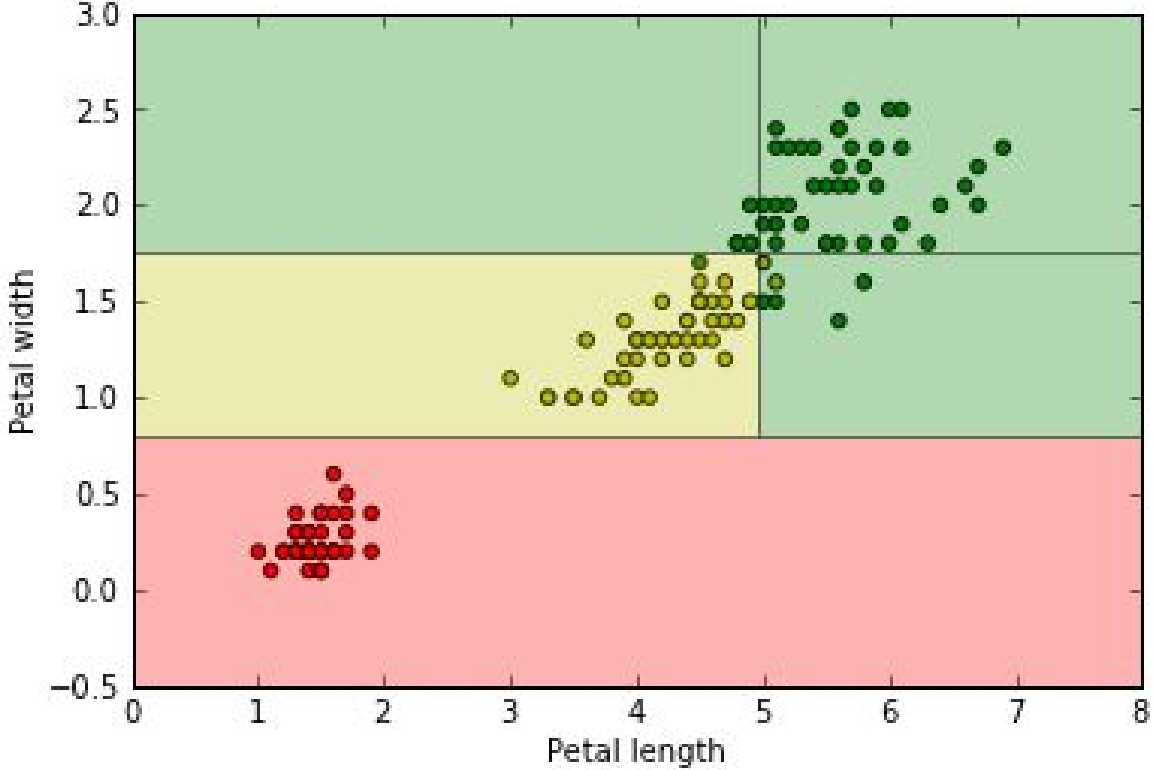
Decision Tree

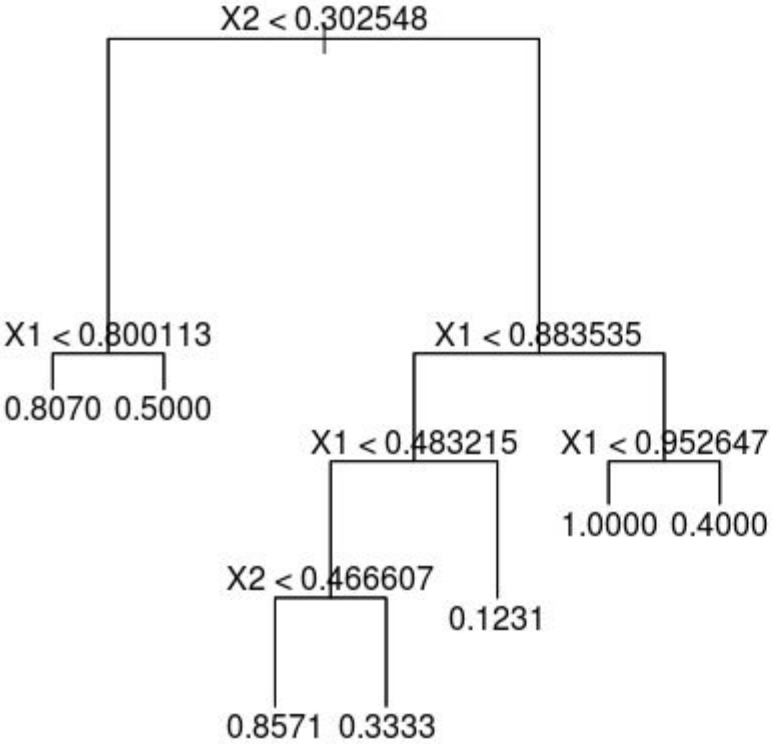
How it works?

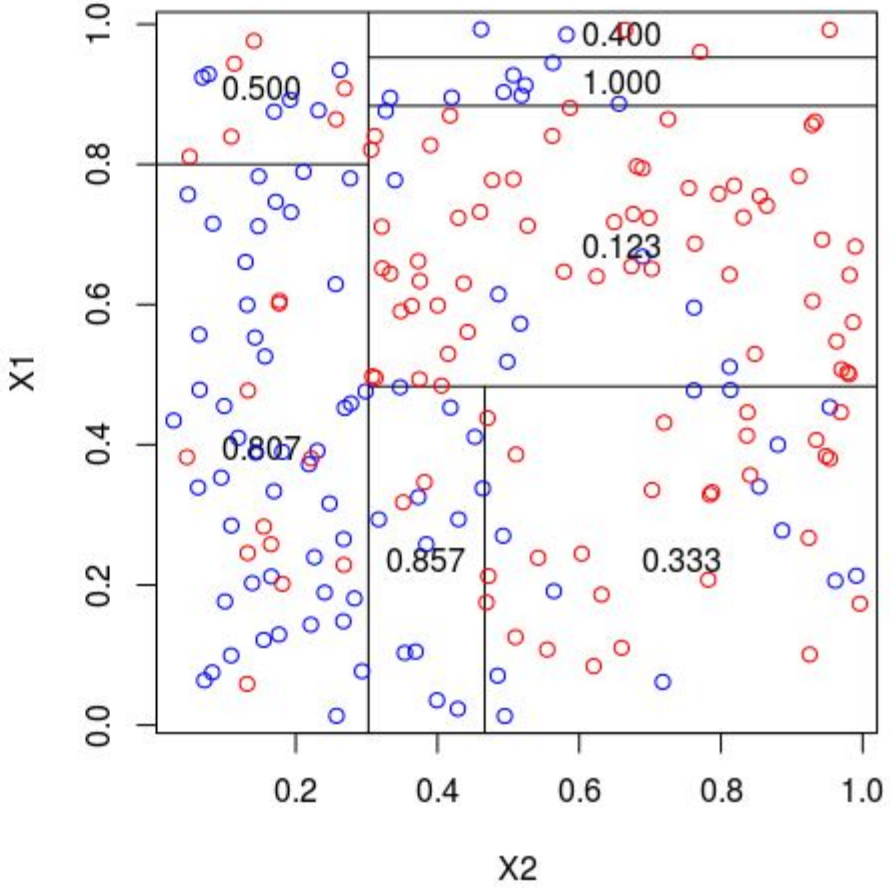
- Decision tree uses a tree structure to represent number of possible decision paths and an outcome for each path
- Decision Trees can be applied to both classification & regression problems
- Classification:
  Entropy,Information Gain
- Regression:
  Mean Square Error






3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
Petal width
0
1
2
3
4
5
6
7
8
Petal length


X2 < 0.302548
X1 < 0.300113
0.8070 0.5000
X1 < 0.883535
X1 < 0.483215
X1 < 0.952647
1.0000 0.4000
X2 < 0.466607
0.1231
0.8571 0.3333


0.500
0.400
1.000
0.123
0.807
0.857
0.333
X1
X2
0.0
0.2
0.4
0.6
0.8
1.0

---

http://www.saedsayad.com/decision_tree.htm

Recursive *divide-and-conquer* fashion

- First: select attribute for root node

Create branch for each possible attribute value

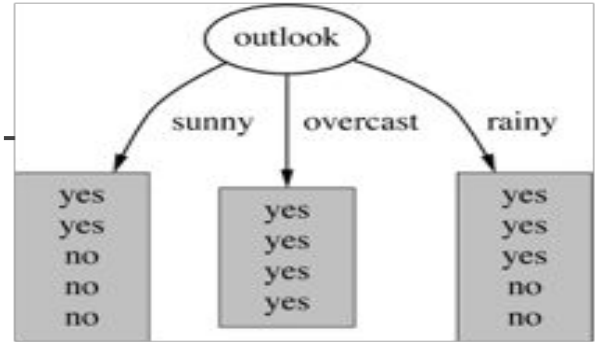
- Then: split instances into subsets

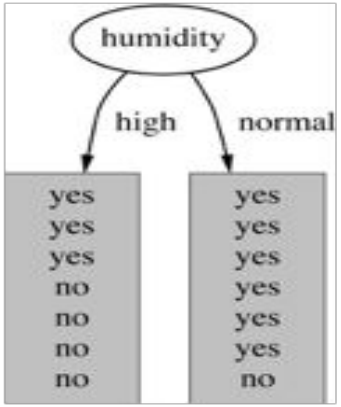
One for each branch extending from the node

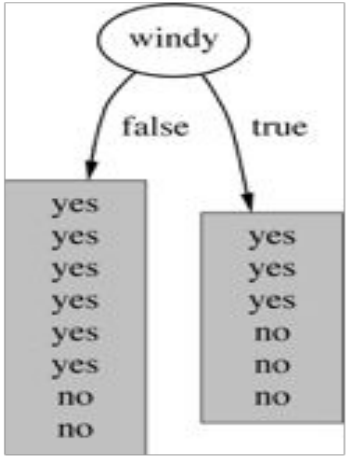
- Finally: repeat recursively for each branch, using only instances that reach the branch
- Stop if all instances have the same class

