URBAN EDGE COLORFUL YOU

7/12/24

Introduction

Definition: ML is a set of methods that can automatically detect patterns in data and then use those uncovered patterns to predict the future value or output.

In other words, it is a computer program which has to learn from the experience 'E' wrt a specific class of task 'T' and the performance measure 'P'. If the performance at task 'T' as measured by 'P' improves with experience 'E'.

example: Handwritten character recognization as learning Problem

Task 'T': recognizing and classifying handwritten characters within images

'E': database of handwritten characters with lables.

'P': percentage of characters correctly classified.

Machine Learning Framework

$y = f(x)$

y: output x: feature representation

f() is prediction function

- Training: Given training set, estimate prediction function f() by minimizing prediction error.
- Testing: Apply f() to unknown test sample x and predict y.

for linear model: $y = f(w, x)$.

Phases for solving AI/ML problem:

- define your task
- collect data
- preprocessing data
- dimensionality reduction / feature scaling
- choose ML algorithm
- experimental design
- test & validate
- Run system

Machine Learning Tools

Data Processing	Common Data Formats
scikit learn, R	CSV
Spark ML lib	HDFS
Tensorflow	Json

Dataset & Subsets

Training set:

- It is used for learning parameters of model
- It has an optimistic bias
- It is usually the biggest one of three sets & used to build the model

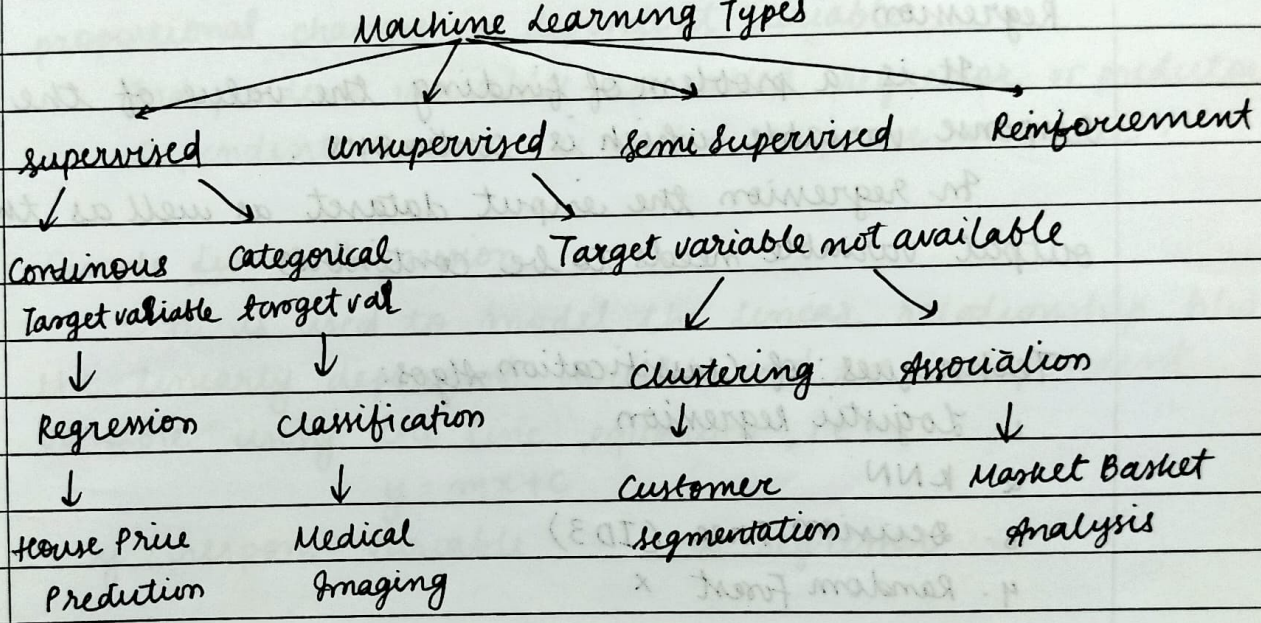
Test set:

- used to get a final, unbiased estimate of how well learning method works.
- we expect it to be worse than training set
- It is not biased
- no used to learn parameters of model.

URBAN EDGE COLORFUL YOU

TYPES OF ML:-

- Supervised learning
- Unsupervised learning
- Semi supervised
- Reinforcement learning



Supervised Learning

It is also known as predictive learning

It is provided with training data with target output.

Steps :-

1. Train the model
2. Test the model

Types of Supervised Learning

- Classification
- Regression

* Classification : It is a problem of assigning a label or category or class to an unlabled test samples.

In classification always output variable will be categorical or descrete whereas input variable can be either continous or descrete
Ex:- Dell

Regression

It is a problem of finding the value of the response variable which is continous.

In regresion the input dataset as well as the output variable needs to be continous.

Techniques of Classification Algos:
1. Logistic Regression
2. KNN
3. Decision Tree (ID3)
4. Random Forest x
5. Naive Bayes Classifier
6. Support Vector Machine
7. Neural Networks

Regression Techniques:
1. Simple Linear Regression
2. Multiple linear Regression
3. Lasso Regression
4. Ridge Regression
5. Polynomial Regression

URBAN EDGE COLORFUL YOU

Linear Regression

It is a statistical method which is used to find out the relation between independent variable and a dependent variable. In other words, it needs to findout the changes in independent variable as associated with proportional changes in dependent variable.

Independent variables are called regressor or predictor
Dependent variable is called response variable.

Simple Linear Regression

- It is used to model the linear relationship b/w the linearly dependent variable and the independent variable using the line equation, i.e.

$$y = mx + c$$

y : response variable x: regressor

- It can be performed if there is a linear relationship could be possible b/w dependent & independent variable

- The degree of linear relationship can be found with the help of corelation.

- Simple linear regression means one independent & one dependent variable.

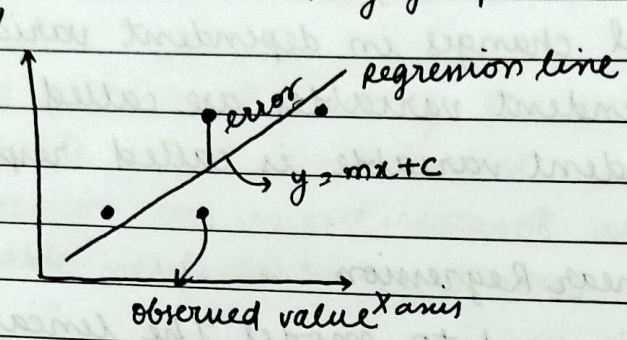
How to solve: ?

Methods

1: Least square Method

- used to find the eqn of best fitting curve or line to set of data points by minimizing the sum of squared differences b/w the observed values & predicted values.

- Independent variables are plotted as x coordinates and dependent variables as y coordinates
- The plot is called as scatter plot.
- It is the most common method used to fit a regression line in the x-y graph



$$m = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sum (x-\bar{x})^2} \qquad c = \bar{y} - m\bar{x}$$

$$m = \frac{n(\sum xy) - \sum x \cdot \sum y}{n\sum x^2 - (\sum x)^2}$$

Q. Find the line of best fit for data points using LSM:

$(x, y) = (1,3), (2,4), (4,8), (6,10), (8,15)$

x	y	xy	x^2	y'	$\lvert e \rvert^2$
1	3	3	1	2.6336	0.134
2	4	8	4	4.3106	0.0964
4	8	32	16	7.6646	0.1125
6	10	60	36	11.0186	1.0375
8	15	120	64	14.3726	0.3936
21	40	223	121		

$$m = \frac{5(223) - 21(40)}{5(121) - (21)^2} = \frac{275}{164} = 1.677$$

$c = 8 - 1.677(4.2)$

2) $c = 8 - 7.0434 = 0.9566$

eqn, $y = 1.677x + 0.9566$

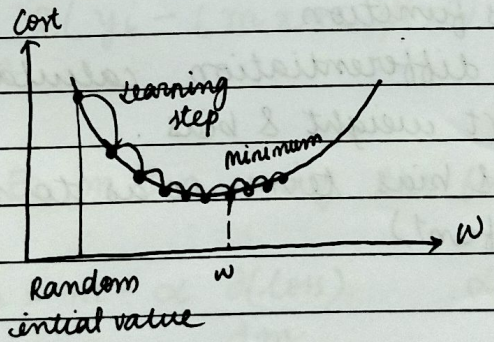
Assumption of the LSM

× i. to apply LSM to predict ×

1. linear relationship b/w the variables
2. the observations are independent of each other.
3. variance of residual is constant with a mean of 0
4. Error are distributed normally.

Method 2:

Gradient Descent Method



It is an optimization algo that can be used to find the global minima of a differentiable function

Cost function is MSE (Mean square Error) & formula

$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ (sum of sq. error)

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

URBAN EDGE
COLORFUL YOU

- Main goal of G.D is to find the 'm' and 'c' values that defines the relationship b/w variable x & y correctly. with the help of loss fun^t
- A loss fun^t is a function that signifies how much the predicted value is deviated from the actual values of dependent variable

note: find values of 'm' and 'c' such that it minimizes the loss function.

STEPS involved in linear regression with G.D

'm' 'c'

1. Initialize the weight & bias randomly or start with '0'.
2. Make prediction with the initialis weight & bias
3. Calculate loss function
4. With help of differentiation, calculate how loss fun^t changes wrt weight & bias.
5. Update weight & bias term, so as to minimize the loss fun^t (cost fun^t)

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

$$\frac{\partial \text{cost}}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum(y_i - \hat{y}_i)^2\right)$$

2) $$\frac{1}{n}\frac{\partial}{\partial m}\left(y_i - (mx+c)\right)^2$$

$$\cdot\, 2(y_i - (mx+c)).$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(y_i^2 + (mx+c)^2 - 2y_i(mx+c)\right)$$

$$= \frac{1}{n}\left[\frac{\partial}{\partial m}\left((mx+c)^2 - 2y_i(mx+c)\right)\right]$$

$$= \frac{1}{n}\left[2x(mx+c) - 2y_i x\right]$$

$$= \frac{2}{n}\left[m^2x^2 + mxc - y_i x\right] \quad \therefore \frac{2}{n}(y_i - (mx+c))(-x)$$

$$\frac{\partial}{\partial c}\left(\frac{1}{n}\sum (y_i - \hat{y}_i)^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left[(y_i - (mx+c)\right]^2$$

$$= \frac{1}{n}2(y_i - (mx+c))(-1) = \frac{-2}{n}(y_i - (mx+c))$$

UPDATE m & c using the following formula.

$$m = m - \alpha \frac{\partial(loss)}{\partial m}$$

α : learning rate

$0 < \alpha < 1$ (0 to 1).

$$c = c - \alpha \frac{\partial(cost)}{\partial c}$$

cost fun^t ~ loss fun^t.

$$m_1 = m - \alpha\left[\frac{-2x}{n}(y_i - (mx+c)\right] \quad \therefore m - \alpha\left[\frac{-2}{n}\sum(y-\hat{y})x\right]$$

$$c = c - \alpha\left[\frac{-2}{n}\sum(y-\hat{y})\right]$$

URBAN EDGE
COLORFUL YOU

Learning Rate:

choosing of learning rate is a matter of trial and error

The reason we do not directly subtract dw (change in weight) from w because it might result in too much change in the value of w. and might not end up in a global minimum.

Q. Apply G.D on following obs and observe MSE value of 3 iterations $\alpha = 0.0001$ with $m = 10$, $b = 300$

age(x)	salary(y)
30	800
32	950
25	600
43	1050
50	1200
29	740
46	1100

initially ~~m=c=0~~ $m = 10$ $b = 300$

So, $\hat{y} = 10x + 300$.

$$\frac{\partial(cost)}{\partial m} = \frac{-2x}{n}(y_i - \hat{y}_i)$$

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

$$m = m - \alpha\left[\frac{-2}{n}\sum(y_i - \hat{y}_i)x\right] \qquad c = c - \alpha\left[\frac{-2}{n}\sum(y_i - \hat{y}_i)\right]$$

$$\sum (y_i - \hat{y}_i) = [800-600] + [950-620] + [600-550] + [1050-730] + [1200-800] + [740-590] + [1100-760]$$

$$= 1790$$

$$MSE = \frac{1}{7}\sum (y_i - \hat{y}_i)^2 = \frac{1}{7}(1790)^2 = 457428.57 \quad 78842.8$$

$$m_{new} = 10 - 0.0001\left[\frac{-2}{7}(71560)\right]$$

$$= 10 + 2.04457$$

$$= 7.9554 \quad 12.04457$$

$$b_{new} = 300 - 0.0001\left[\frac{-2}{7}(1790)\right]$$

$$= 299.948 \quad 300.05114$$

$$\hat{y}_i = 12.04457x + 300.05114 \quad (2^{nd} \text{ epoch})$$

$$\sum (y_i - \hat{y}_i)$$

$$= [800-661.388] + [950-685.477] + [600-601.165] + [1050-817.967] + [1200-902.279] + [740-649.39] + [1100-854.101]$$

$$= 1268.283$$

$$MSE = \frac{1}{7}(300349.72) = 42907.040$$

$$m_{new} = 12.04457 - 0.0001\left[\frac{-2}{7}(51397.93)\right]$$

$$= 13.513$$

$$b_{new} = 300.05114 - 0.0001\left[\frac{-2}{7}(1268.283)\right] = 300.087$$

URBAN EDGE
COLORFUL YOU

Multiple Linear Regression (G.D)

Two independent variable

i.e, x_1, x_2

$y = w_1 x_1 + w_2 x_2 + b$

$$MSE = \frac{1}{n}\left(\sum (y_i - \hat{y}_i)^2\right)$$

$$w_1 = w_1 - \alpha\left(\frac{\partial(\text{loss})}{\partial w_1}\right) = w_1 - \alpha\left[\frac{-2}{n}\sum (y_i - \hat{y}_i) x_{1i}\right]$$

$$w_2 = w_2 - \alpha\left(\frac{\partial(\text{loss})}{\partial w_2}\right) = w_{2a} - \alpha\left[\frac{-2}{n}\sum (y_i - \hat{y}_i) x_{2i}\right]$$

$$b = b - \alpha\left(\frac{\partial(\text{loss})}{\partial b}\right) = b - \alpha\left[\frac{-2}{n}\sum (y_i - \hat{y}_i)\right]$$

$$\boxed{y = \sum x_i w_i + b}$$

MLP is a method which is used to find linear out relationship b/w a response variable and more than one predictor variable