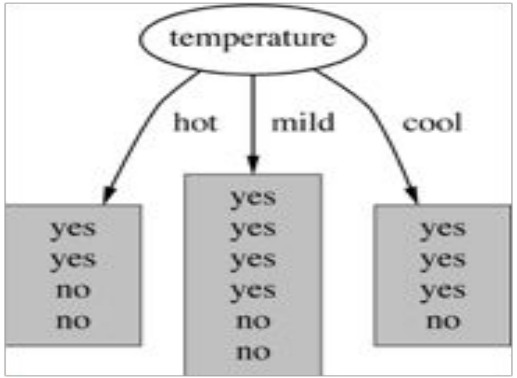
```
OUTLOOK      TEMP     HUMIDITY    WINDY  PLAY
-------      ----     --------    -----  ----
Sunny        Hot      High        False    No
Sunny        Hot      High        True     No
Overcast     Hot      High        False    Yes
Rainy        Mild     High        False    Yes
Rainy        Cool     Normal      False    Yes
Rainy        Cool     Normal      True     No
Overcast     Cool     Normal      True     Yes
Sunny        Mild     High        False    No
Sunny        Cool     Normal      False    Yes
Rainy        Mild     Normal      False    Yes
Sunny        Mild     Normal      True     Yes
Overcast     Mild     High        True     Yes
Overcast     Hot      Normal      False    Yes
Rainy        Mild     High        True     No%%
```

Which attribute to select?

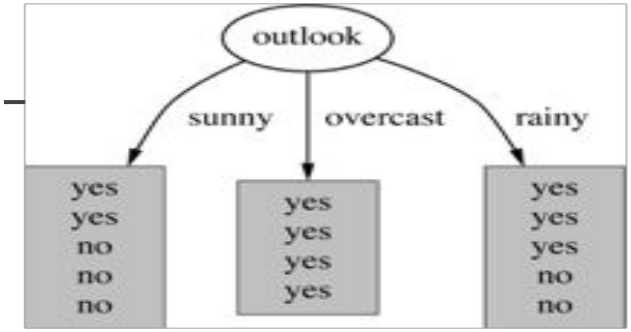


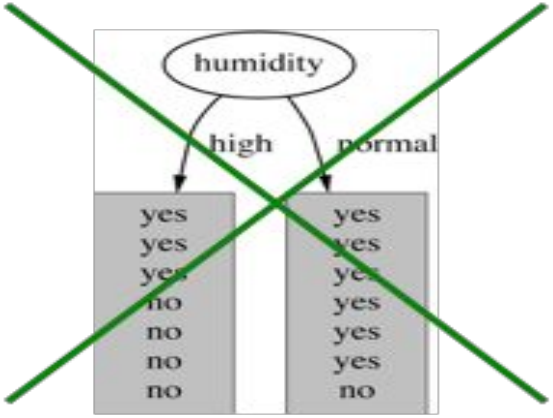


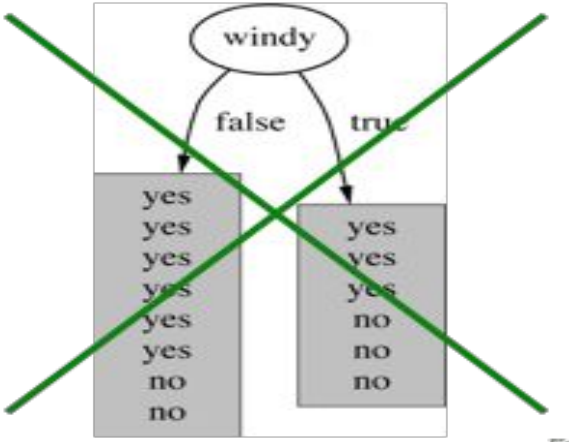


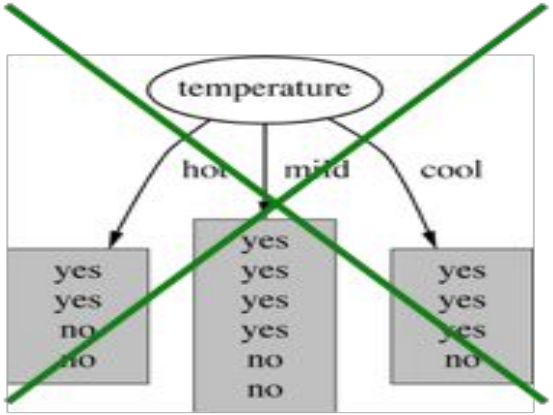


Which attribute to select?

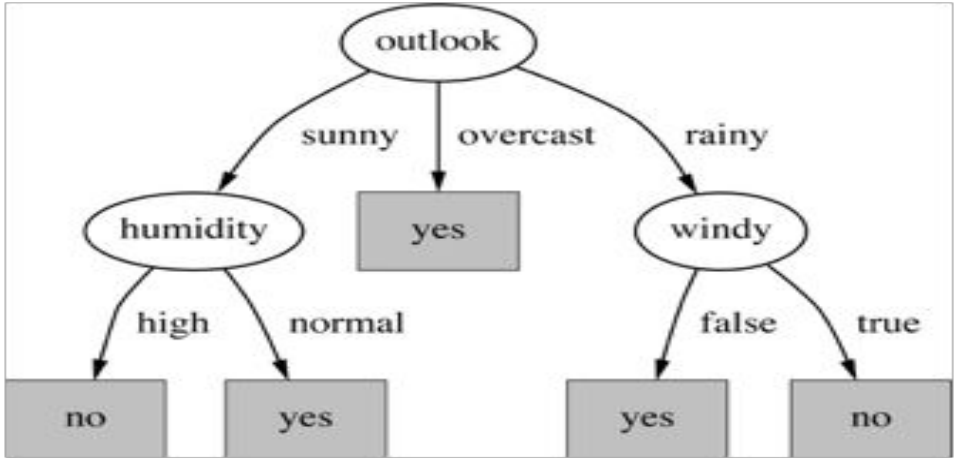








Final decision tree



⇒ Splitting stops when data can't be split any further

Computing Information Gain

- Information gain: information before splitting – information after splitting

$$\text{gain}(\textit{Outlook}) = \text{info}([9,5]) - \text{info}([2,3],[4,0],[3,2])$$
$$= 0.940 - 0.693$$
$$= 0.247 \text{ bits}$$

- Information gain for attributes from weather data:

gain(*Outlook*)	= 0.247 bits
gain(*Temperature*)	= 0.029 bits
gain(*Humidity*)	= 0.152 bits
gain(*Windy*)	= 0.048 bits

Points to consider

- Information gain:
- Increases with the average purity of the subsets
- **Entropy**:
- to calculate the homogeneity of a sample.
- Pruning:
- Prevent overfitting to noise in the data

Decision Tree Pros

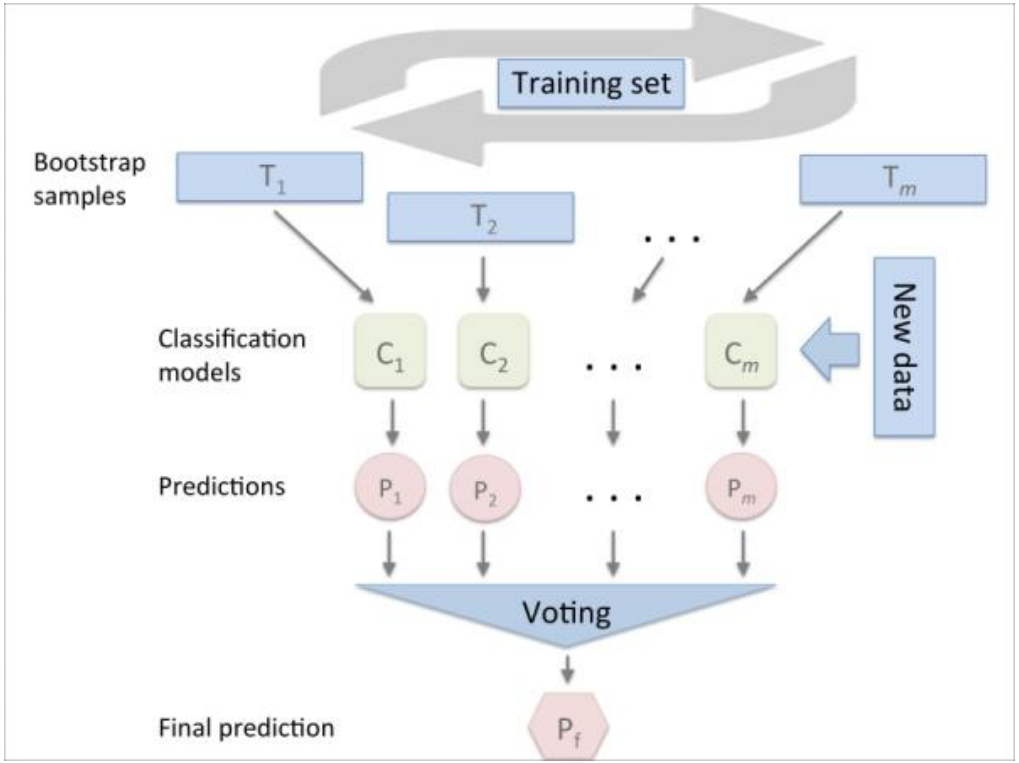
- Inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to **interpret** for small-sized trees
- Accuracy is comparable to other classification techniques for many simple data sets

Decision Tree Cons

- Inadequacy in applying regression and predicting continuous values
- Unsuitability for estimation of tasks to predict values of a continuous attribute
- Limited to one output per attribute, and inability to represent tests that refer to two or more different objects
- Possibility of overfitting
- High variance and bias problem to overcome we can go for bagging technique