Q. Consider the following dataset. Find out linear relⁿ b/w the response var 'y' and two predictor variables x_1 & x_2 using G.D

y	x_1	x_2
140	60	22
155	62	25
159	67	24
179	70	20
192	71	15
200	72	14
212	75	14
215	78	11

$w_1 = 3$

$w_2 = -1.2$

$b = -6.1$

$\alpha = 0.001$ ($\alpha = 0.0001$)

epoch = 2

$y = 3x_1 - 1.2x_2 - 6.1$

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

$\sum y_i - \hat{y}_i = (140 - 147.5) + (155 - 149.9) + (159 - 166.1) + (179 - 179.9) + (192 - 188.9) + (200 - 193.1) + (212 - 202.1) + (215 - 214.7)$

$= 9.8$

$\sum(y_i - \hat{y}_i)x_{1i} = 810.3$

$\sum(y_i - \hat{y}_i)^2 = 288.8$

$\sum(y_i - \hat{y}_i)x_{2i} = 59.1$

$MSE = 36.1$

$w_{1new} = w_1 - \alpha\left[\frac{-2}{8}\sum(y_i - \hat{y}_i)x_{1i}\right]$

$= 3 - 0.001\left[-\frac{1}{4}(810.3)\right] = \cancel{6.2412}\ 3.202575$

$w_{2new} = w_2 - \alpha\left[\frac{-2}{8}\sum(y_i - \hat{y}_i)x_{2i}\right]$

$= -1.2 - 0.001\left[-\frac{1}{4}(59.1)\right] = \cancel{0.9636}\ -1.185$

$b_{new} = b - \alpha\left[\frac{-2}{8}\sum(y_i - \hat{y}_i)\right]$

$= -6.1 - 0.001\left[-\frac{1}{4}(9.8)\right] = \cancel{6.0608}\ -6.09$

~~$y = 6.2412x_1 - 0.9636x_2 - 6.0608$~~

Iteration 2. $y = 3.2x_1 - 1.185x_2 - 6.09$

Iteration 2

$\sum(y_i - \hat{y}_i) = (140 - 159.84) + (155 - 162.685) + (159 - 179.87) + (179 - 194.21) + (192 - 203.335) + (200 - 207.72) + (212 - 217.32) + (215 - 230.475)$

$= -103.455$

$\Sigma(y_i - \hat{y}_i) x_{1i} = -7096.535$

$\Sigma(y_i - \hat{y}_i) x_{2i} = -1956.495$

$\Sigma(y_i - \hat{y}_i)^2 = 1575.45$

$MSE = \frac{1}{8}(1575.45) = 196.93$

$w_{1new} = w_1 - 0.001\left[\frac{-1}{4}(-7096.535)\right]$

$= 3.2 - 1.774$

$= +1.426$

$w_{2new} = w_2 - 0.001\left[\frac{-1}{4}(-1956.495)\right]$

$= -1.185 - 0.489$

$= -1.674$

$b_{new} = b - 0.001\left[\frac{-1}{4}(-103.455)\right]$

$= -6.09 - 0.0258$

$= -6.1158$

ii) $\alpha = 0.0001$

$MSE = 36.1$

$w_{1new} = 3 - 0.0001\left[\frac{-1}{4}(810.3)\right] = 3.0203$

$w_{2new} = -1.2 - 0.0001\left[\frac{-1}{4}(59.1)\right] = -1.19$

$b_{new} = -6.1 - 0.0001\left[\frac{-1}{4}(9.8)\right] = -6.099$

$$y = 3.02034 - 1.19 x_2 - 6.099$$

MSE:

$$\sum (y_i - \hat{y}_i) = (140 - 148.939) + (155 - 151.4096) + (159 - 167.7011) + (179 - 181.522) + (192 - 190.49) + (200 - 194.70) + (212 - 203.76) + (215 - 216.39)$$

$$= -2.9117$$

$$\sum (y_i - \hat{y}_i) x_{1i} = -74.85 \qquad \sum (y_i - \hat{y}_i) x_{2i} = -169.249$$

$$MSE = \frac{1}{8}(275.06) = 34.38$$

Multiple Linear Regression using Least Square Method

MLR can be expressed as $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$

$$\boxed{y = x\beta} \qquad (y = x\beta + \epsilon) \rightarrow \text{error will always be there}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nk} \end{bmatrix} * \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_k \end{bmatrix} \quad \text{—(1)}$$

$[N \times k]$ $\qquad (N >> k)$

If we consider β_0 - then eq (1) can be expressed as:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & Nnk \end{bmatrix} * \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

$[N * (k+1)]$ $\qquad (N >> k+1)$

To estimate the parameters of β, we need to solve n equations and it will be difficult to solve this way because

$$N >> k+1$$

Consider the sum of sq of residuals, in order to estimate β. we have to minimize

$$\sum_{i=1}^{N} e_i^2$$

$$e_i^2 = e'e = \begin{bmatrix} e_1 & e_2 & \dots & e_N \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

This can be written as a optimization problem or equation i.e.

$$\min_{\beta} e'e$$

$$\frac{\partial e'e}{\partial \beta} = 0 \qquad [\because e = y - X\beta]$$

$$\Rightarrow \frac{\partial}{\partial \beta}[(y-X\beta)'(y-X\beta)]$$

$$\Rightarrow \frac{\partial}{\partial \beta}(y'y - y'X\beta - (X\beta)'y + (X\beta)'X\beta) = 0$$

$$\Rightarrow \frac{\partial}{\partial \beta}(y'y - (y'X\beta)' - X'\beta'y + (X\beta)'X\beta) = 0$$

$$\Rightarrow \frac{\partial}{\partial \beta}(y'y - yX'\beta' - X'\beta'y + (X\beta)'X\beta) = 0$$

$$\Rightarrow \frac{\partial}{\partial \beta}(y'y - 2\beta'X'y + \beta'X'X\beta) = 0$$

$$\Rightarrow 0 - 2X'y + 2\beta X'X = 0$$

$$\Rightarrow \beta = \frac{yX'}{X'X} = (X'X)^{-1} X'y$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}_{(k+1)\times 1} = \underbrace{(X'_{(k+1)\times N} X_{N\times(k+1)})}_{(k+1)\times(k+1)} \underbrace{X'_{(k+1)\times N}\, y_{N\times 1}}_{(k+1)\times 1}$$

$$= (k+1)\times 1$$

Q. Using LSM find out best fit regression eqn of the following dataset

x_1	x_2	y
2	1	1.7
5	3	2.6
6	4	2.8
8	9	2.3
10	11	2.7
11	14	2.4

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

$$\begin{bmatrix} 1.7 \\ 2.6 \\ 2.8 \\ 2.3 \\ 2.7 \\ 2.4 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 5 & 3 \\ 6 & 4 \\ 8 & 9 \\ 10 & 11 \\ 11 & 14 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix}$$

$$X = \begin{bmatrix} 2 & 1 \\ 5 & 3 \\ 6 & 4 \\ 8 & 9 \\ 10 & 11 \\ 11 & 14 \end{bmatrix} \qquad X' = \begin{bmatrix} 2 & 5 & 6 & 8 & 10 & 11 \\ 1 & 3 & 4 & 9 & 11 & 14 \end{bmatrix}$$

$$X = \begin{bmatrix} 2 & 1 \\ 5 & 3 \\ 6 & 4 \\ 8 & 9 \\ 10 & 11 \\ 11 & 14 \end{bmatrix}_{6\times 2} \qquad \begin{bmatrix} 2 & 5 & 6 & 8 & 10 & 11 \\ 1 & 3 & 4 & 9 & 11 & 14 \end{bmatrix}_{2\times 6}$$

1) $X'X = \begin{bmatrix} 2 & 5 & 6 & 8 & 10 & 11 \\ 1 & 3 & 4 & 9 & 11 & 14 \end{bmatrix}_{2\times 6} \begin{bmatrix} 2 & 1 \\ 5 & 3 \\ 6 & 4 \\ 8 & 9 \\ 10 & 11 \\ 11 & 14 \end{bmatrix}_{6\times 2}$

2) $X'X = \begin{bmatrix} 286 & 377 \\ 377 & 424 \end{bmatrix}$

$X'Y = \begin{bmatrix} 2 & 5 & 6 & 8 & 10 & 11 \\ 1 & 3 & 4 & 9 & 11 & 14 \end{bmatrix}_{2\times 6} \begin{bmatrix} 1.7 \\ 2.6 \\ 2.8 \\ 2.3 \\ 2.7 \\ 2.4 \end{bmatrix}_{6\times 1}$

$= \begin{bmatrix} 105 \\ 1047 \end{bmatrix}$

$(X'X)^{-1}(X'Y)$

$(X'X)^{-1} = \dfrac{adj(X'X)}{|X'X|}$

$|X'X| = \begin{vmatrix} 286 & 377 \\ 377 & 424 \end{vmatrix} = -20865$

$adj(X'X) = \begin{bmatrix} 424 & -377 \\ -377 & 286 \end{bmatrix}$

$$(X'X)^{-1} = \begin{bmatrix} -424/20865 & 29/1605 \\ 29/1605 & -22/1605 \end{bmatrix}$$

$$(X'X)^{-1}(X'Y) = \begin{bmatrix} -424/20865 & 29/1605 \\ 29/1605 & -22/1605 \end{bmatrix} \begin{bmatrix} 105 \\ 104.7 \end{bmatrix}$$

$$= \begin{bmatrix} -0.242 \\ 0.462 \end{bmatrix}$$

$$y = -0.242 x_1 + 0.462 x_2$$

Multiple linear Regression (Two independent variable) using LSM

Q find out best fit multiple linear regression eqn for following obs:

x_1	x_2	y	x_1^2	x_2^2	x_1y	x_2y	x_1x_2
60	22	140	3600	484	8400	3080	1320
62	25	155	3844	625	9610	3875	1550
67	24	159	4489	576	10653	3816	1608
70	20	179	4900	400	3580	12530	1400
71	15	192	5041	225	2880	13632	1065
72	14	200	5184	196	14400	2800	1008
75	14	212	5625	196	15900	2968	1050
78	11	215	6084	121	16770	2365	858
555	145	1452	38767	2823	101895	25364	9859

$$y = \beta_0 + \beta x_1 + \beta_2 x_2$$

$$\beta_1 = \frac{\sum x_2^2 \sum x_1 y - \sum x_1 x_2 \sum x_2 y}{\sum x_1^2 \sum x_2^2 - (\sum x_1 x_2)^2}$$

$$\beta_2 = \frac{\sum x_1^2 \sum x_2 y - \sum x_1 x_2 \sum x_1 y}{\sum x_1^2 \sum x_2^2 - (\sum x_1 x_2)^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}_1 - \beta_2 \bar{x}_2$$

Regression sum calculation

$$\sum x_1^2 = \sum x_1^2 - \frac{(\sum x_1)^2}{n}$$

$$\sum x_2^2 = \sum x_2^2 - \frac{(\sum x_2)^2}{n}$$

$$\sum x_1 y = \sum x_1 y - (\sum x_1 \sum y)/n$$

$$\sum x_2 y = \sum x_2 y - (\sum x_2 \sum y)/n$$

$$\sum x_1 x_2 = \sum x_1 x_2 - (\sum x_1 \sum x_2)/n$$

Step 2: find reg. sum

$$\sum x_1^2 = 38767 - \frac{(555)^2}{8} = 263.875$$

$$\sum x_2^2 = 2823 - \frac{(145)^2}{8} = 194.875$$

$$\sum x_1 y = 101895 - \frac{805860}{8} = 1162.5$$

$$\sum x_2 y = 25364 - \frac{210540}{8} = -953.5$$

$$\sum x_1 x_2 = 9859 - \frac{80475}{8} = -200.375$$

$$\beta_1 = \frac{(194.875)(1162.5) - (-200.375)(-953.5)}{263.875(194.875) - (-200.375)^2}$$

$$= \frac{-189700.1875}{11272.5} = \cancel{-16.83}\ 3.147$$

$$\beta_2 = \frac{(263.875)(-953.5) - (-200.375)(1162.5)}{11272.5}$$

$$= \cancel{20.60}\ \cancel{-42.98}\ -1.65$$

$$\beta_0 = \cancel{\frac{1452}{8}} - \cancel{(16.83)\frac{555}{8}} - \cancel{20.60\frac{(145)}{8}}$$

$$= \cancel{975.70}$$

$$\beta_0 = \frac{1452}{8} - (3.147)\frac{555}{8} - (-1.65)\frac{(145)}{8} = -6.9$$

Q MLR using gradient descent

$$\vec{X}_{(i)} = [x_1, x_2, x_3, \ldots, x_n]$$

$$f_{\vec{w},b}(\vec{x}^{(i)}) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

$$f_{\vec{w},b}(\vec{x}^{(i)}) = \vec{w} \cdot \vec{x}^{(i)} + b \quad \Big\} \text{ eq}^n$$

cost fun^n :-

$$J(\vec{w},b) = \frac{1}{2m} \sum_{i=1}^{m} (\cancel{f_{\vec{w},b}} y_i - (f_{\vec{w},b}(\vec{x}^{(i)}))^2$$

$$\frac{\partial}{\partial w_j} J(\vec{w}, b) = \frac{-2}{m} \sum_{i=1}^{m} (y_i - (\vec{w}\vec{x}^{(i)} + b)) x_j^{(i)}$$

j: no. of features

$$\frac{\partial}{\partial b} J(\vec{w}, b) = \frac{-2}{m} \sum_{i=1}^{m} (y_i - (\vec{w}\vec{x}^{(i)} + b))$$

update w & b using the following formula till convergence

$$w_j = w_j - \alpha \left(-\frac{2}{m} \sum_{i=1}^{m} (y_i - (\vec{w}\vec{x}^{(i)} + b)) x_j^{(i)} \right)$$

$$b = b - \alpha \left(\frac{-2}{m} \sum_{i=1}^{m} (y_i - (\vec{w}\vec{x}^{(i)} + b)) \right)$$