Ensemble

Machine learning paradigm which combine weak learners to become a strong learner



Random Forest

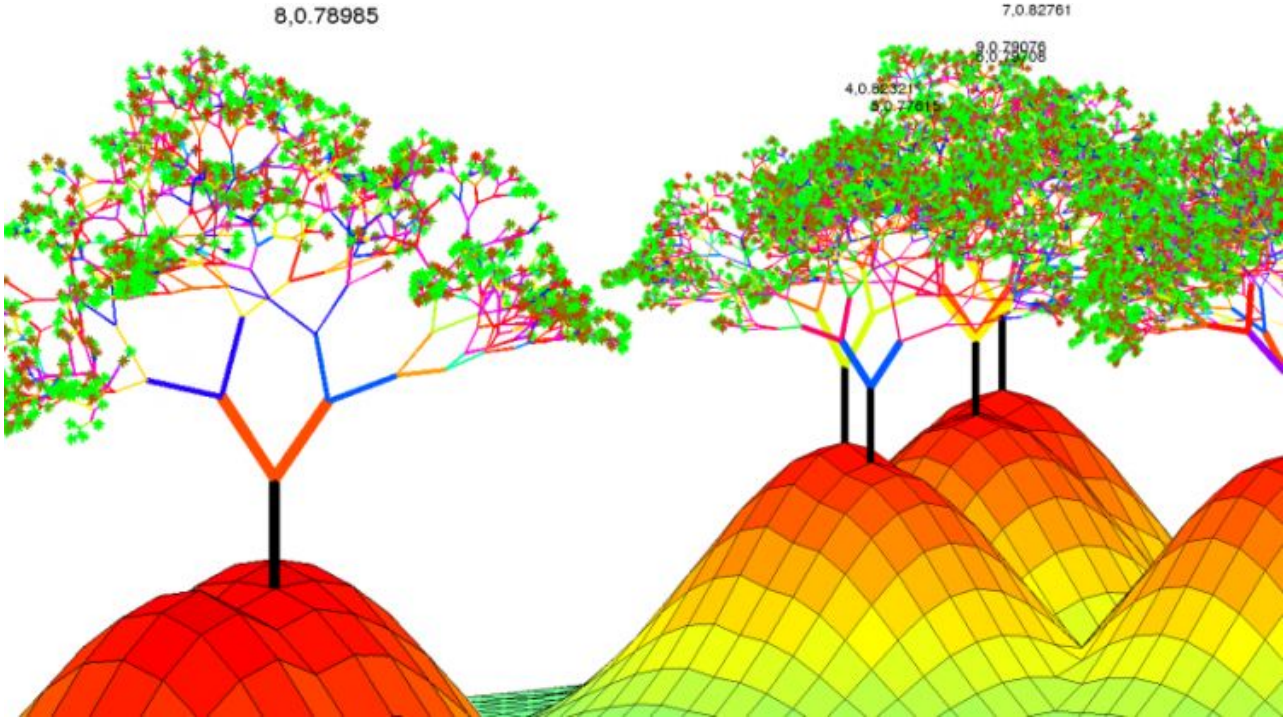
Most used algorithm- Bagging Technique

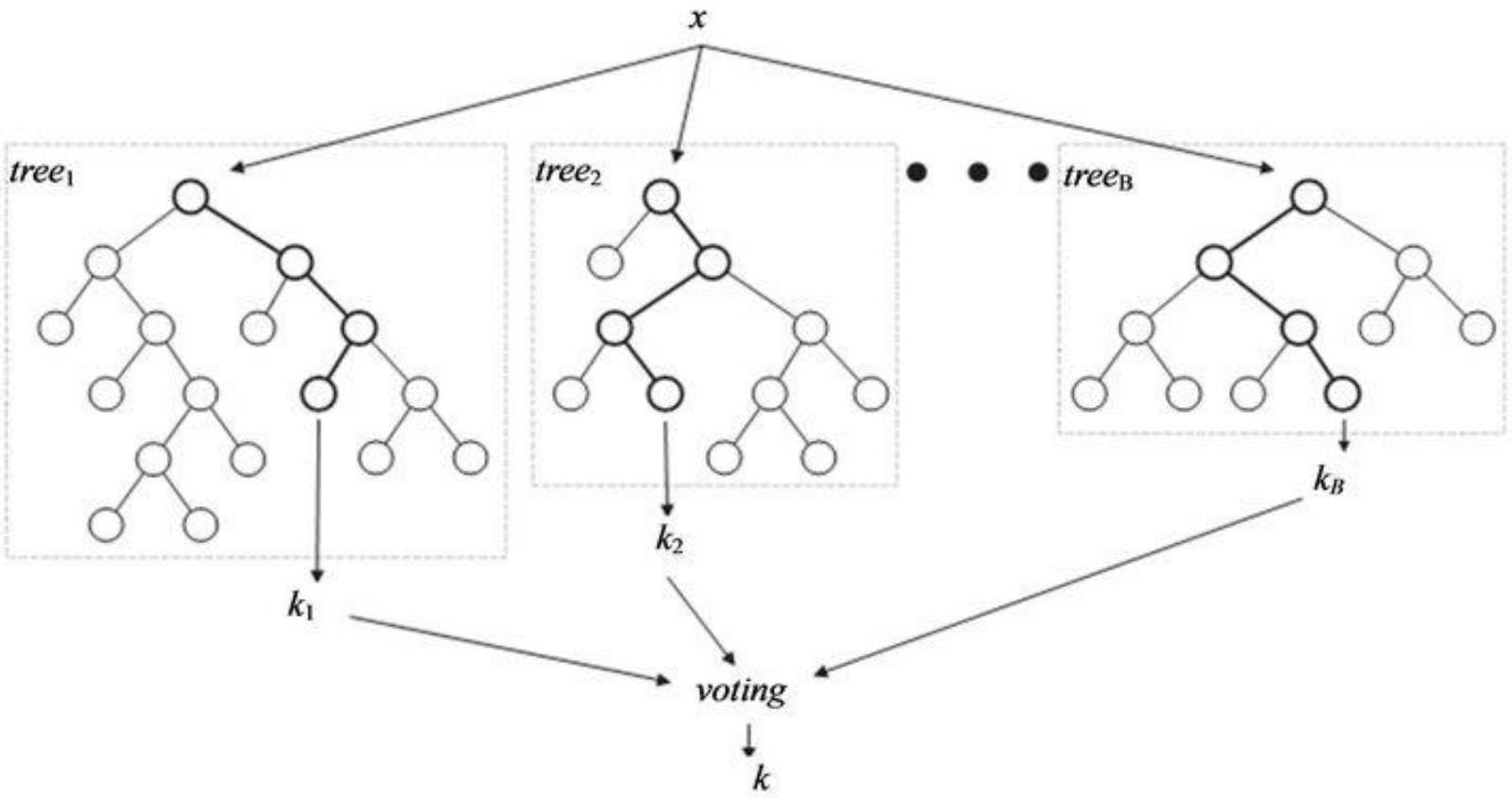
Bagging

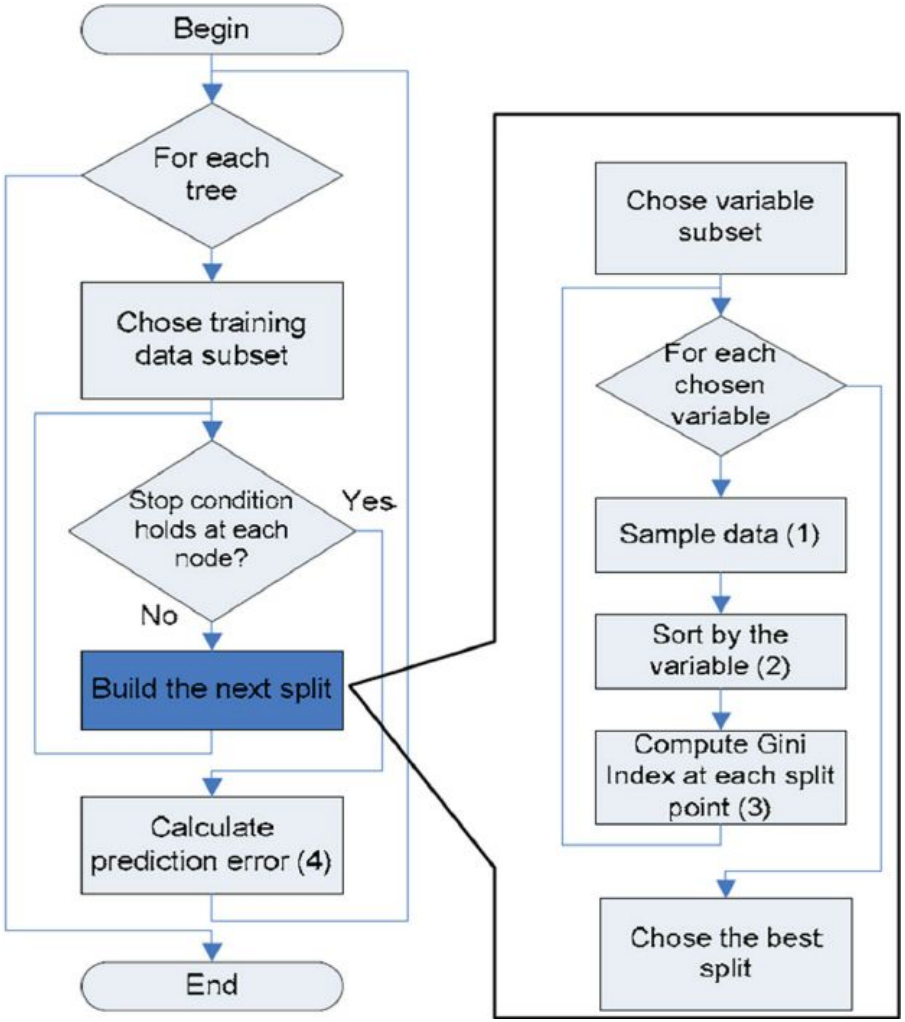
Bootstrap **agg**regat**ing**, also called **bagging**, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression.