URBAN EDGE
COLORFUL YOU

Analysis: (Gradient Descent Method)

Pseudocode:

1. repeat until convergence {
2. $\beta_j = \beta_j - \alpha \cdot \frac{\partial}{\partial \beta_j} J(\beta_0, \beta_1)$
3. }

where $J(\beta_0, \beta_1) = \min_{\beta_0, \beta_1} \frac{1}{2N} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

N: no of samples

Implementation: (Initially $\beta_0 = \beta_1 = 0$)

let $temp_0 = \beta_0 - \alpha \frac{\partial}{\partial \beta_0} J(\beta_0, \beta_1)$

$temp_1 = \beta_1 - \alpha \frac{\partial}{\partial \beta_1} J(\beta_0, \beta_1)$

$\beta_0 \leftarrow temp_0$

$\beta_1 \leftarrow temp_1$

} need to be repeated until convergence

Convergence criteria

i.e., the user has to provide the error tolerance (ϵ)

$\epsilon = 0.001$ (let)

Let at i^{th} step :- β_{0i}, β_{1i}

at $(i+1)^{th}$ step :- $\beta_{0(i+1)}$, $\beta_{1(i+1)}$

if $(|\beta_{0(i+1)} - \beta_{0i}| \;\&\&\; |\beta_{1(i+1)} - \beta_{1i}| \leq \epsilon)$

$\Rightarrow$ STOP the iteration

$\therefore \beta_{0final} = \beta_{0(i+1)}$ & $\beta_{1final} = \beta_{1(i+1)}$

$\Rightarrow \boxed{\hat{y}_{new} = \beta_{0final} + \beta_{1final} * x_{new}}$ → new sample for which target value is to be estimated

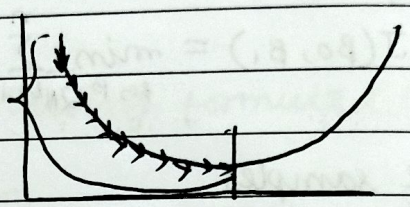
Analysis of Gradient Descent

1. The rate of convergence of gradient descent depends on the learning rate.

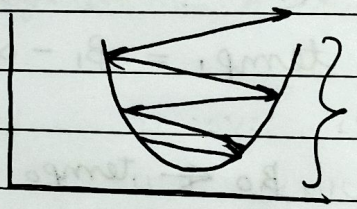
2. If (α is too small) ⇒ no of iterations are large.

possible values of
α = {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}

↳ no learning



3. If (α is too large) ⇒ no of iterations may overshoot or iterations might diverse



4. If(α is fixed)

⇒ Then algorithm might be TRAPPED into a local MINIMA

i.e., we may not reach global MINIMA

i.e, the cost function may not tends to zero

⇒ It lacks the accuracy

5. Remedy:

i. choose a suitable learning rate : α

ii. Then apply iteration for that α

iii. Compute accuracy, precision, recall, F1-score, similarity Index

iv. whether(the observed parameters == OK)

⇒ STOP

else change the learning rate & repeat above steps.

There are different types of Gradient Descent are used:

1. Batch G.D
2. Stochastic G.D
3. Mini Batch G.D

... continued in *

Normalization & Standardization (feature scaling)

- Feature scaling is one of the most imp data preprocessing step of machine learning
- If we have to compute the distance b/w the features it may bias towards numerically larger values if the data is not scaled
- There are two main feature scaling techniques:
  - normalization
  - standardization

normalization

It is also called min-max scalling. It is used to transform features to be on a similar scale using the given formula.

$$x_{new} = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$

This scale ranges from [0, 1] or sometimes [-1, 1].

Standarization AKA Z-score normalization

It transforms the features by subtracting it from mean and divide by the std. deviation

i.e,

$$x_{new} = \frac{(x - \bar{x})}{\sigma}$$

Generally std deviation is used when data follows a gaussian distribution

Normalization	Standardization
1. min & max value of features are used	1. mean and std deviation is used
2. used when features are of diff scales	2. used when we want to ensure zero mean & unit std. deviation
3. scales bet^n [0,1] or [-1,1]	3. not bounded to range
4. It is affected by outliers	4. much less affected by outliers
5. transformation squishes n-dimensional data into n-dimensional unit hypercube	5. translates data to mean vector of original data to origin & squishes / expands
6. useful when we do not know about distribution	6. useful when feature distribution is normal/gaussian

Bias and Variance

Bias: It refers to error due to overly simplistic assumptions in learning models / algorithm. These assumptions make model easier to

comprehend & learn but might not capture underlying complexities of data. When a model does not perform well neither in training nor in testing that means it has high bias & indicate underfitting.
It has high bias & low variance (underfitting model)

Reasons for underfitting
1. Model is too simple
2. no. of input features are not sufficient
3. The size of training sample is not enough
4. The features are not scaled well.

Techniques to reduce underfitting
1. increase model complexity
2. increase no. of i/p features
3. remove noise from data & increase training sample
4. increase no. of epochs during training

- The learning rate in gradient descent is a parameter that controls size of steps taken to adjust model parameters
→ It determines how quickly the algorithm will converge to the optimal values of model parameters
→ If it is too small, it may traverse local minima and take long time to converge
→ If its too large, it may overshoot minima & diverge
→ To find optimal learning rate, we can start with a high learning rate & gradually reducing it during training

URBAN EDGE COLORFUL YOU

TYPES OF GRADIENT DESCENT:

→ Batch GD the entire dataset is used to calculate the gradient and update the parameters of relation model.

Pros:- simple algo only read to compute gradient
- fixed learning rate be used during training
- very quick convergence ratio to a global minimum if loss function is convex.

Cons:- it may be slow on huge datasets
- may not work for non convex functions

→ Stochastic GD: a single sample is used to update the parameters unlike GD.
samples can be chosen randomly or in a cyclic rule

Pros:- It converges quickly than Batch GD
It can escape from local minima

Cons: It requires more iterations to converge with limits

→ Mini Batch GD: It is the fine balance between Stochastic and Batch GD in which a subset of observation is used to update the gradient
The no. of samples used for each size is called batch size & each iteration over a batch is called epoch.
Here one more parameter needs to be optimized that is the batch size

Variance: It is the error due to model sensitivity to fluctuation in the training data.