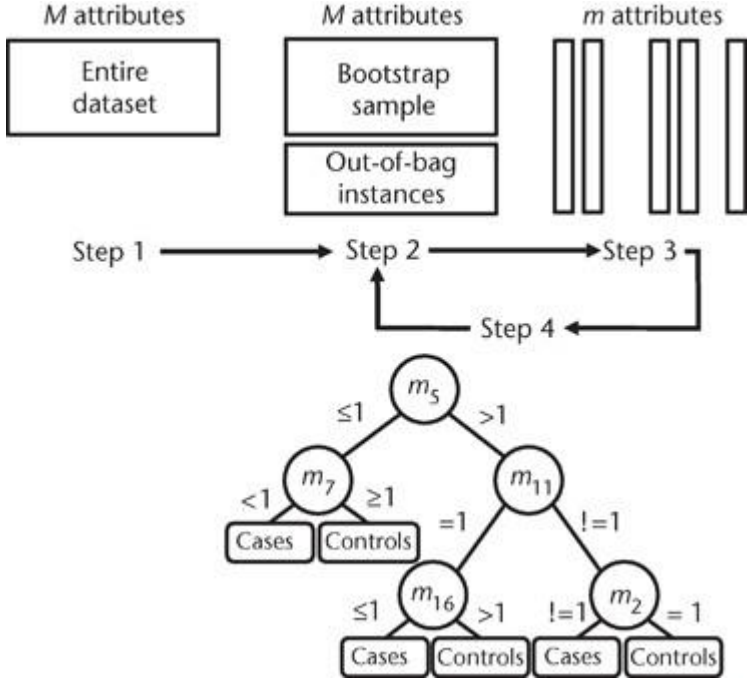
It also reduces variance and helps to avoid overfitting.

What insight can u get?




x
tree1
tree2
treeB
k1
k2
kB
voting
k


Begin
For each tree
Chose training data subset
Stop condition holds at each node?
Yes
No
Build the next split
Calculate prediction error (4)
End
Chose variable subset
For each chosen variable
Sample data (1)
Sort by the variable (2)
Compute Gini Index at each split point (3)
Chose the best split


M attributes
Entire dataset
M attributes
Bootstrap sample
Out-of-bag instances
m attributes
Step 1
Step 2
Step 3
Step 4
m_5
≤1
>1
m_7
m_{11}
<1
≥1
=1
!=1
Cases
Controls
m_{16}
m_2
≤1
>1
!=1
= 1
Cases
Controls
Cases
Controls

Points to remember

Feature selection

- For classification a good default is: m = sqrt(p)
- For regression a good default is: m = p/3

Estimated Performance

- For each bootstrap sample taken from the training data, there will be samples left behind that were not included. These samples are called **Out-Of-Bag samples** or OOB.
- The performance of each model on its left out samples when averaged can provide an estimated accuracy of the bagged models. This estimated performance is often called the OOB estimate of performance.
- These performance measures are reliable test error estimate and correlate well with cross validation estimates.

Variable importance

- It will find most important variable for feature selection based on gini index

Random forest cons

It is one of the most accurate learning algorithms available.

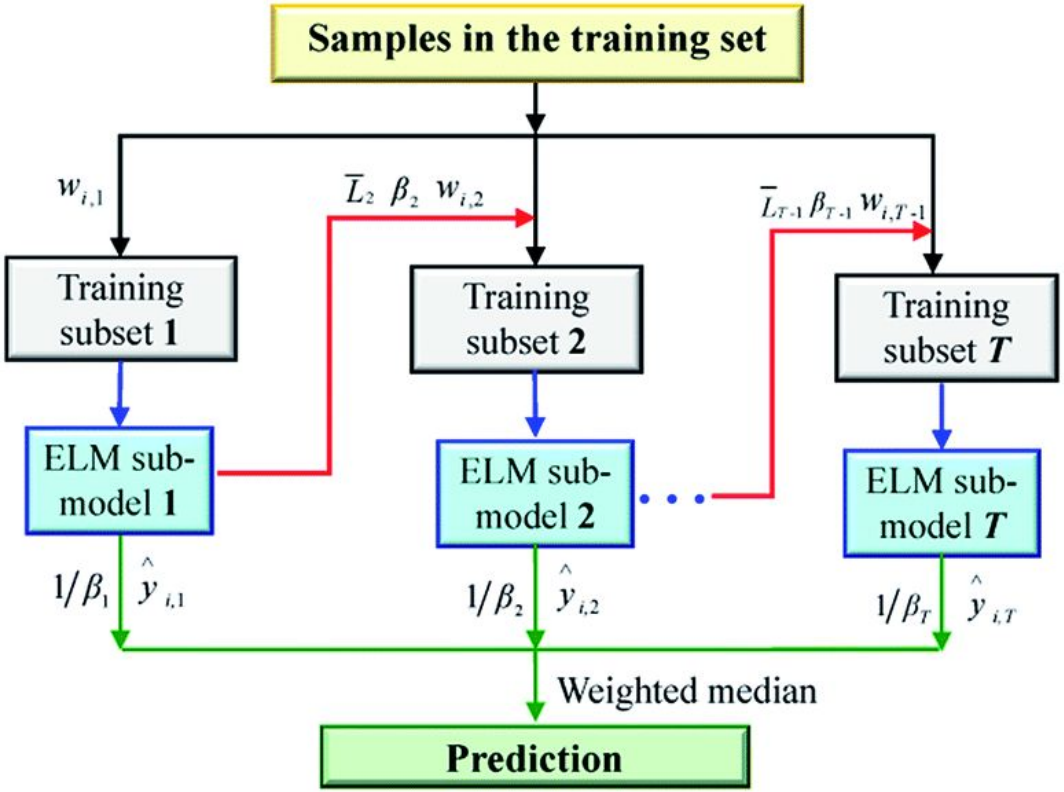
For many data sets, it produces a highly accurate classifier.

It runs efficiently on large databases.

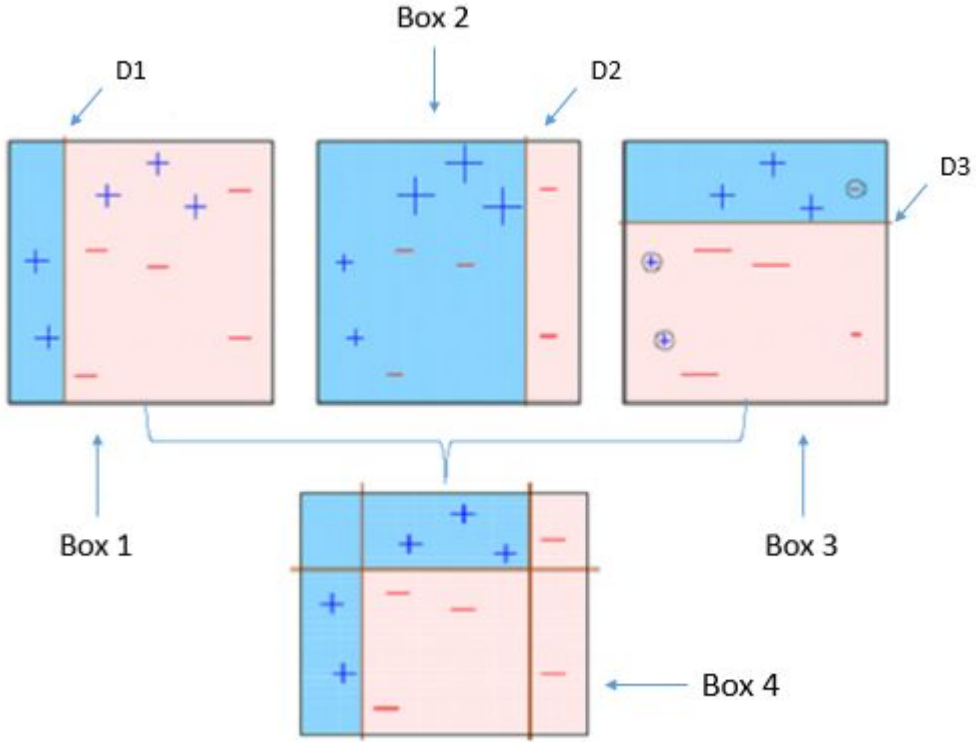
Boosting

Competition winning algorithm

Boosting


Samples in the training set
$w_{i,1}$
$\overline{L}_2\ \beta_2\ w_{i,2}$
$\overline{L}_{T-1}\ \beta_{T-1}\ w_{i,T-1}$
Training subset 1
Training subset 2
Training subset T
ELM sub-model 1
ELM sub-model 2
ELM sub-model T
$1/\beta_1$ $\hat{y}_{i,1}$
$1/\beta_2$ $\hat{y}_{i,2}$
$1/\beta_T$ $\hat{y}_{i,T}$
Weighted median
Prediction

AdaBoost



Gradient Boosting

```
Y = M(x) + error

error = G(x) + error2

error2 = H(x) + error3

Y = M(x) + G(x) + H(x) + error3

Y = alpha * M(x) + beta * G(x) + gamma * H(x) + error4
```

Boosting pros

Boosting (machine learning)
Boosting is a machine learning ensemble meta-algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms which convert weak learners to strong ones.

Boosting cons

Time and computation expensive.