High variance occurs when a model learns the training data's noise and random fluctuation rather than the underlying pattern as a result the model performs well on the training data but perform poorly on the testing data. This is a case of overfitting

Reasons of overfitting
1. high variance & low bias
2. the model is too complex
3. the size of training data

Techniques to reduce overfitting:
1. by focusing on meaningful patterns
2. by ignoring irrelevant features
3. increase the size of training data that may improve the models ability to generalize
4. Ridge & Lasso Regularization
5. early stopping (stop training when error is stagnant)

★ Ridge Regression vs Lasso Regression

Both the regularization used for regularising linear model to avoid overfitting and improve the predictive performance

Both methods add a penalty term to the models cost function to constraint the coefficient

Ridge Regression also know as L2 regularization it adds the squared of the coeff as a penalty. However, Lasso regression or L1 regularization introduced a penalty that is the absolute value of coefficient

The loss function in the ridge regression as follows

$$\text{Loss} = MSE + \lambda \sum_{i=1}^{n} w_i^2$$

where, λ is the regularization parameters that controls strength of the penalty & w_i are the coefficient

$$\text{Loss} = RSS + \sum_{i=1}^{n} w_i^2$$

The cost function of lasso regression is

$$\text{Loss} = MSE + \sum_{i=1}^{n} |w_i|$$

$$\text{Loss} = RSS + \sum_{i=1}^{n} |w_i|$$

- In ridge regression the coefficient shrinks towards zero whereas in lasso regression the coeff may shrink to exactly zero, thereby it performs the feature selection by eliminating the features whose coeff reduced to zero.
- When the data have multicolinearity exists among the independent variable apply ridge

regularization otherwise apply Lasso.

Lasso stands for Least absolute selection and shrinkage Operator.

Closed Form Equation of Linear Regression

$$y = \sum_{i=1}^{n} w_i x_i + \text{bias} \qquad w = (X'X)^{-1} X'Y$$

simple linear regression, two coefficient

w_0 & w_1

$$w_1 = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sum (x-\bar{x})^2}$$

The closed form solutions are prefered because they are faster than the iterative optimization algo like G.D.

$$\left\{ y = \left(\frac{\sum xy}{\sum x^2}\right) x + \frac{\sum y}{n} - \frac{\sum x \sum y}{\sum x^2 n} \right\}$$ closed form eqn to predict the value of y.

★ CLASSIFICATION ★

- It is a supervised learning algorithm that is used to identify the level or category of new observation (test sample) on the basis of no. of obs (training samples)

- In the training samples all the features are of real values or continous or numeric value whereas the outcome of class level are categorical.

K-NN Algorithm

- It is a supervised learning algorithm which is used to predict the category of a base sample based on the category of nearest neighbours.
- It is a non parametric method that makes prediction based on the similarity of datapoints in a given data set.
- It is used for classification as well as regression
- It is also called an lazy learner algorithm because it does not go through the training process & directly performs classification for the test sample whenever available

Distance Matrix

- Euclidean Distance

$$x_2 = (x_1, x_2) \qquad y_2 = (y_1, y_2)$$

$$ED(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

- Manhatten distance

$$dist(x, y) = |x_1 - y_1| + |x_2 - y_2|$$

- Minkowski Distance

$$dist(x, y) = \sqrt[p]{|x_1 - y_1|^p + |x_2 - y_2|^p}$$

URBAN EDGE

Steps to perform K-NN

1: Initialize K (K to be chosen as a odd number, K can be chosen as a value of sq.rt of n, K can be chosen from the elbow method, K can be chosen from cross validation).

2. Calculate distance from each training sample to the test sample using any of the dist formula

3. Sort the distance and training samples in the ascending order by the distance

4. Pick the first K training sample & get their levels

5. Declare the class level of test sample based on the majority of class level of K-neighbours

Example:

X_1	X_2	Class label
40	20	C_1
50	50	C_2
60	90	C_2
10	25	C_1
25	80	? (Test sample)

let K = 3.

$y_1 = 25 \quad y_2 = 80$

$d_1 = \sqrt{(25-40)^2 + (80-20)^2} = \sqrt{225+3600} = 61.84$

$d_2 = \sqrt{(25-50)^2 + (50-80)^2}, \sqrt{(25)^2 + (30)^2} = 39.05$

ED(X, Y)

X_1 X_2

$d3 = \sqrt{(60-25)^2 + (90-80)^2} = 36.40$

$d4 = \sqrt{(10-25)^2 + (25-80)^2} = 57.008$

$d3 < d2 < d4 < d1$

C2, C2, C1

So, class of (25, 80) is C2

Adv:

- simple Algo, less complex
- very few parameters to be considered
- Does not require training

Disadv:

- chosing k value is difficult
- if you choose very small value model may overfit because of high variance
- if k value is large it may underfit because of high bias
- it is sensetive to outliers
- it does not scale well, not suitable for large dataset

How K-NN is used for regression

URBAN EDGE
COLORFUL YOU

Performance Matrixes of Classification Model

To evaluate the performance of a classification model a confusion matrix needs to be calculated.

A confusion matrix is a table that shows how well a classification model is performing. It helps to identify which classes of data are most often correctly classified or misclassified.

Actual \ Predicted	1	0
1	TP	FN
0	FP	TN

Confusion Matrix

1: +ve class
0: -ve class

Performance Matrixes.

Accuracy : $\frac{TP + TN}{\text{Total instances}}$

Precision : $\frac{TP}{TP + FP}$

Recall : $\frac{TP}{TP + FN}$

Specificity : $\frac{TN}{TN + FP}$

F1-score : $\frac{2 * \text{precision} * \text{Recall}}{\text{precision} + \text{recall}}$

Recall or sensitivity:

45	5
8	92

$$\text{Accuracy} = \frac{45+92}{45+5+8+92} = \frac{137}{150} = 0.913$$

$$\text{Precision} = \frac{45}{45+8} = \frac{45}{53} = 0.849$$

$$\text{Recall} = \frac{45}{45+5} = \frac{45}{50} = 0.9$$

$$\text{Specificity} = \frac{92}{92+8} = \frac{92}{100} = 0.92$$

$$\text{F1-score} = \frac{2 \times 0.842 \times 0.9}{0.849 + 0.9} = 0.8665$$

Confusion Matrix for Multi class:

Actual \ Predicted	C1	C2	C3
C1	TP_9	FN_1	FN_5
C2	FP_6	TN_7	TN_4
C3	FP_3	TN_2	TN_8

C1 : 1
C2 : 0
C3 : 0

Precision (C1) = $\frac{TP}{TP+FP} = \frac{9}{9+9} = 0.5$

Precision (C2) = $\frac{7}{7+3} = 0.7$

C2 : +ve

	C1	C2	C3
C1	TN	FP	TN
C2	FN	TP	FN
C3	TN	FP	TN

Precision (C3) = $\frac{8}{8+9} = \frac{8}{17} = 0.47$

C3 : +ve

	C1	C2	C3
C1	TN	TN	FP
C2	TN	TN	FP
C3	FN	FN	TP

Recall (C1) = $\frac{9}{9+1+5} \quad \frac{9}{15} = 0.6$

Recall (C2) = $\frac{7}{7+6+4} = \frac{7}{17} = 0.411$

Recall (C3) = $\frac{8}{8+3+2} = \frac{8}{13} = 0.615$

F1-score (C1) = $\frac{2 \times 0.5 \times 0.6}{0.5+0.6} \quad \frac{0.6}{1.1} = 0.54$

F1- score (C2) = $\frac{2 \times 0.7 \times 0.411}{0.7+0.411} = 0.517$

F1- score (C3) = $\frac{2 \times 0.47 \times 0.615}{0.47+0.615} = 0.53$

Accuracy of entire model = $\frac{9+7+8}{46} = 0.53$

Logistic Regression

It is a supervised ML algorithm used for classification. It is a statistical algorithm which analyzes relationship b/w two data factors

The goal of logistic regression is to predict the probability than an instance belongs to a given class or not.

It is used for usually binary classification where we use sigmoid funt that takes input as independent variables and produces a probability value b/w 0 & 1

If the value of the logistic function (sigmoid funt) is greater than 0.5 (threshold value) then it belongs to class 1 otherwise it belongs to class 0.

There are 3 types of logistic Regression:

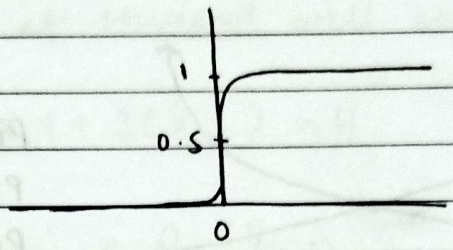
It can be applicable for

1. Binomial or Binary classification
2. Multinominal classification (more than 2 classes)
3. Ordinal classification

Eg:

Sigmoid Function

$$\frac{1}{1+e^{-z}}$$



$z = \beta_1 x_1 + \beta_0$ (if one feature/independent var).

$z = \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_K x_K + \beta_0$ [K features]

Steps of Logistic Regression for Binary classification

Given

$\{x_i, y_i\}_{i=1}^{N}$ $x_i \in R^p$, $y_i \leftarrow \{0, 1\}$

predict y_{new} for any x_{new}.

step 1: initialize β and α and choose $\epsilon > 0$.

(β: all coeff. values α: learing rate)

step 2: compute z

$$z_i = \left(\sum_{i=1}^{n} x_i \beta_i\right) + \beta_0$$

step 3: Then compute $p_i = \frac{1}{1+e^{-z_i}}$ $\begin{cases} y_i = 1 & \text{if } p_i > 0.5 \end{cases}$

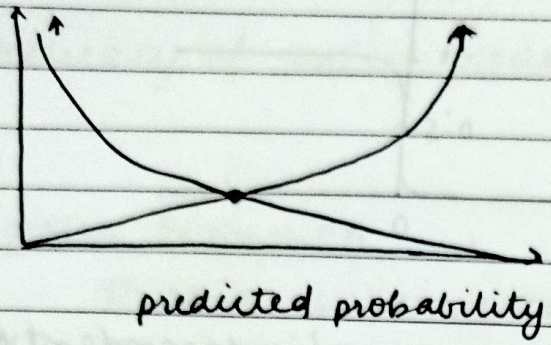
step 4: Then calculate cost function, i.e.

$$\text{cost} = \sum_{i=1}^{n} y_i \log(p_i) + (1-y_i)\log(1-p_i)$$

This is called log loss function or binary cross entropy function.

step 5: Compute $\beta_{i\,new} = \beta_{i\,old} + \alpha(y_i - p_i)x_i$

Repeat this process until the stopping criteria.



$P_i \to 1$ if $z \to \infty$

$P_i \to 0$ if $z \to -\infty$

$P_i = 0.5$ if $z = 0$

Q. Consider the dataset of 5 obs of no. of hours studied by a student. Predict the class level of the student who studied for 30 hrs using logistic regression. Calculate how many hours a student should study so that the student will pass with 95% of probability

Hrs of studies	Pass/Fail (class level)
28	0
14	0
32	1
27	1
38	1

$\beta_1 = 2 \quad \beta_0 = -64 \quad x = 30$

$\log(\text{odds}) = z = \beta_0 + \beta_1 x$

$\Rightarrow z = -64 + 2*(30)$

$\Rightarrow z = -4$

$$p = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{+4}} = 0.017$$

$\hat{y} = 1$ if $p > 0.5$

0 if $p < 0.5$

$\therefore$ class level = fail or 0

Calculate class level if student will study for 34 hrs

$z = \beta_0 + \beta_1 x_4 = -64 + 2(34) = 4$

$P = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-4}} = 0.98$

2. $P = \frac{1}{1+e^{-z}}$ $P = 0.95$

$0.95 = \frac{1}{1+e^{-z}} \Rightarrow \frac{1}{1+e^{-(\beta_0+\beta_1 x_4)}} = 0.95$

$\Rightarrow \frac{1}{1+e^{-(-64+2x)}} = 0.95 \Rightarrow \frac{1}{1+e^{-(2.94)}} = 0.95$

$\Rightarrow z = 2.94$

$\Rightarrow x = 33.47$

NAIVE BAYES CLASSIFIER

- It is a supervised ML classification used for classification task
- It is based on bayes theorem to probability to classify the data based on given features of diff classes.
- It is a simple probabilistic classifier and it has few parameters to build ML model to perform classifications.
- It assumes that one feature is independent of existance of another feature. In other words each feature contribute to the prediction with no relation b/w each other
- This algo is used in spm filteration, and classifying article & objectives
- It is named as naive because it assumes the presence of one feature does not affect other feature

and bayes part of the name because it refers to bayes theorem

Assumptions

1. Feature independence
2. Continous feature are normally distributed
3. Discrete feature have multinational distance
4. features are equally important
5. no missing data

Bayes Formula

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

where A and B are events $P(B) \neq 0$

where this is called posterior probability [P(A|B)]

P(B|A) - Likelihood or conditional prob (evidence given that prob of hypothesis is true)

P(A) - prior probability

This eq can also be written in this way:

$$P(y|X) = \frac{P(X|y)\,P(y)}{P(X)}$$

where X is the feature vector of size n & Y is class variable.

- Example: consider given dataset & find out the test sample today (x) = {Sunny, Hot, Normal, False}
The class level using naive bayes classification.

To reach out the result we need to calculate

$$P(y|(x = x_1, x_2, \ldots x_n) = \frac{P(x_1|y) \cdot P(x_2|y) \cdot P(x_n|y) \cdot P(y)}{P(x_1) \cdot P(x_2) \cdot \ldots P(x_n)}$$

OR $$P(y|x_1 x_2 x_3 \ldots x_n) = P(y) \prod_{i=1}^{n} P(x_i|y)$$

$$y = \arg\max P(y) \prod_{i=1}^{n} P(x_i|y)$$

	Outlook	Temp	Humidity	Windy	Play Golf
1	Rainy	Hot	High	false	No
2	Rainy	Hot	High	True	No
3	Overcast	Hot	High	F	Yes
4	Sunny	Mild	High	F	Yes
5	Sunny	Cool	Normal	F	Yes
6	Sunny	Cool	Normal	T	No
7	Overcast	Cool	Normal	T	Yes
8	Rainy	Mild	High	F	No
9	Rainy	Cool	Normal	F	Yes
10	Sunny	Mild	Normal	F	Yes
11	Rainy	Mild	Normal	T	Yes
12	Overcast	Mild	High	T	Yes
13	Overcast	Hot	Normal	F	Yes
14	Sunny	Mild	High	T	No

Class	Outlook	Temp	Humidity	Windy	Prior Prob P(C)
Yes	P(Sunny|Y) = 3/9	P(H|Y) = 2/9	P(N|Y) = 6/9	P(F|Y) = 6/9	P(Y) = 9/14
NO	P(Sunny|N) = 2/5	P(H|N) = 2/5	P(N|N) = 1/5	P(F|N) = 2/5	P(N) = 5/14

$P(C_k|X) = P(X|C_k) \cdot P(C_k)$

$P(\text{Yes}|\text{today}) = P(x_1|C_k) \cdot P(x_2|C_k) \ldots P(x_n|C_k) P(C_k)$

$= 3/9 \times 2/9 \times 6/9 \times 6/9 \times 9/14$

$= 0.021$

$P(\text{No}|\text{today}) = 2/5 \times 2/5 \times 1/5 \times 2/5 \times 5/14$

$= 0.004$

$y = \arg\max_y P(y) * \prod_{i=1}^{n} P(x_i|y)$

class label of the test sample today is 'yes'.