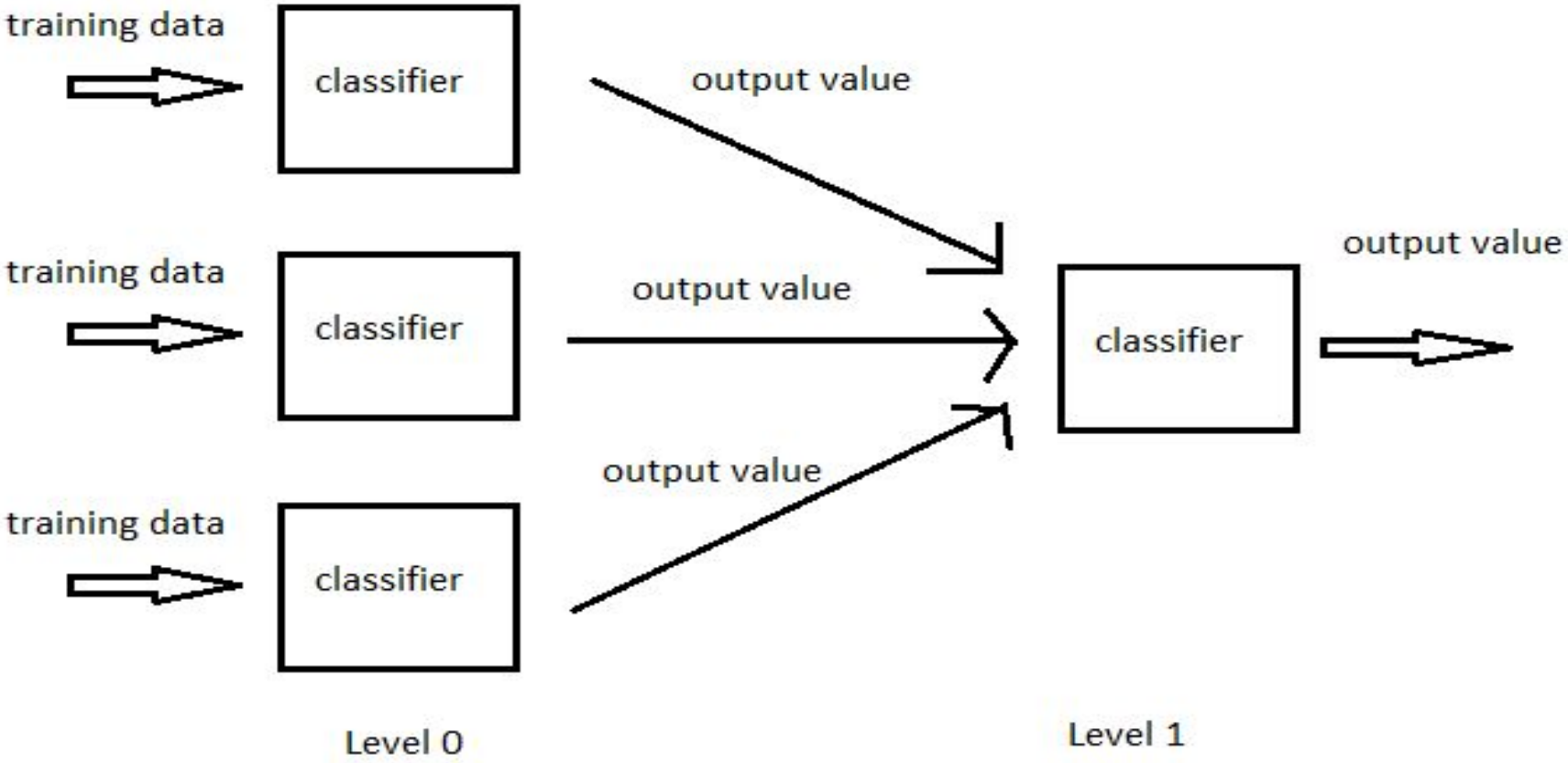
Hard to implement in real time platform.

Complexity of the classification increases.

Stacking

Competition winning algorithm

Concept Diagram of Stacking



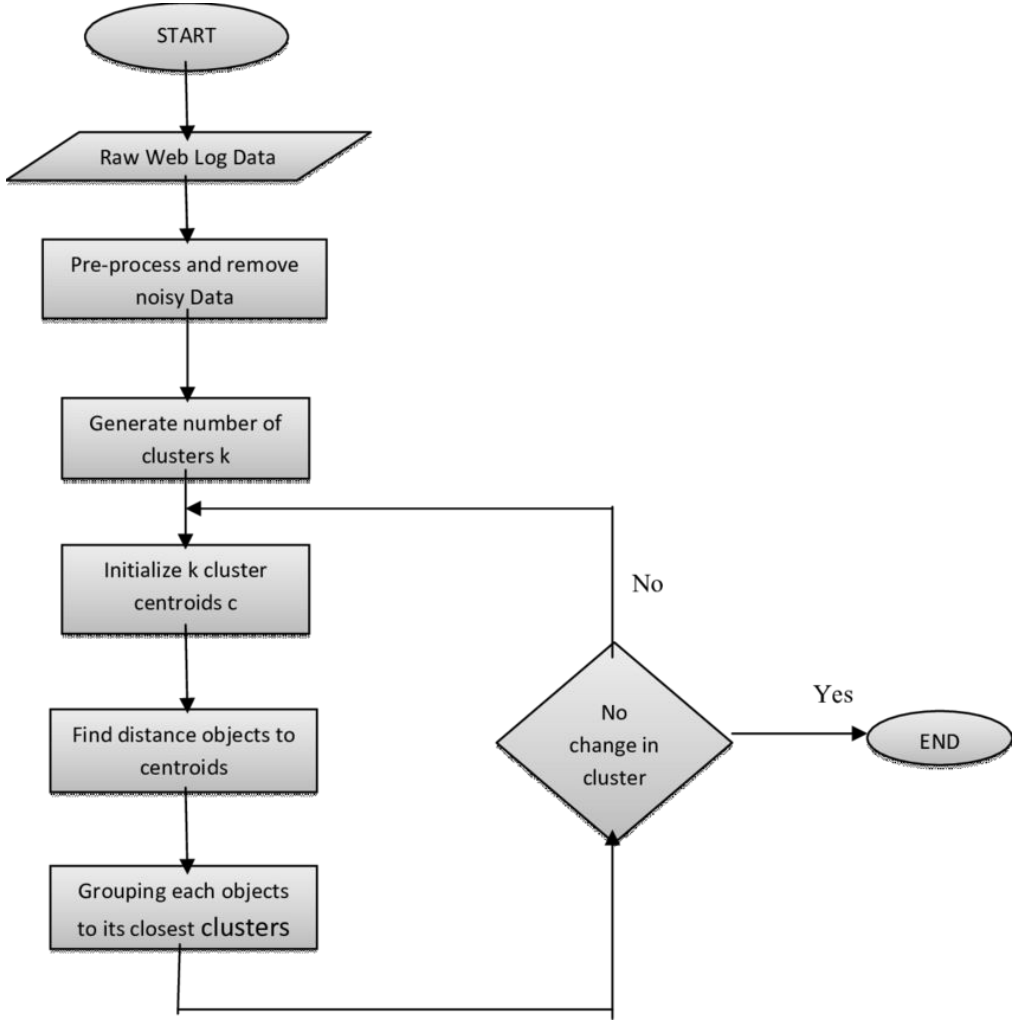
K-Means Clustering

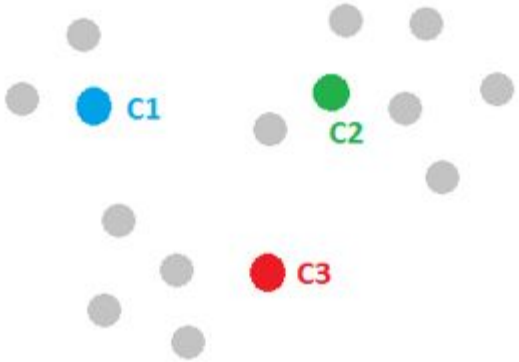
Step one: Initialize cluster centers

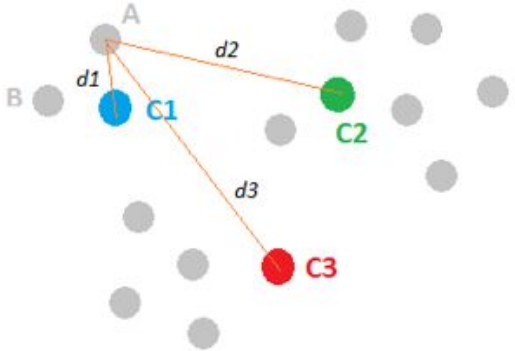
Step two: Assign observations to the closest cluster center

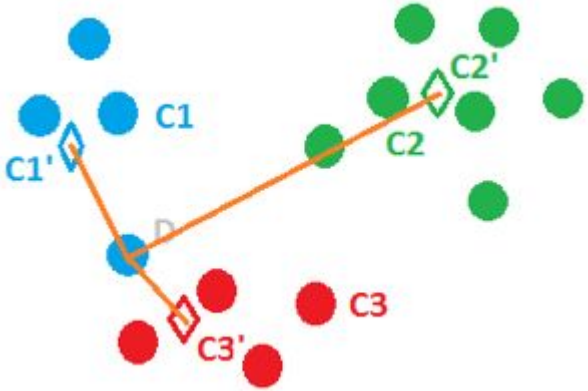
Step three: Revise cluster centers as mean of assigned observations

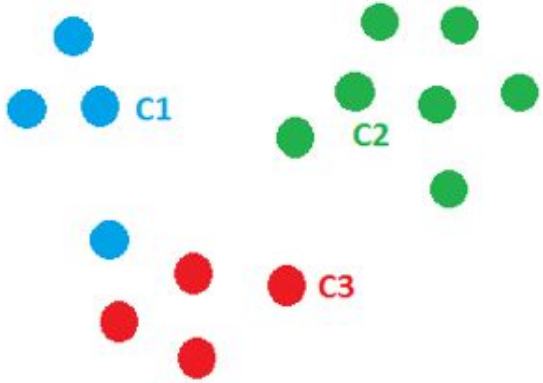
Step four: Repeat step 2 and step 3 until convergence

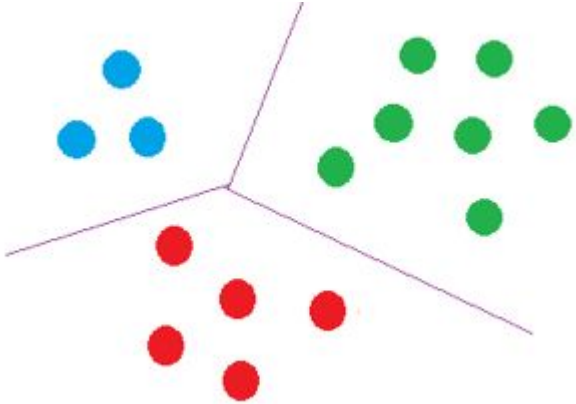



C1
C2
C3

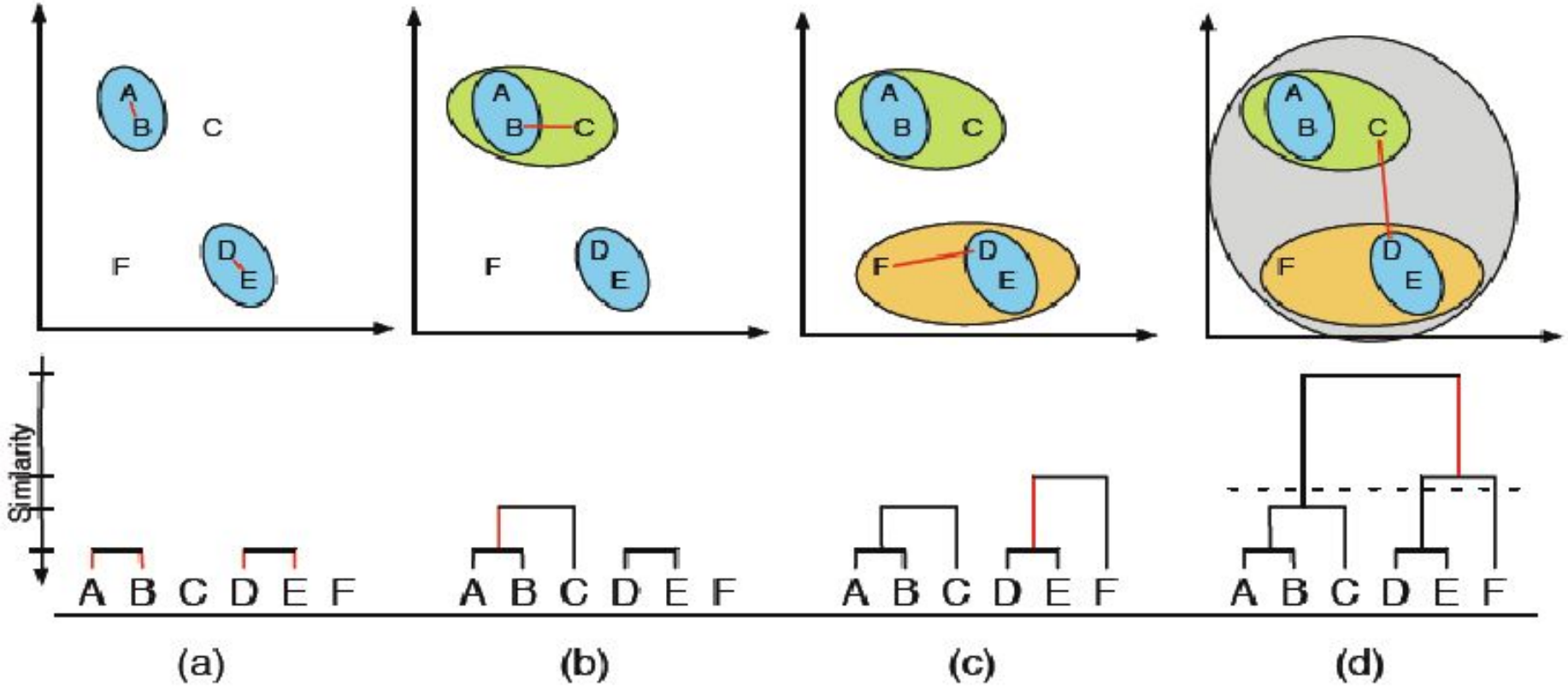

A
B
d1
d2
d3
C1
C2
C3


C1
C1'
C2'
C2
D
C3
C3'

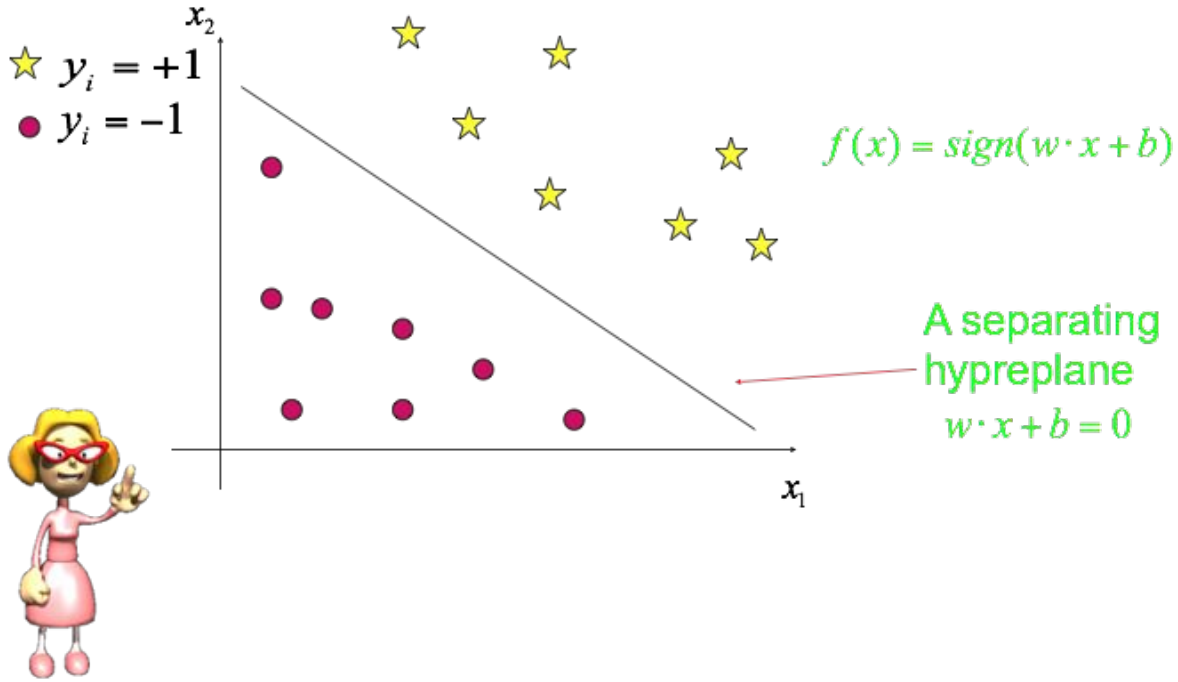

C1
C2
C3



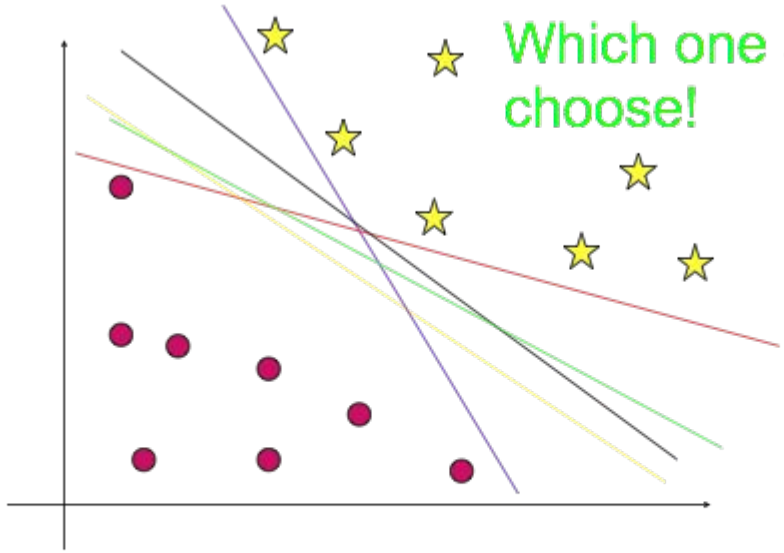
Example: Hierarchical Agglomerative Clustering



Support Vector Machine


$y_i = +1$
$y_i = -1$
x_2
x_1
$f(x) = sign(w \cdot x + b)$
A separating hypreplane
$w \cdot x + b = 0$

$y_i = +1$
$y_i = -1$



Which one should we choose!

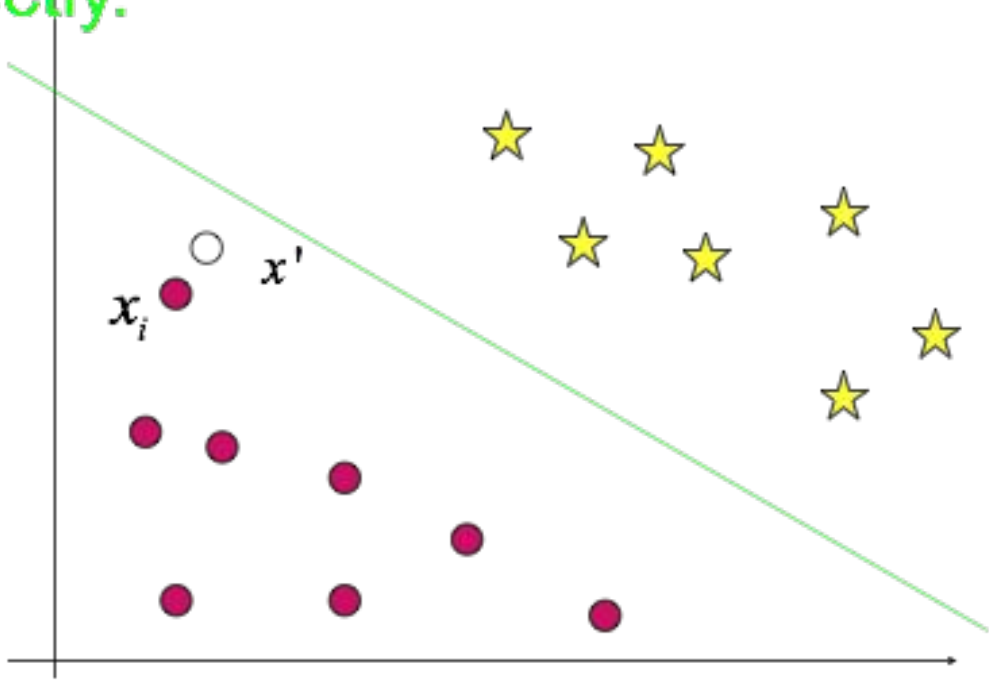
Yes, There are many possible separating hyperplanes

It could be this one or this or this or maybe….!

-Hyperplane should be as far as possible from any sample point.

-This way a new data that is close to the old samples will be classified correctly.

Good generalization!



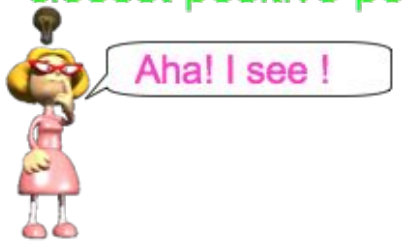
Choosing a separating hyperplane.
The SVM approach: Linear separable case

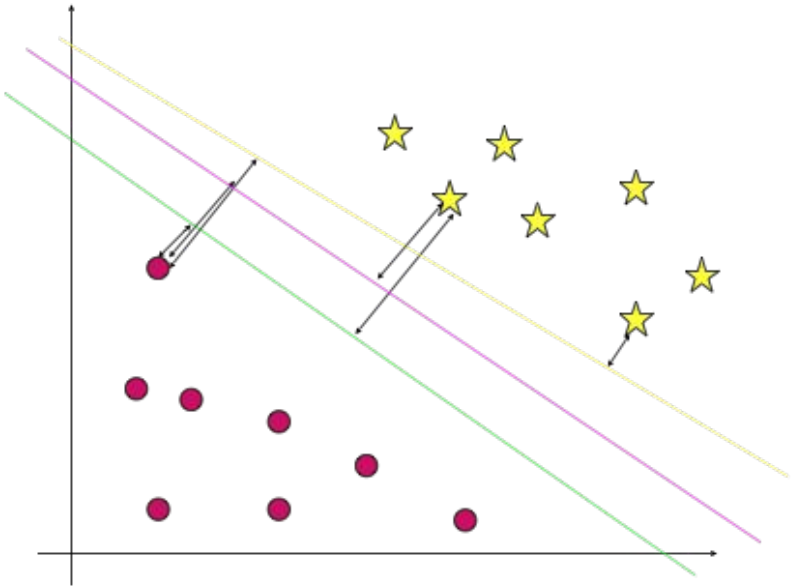
-The SVM idea is to maximize the distance between The hyperplane and the closest sample point.

In the optimal hyper-plane:

The distance to the closest negative point =

The distance to the closest positive point.





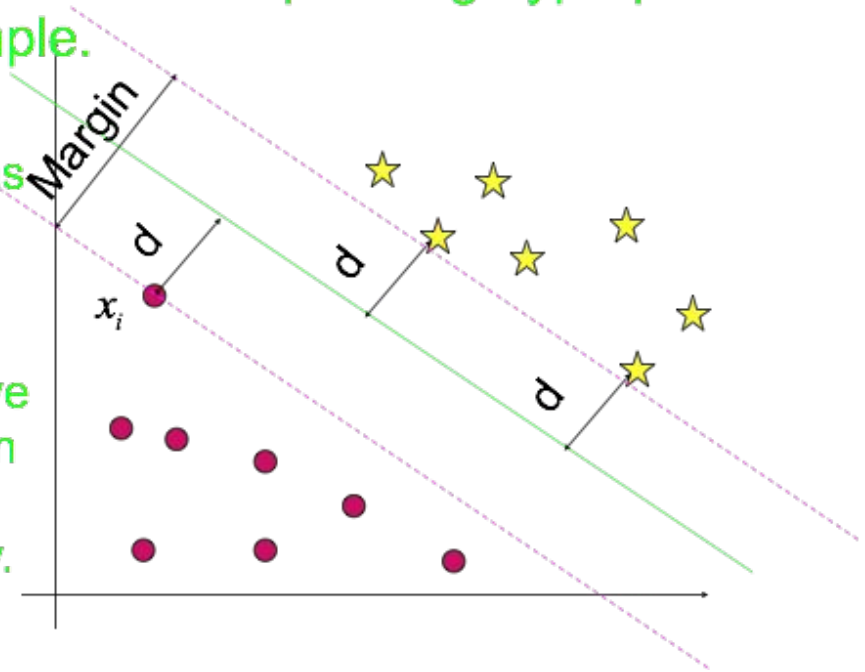
Choosing a separating hyperplane.
The SVM approach: Linear separable case

SVM's goal is to maximize the Margin which is twice the distance "d" between the separating hyperplane and the closest sample.

Why it is the best?

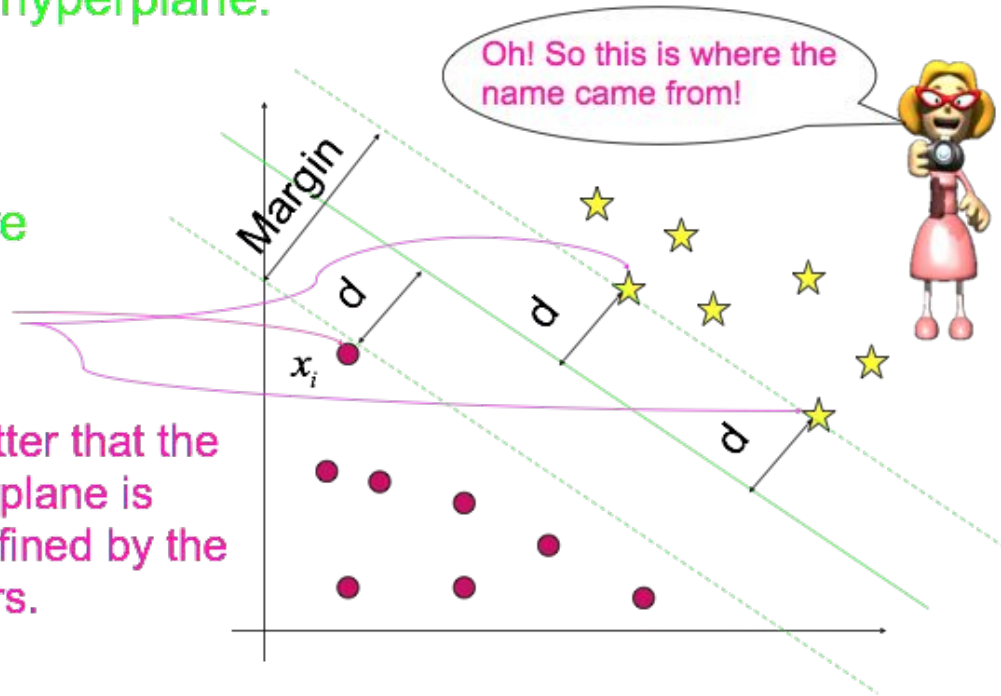
-Robust to outliners as we saw and thus strong generalization ability.

-It proved itself to have better performance on test data in both practice and in theory.



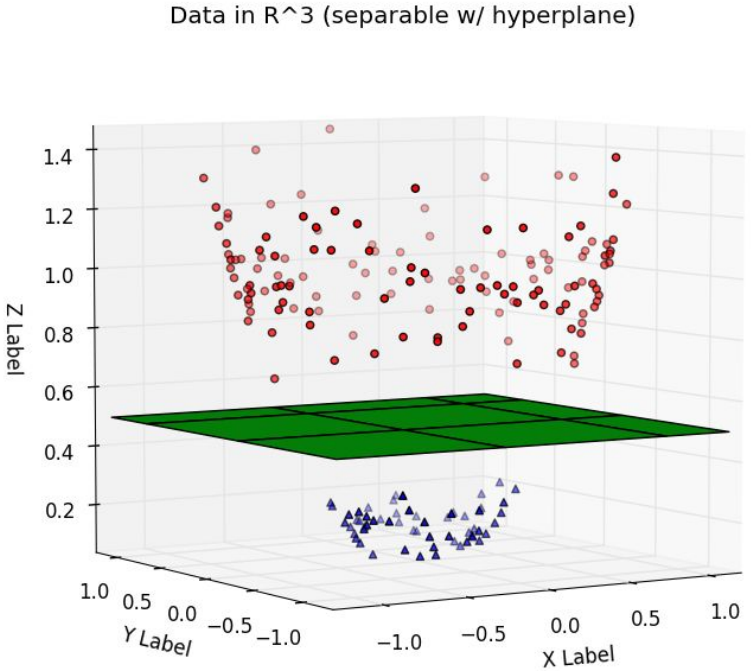
Choosing a separating hyperplane.
The SVM approach: Linear separable case

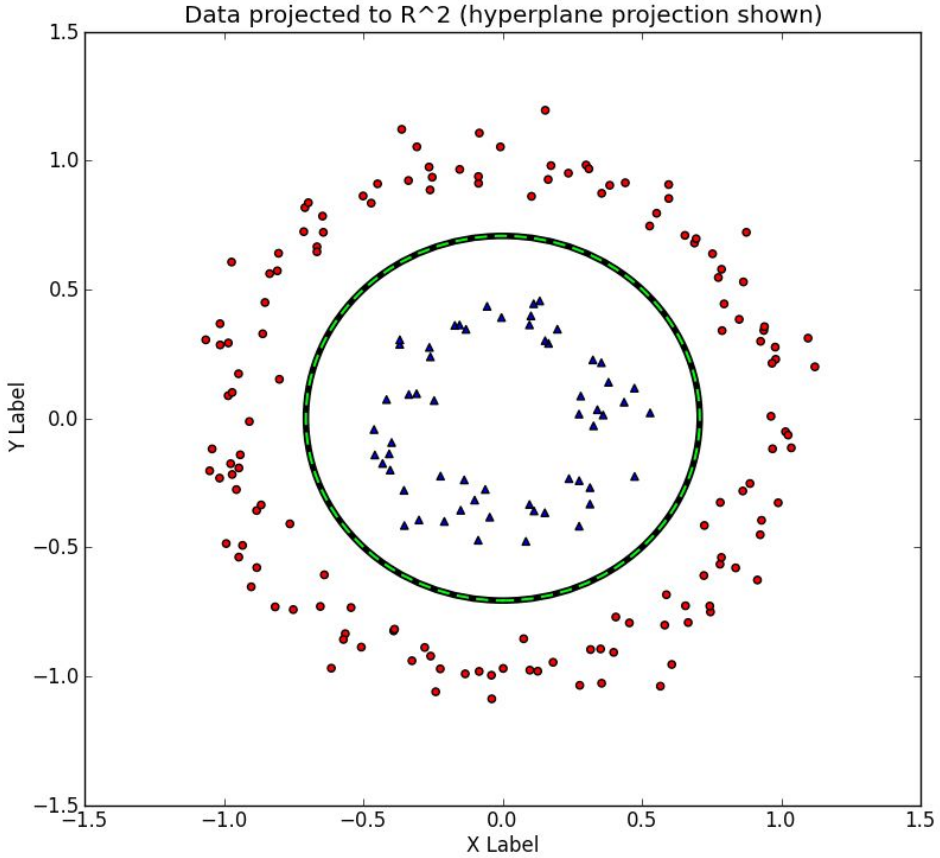
Support vectors are the samples closest to the separating hyperplane.



These are Support Vectors

We will see latter that the Optimal hyperplane is completely defined by the support vectors.


Data in R^3 (separable w/ hyperplane)
Z Label
1.4
1.2
1.0
0.8
0.6
0.4
0.2
1.0
0.5
0.0
−0.5
−1.0
Y Label
−1.0
−0.5
0.0
0.5
1.0
X Label


Data projected to R^2 (hyperplane projection shown)
Y Label
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
X Label

SVM : Linear separable case.

The optimization problem:

-Our optimization problem so far:

I do remember the Lagrange Multipliers from Calculus!

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

We will solve this problem by introducing Lagrange multipliers α_i associated with the constrains:

$$\text{minimize} \quad L_p(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n}\alpha_i(y_i(x_i \cdot w + b) - 1)$$

$$s.t \quad \alpha_i \geq 0$$

SVM : Linear separable case.

The optimization problem cont':

So our primal optimization problem now:

$$\text{minimize } L_p(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n}\alpha_i(y_i(x_i \cdot w + b) - 1)$$

$$s.t \quad \alpha_i \geq 0$$

We star solving this problem:

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

SVM : Linear separable case.
Inroducing The Legrangin Dual Problem.

By substituting the above results in the primal problem and doing some math manipulation we get:

Lagrangian Dual Problem:

$$\text{maximaize } L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=0}^{n} \sum_{j=0}^{n} \alpha_i \alpha_j y_i y_j x_i^t x_j$$

$$s.t \;\; \alpha_i \geq 0 \;\; and \;\; \sum_{i=1}^{n} \alpha_i y_i = 0$$

$\alpha = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ are now our variables, one for each sample point x_i.

Summary

Supervised Learning

Linear Regression

Logistic Regression

SVM

Ensemble

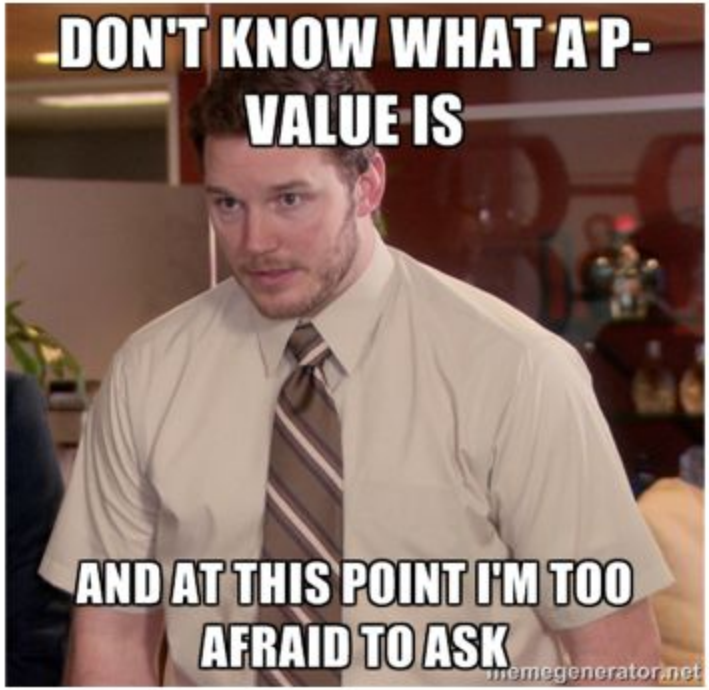
KNN

Unsupervised

K-means

Hierarchical

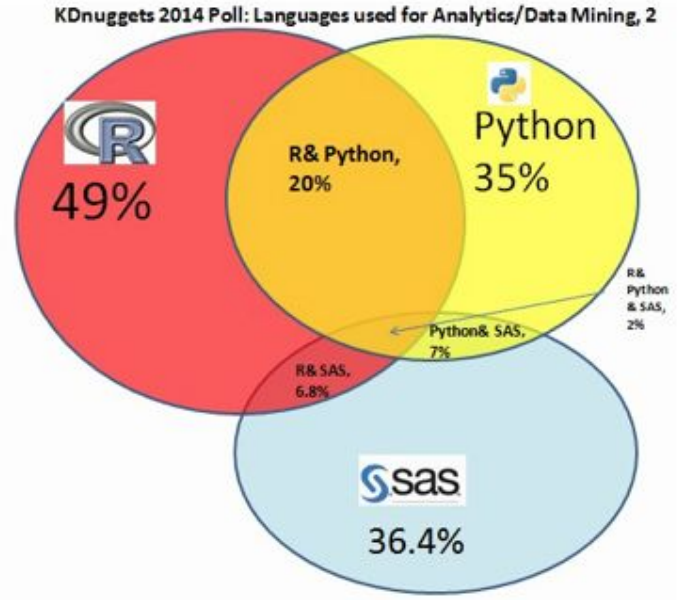
Revise Statistics and Probability



Learn algorithm how it works rather concentrating on output

HOW IT WORKS

Learn and practice a programming language



Big data and Data science are happily married

Practice all algorithm with each use case

Make use of Kaggle and Analytics vidhya

Contribute your work to Opensource

Write a blog


blog
blogs
content
Web
online
posts
blogging
interactive
individual

Make your resume attractive

MOHAMED NOORDEEN A
Data Scientist
Tiger Analytics
nursnaaz@gmail.com
https://www.linkedin.com/in/nursnaaz/
https://github.com/nursnaaz/
www.technaaz.com
(+91) 9789-830021

PROFILE SUMMARY

- A Data Scientist with 6 years of experiences in advance analytics, Machine Learning, Predictive Modelling, Data Mining, Text Analytics and Statistical Data analysis.

MOHAMED NOORDEEN A
Senior Software Engineer |Data Science|
TIGER ANALYTICS

CONTACT

PHONE
+91 97888 30021

E-MAIL
nursnaaz@gmail.com

LINKEDIN
www.linkedin.com/in/nursnaaz

GITHUB
www.github.com/nursnaaz

WEBSITE
www.technaaz.com

SKILLS

SQL JAVASCRIPT C++ JAVA PYSPARK DATA SCIENCE PYTHON

DATA ANALYTICS PROFESSIONAL

5 years of proven engineering abilities with acquired skills of mining hidden gems located within the large data sets of structured, semi-structured and unstructured data. The profound expertise will be used for improving organizational productivity and my own progress.

SAMPLE WORK IN ANALYTICS

Developed a forecast on industry shipment in the US home appliances for next two quarters based on the past data. The forecast was done using univariate time series models like seasonal regression, SMA, WMA, and EMA. Later the model was the best fit with advanced time series model like ARIMA.

Image recognition using deep learning model. I even participated similar problems in online competitions like Analytics Vidhya and Kaggle and placed in top 10 leaderboard score.

ACADEMIC

Matriculation 85% 2005

2007 85% High Secondary

B.Tech 81% 2011

CORE COMPETENCIES

- CLUSTERING & CLASSIFICATION
- REGRESSION & OPTIMIZATION
- TIME SERIES FORECASTING
- DEEP LEARNING
- STATISTICS ANALYSIS
- TEXT MINING & NLP
- BIG DATA TOOLS
- STORY TELLING

CERTIFICATION

2017 Certified Program in Big Data Analytics & Optimization at INSOFE
2016 Certified Big Data Hadoop developer by Collabera TACT
2013 Oracle Certified Professional, Java SE 6 Programmer
2012 Oracle Database SQL Certified Expert

Contact

Mohamed Noordeen
nursnaaz@gmail.com
www.technaaz.com