Find out the class label of the test sample $X = (\text{Rainy}, \text{Hot}, \text{High}, \text{False})$ using naive Bayes

Class	Outlook	Temp	Humidity	Windy	Prior Prob
Yes	2/9	2/9	3/9	6/9	P(Y) = 9/14
No	3/5	2/5	4/5	2/5	P(N) = 5/14

$P(\text{Yes}|\text{today}) = 2/9 \times 2/9 \times 3/9 \times 6/9 \times 9/14 = 0.007$

$P(\text{No}|\text{today}) = 3/5 \times 2/5 \times 4/5 \times 2/5 \times 5/14$

$= 0.027$

Class is 'no'.

Advantage:

- easy to implement and computationaly efficient
- it is effective with a large no. of feature
- it performs well with the limited training data
- it also performs well in the presence of categorical features
- for numerical features, it performs well if it shows normal distribution

Disadvantage:

- it may be influenced by irrelevant attributes so feature selection is required before applying naive Bayes classifier.
- It may assign zero probability to unseen events leading to poor generalisation
- As it assumes all features are independent which may not always hold true in the real world data.

Q. Consider the given dataset. Predict the class level of given test sample $X = \{f11, f21, f31\}$ using naive Bayes classifier based on the following obs

Using NB classifier based on following obs

class label	f11	f21	f31	Total
obj1	350	450	0	650
obj2	400	300	350	400
obj3	50	100	50	150

summarized table or frequency table

$X = \{f11, f21, f31\}$ class table.

$$P(obj1 \mid X) = P(X \mid obj1) \cdot P(obj1)$$
$$= P(f11 \mid obj1) \cdot P(f21 \mid obj1) \cdot P(f31 \mid obj1) \cdot P(obj1)$$
$$= \frac{350}{650} \times \frac{450}{650} \times \frac{0}{650} \times \frac{650}{1200}$$

$$P(obj2 \mid X) = P(X \mid obj2) \cdot P(obj2)$$
$$= P(f11 \mid obj2) \cdot P(f22 \mid obj2) \cdot P(f31 \mid obj2) \cdot P(obj2)$$
$$= \frac{400}{400} \times \frac{800}{400} \times \frac{350}{400} \times \frac{400}{1200} = 0.218$$

$$P(obj3 \mid X) = P(f11 \mid obj3) \cdot P(f21 \mid obj3) \cdot P(f31 \mid obj3) \cdot P(obj3)$$
$$= \frac{50}{150} \times \frac{100}{150} \times \frac{50}{150} \times \frac{150}{1200} = 0.009$$

Gaussian Naive Bayes Classification

$$P(Y=c \mid X) = P(X \mid Y=c) \cdot P(Y=c)$$

$$P(X \mid Y=c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(x-\mu_c)^2}{2\sigma_c^2}}$$

μ_c = mean of class c

σ_c = std. deviation of class c

Q. Based on given dataset determine class level of the test sample $X = \{x_1 = 6, x_2 = 130, x_3 = 8\}$ using Naive Bayes classifier.

f_1	f_2	f_3	class
6	180	12	C1
5.92	190	11	C1
5.58	170	12	C1
5.78	165	10	C1
5	100	6	C2
5.5	150	8	C2
5.4	130	7	C2
5.72	150	9	C2

$\mu_{f_1C_1} = 6+5.92+5.58+5.78/4 = 5.82$

$\mu_{f_1C_2} = 5+5.5+5.4+5.72/4 = 5.405$

$$\sigma_{f_1C_1} = \sqrt{\frac{\sum (x-\bar{x})^2}{n-1}} = \sqrt{\frac{(6-5.8)^2+(5.92-5.8)^2+(5.58-5.8)^2+(5.78-5.8)^2}{3}}$$

$= 0.185$

$$\sigma_{f_1C_2} = \sqrt{\frac{(5-5.405)^2+(5.5-5.405)^2+(5.4-5.405)^2+(5.72-5.405)^2}{3}}$$

$= 0.301$

$P(C_1|X) = P(f_1|C_1) \cdot P(f_2|C_1) \cdot P(C_1) \cdot P(f_3|C_1)$

$\mu_{f_2C_1} = 180+190+170+165/4 = 176.25$

$\mu_{f_2C_2} = 100+150+130+150/4 = 132.5$

$$\sigma_{f_2C_1} = \sqrt{\frac{(180-176.25)^2+(190-176.25)^2+(170-176.25)^2+(165-176.25)^2}{3}}$$

$= 11.08$

$\sigma_{f_2C_2} = 23.63$

$\mu_{f3C1} = 12+11+12+10/4 = 11.25$

$\mu_{f3C2} = 6+8+7+9/4 = 7.5$

$\sigma_{f3C1} = 0.96$

$\sigma_{f3C2} = 1.3$

$P(f1|C1) = \frac{1}{\sqrt{2\pi(0.185)^2}} e^{-\frac{(6-5.82)^2}{2(0.185)^2}} = 0.14$

$P(f2|C1) = \frac{1}{\sqrt{2\pi(11.08)^2}} e^{-\frac{(130-176.25)^2}{2(11.08)^2}} = 0.000037$

$P(f3|C1) = \frac{1}{\sqrt{2\pi(0.96)^2}} e^{-\frac{(8-11.25)^2}{2(0.96)^2}} = 0.00073$

$P(C1|X) = P(C1) \cdot P(f1|C1) \cdot P(f2|C1) \cdot P(f3|C1)$

$= 0.5 \times 0.147 \times 0.00003 \times 0.00073 = 0.000063$

$= 1.8 \times 10^{-9}$

$P(f1|C2) = \frac{1}{\sqrt{2\pi(0.3)^2}} e^{-\frac{(6-5.4)^2}{2(0.3)^2}} = 0.03386$

$P(f2|C2) = \frac{1}{\sqrt{2\pi(23.6)^2}} e^{-\frac{(130-132.5)^2}{2(23.6)^2}} = 0.22$

$P(f3|C2) = \frac{1}{\sqrt{2\pi(1.3)^2}} e^{-\frac{(8-7.5)^2}{2(1.3)^2}} = 0.21$

$P(C2|X) = 0.5 \times 0.143 \times 0.124 \times 0.0004 = 0.0000083$

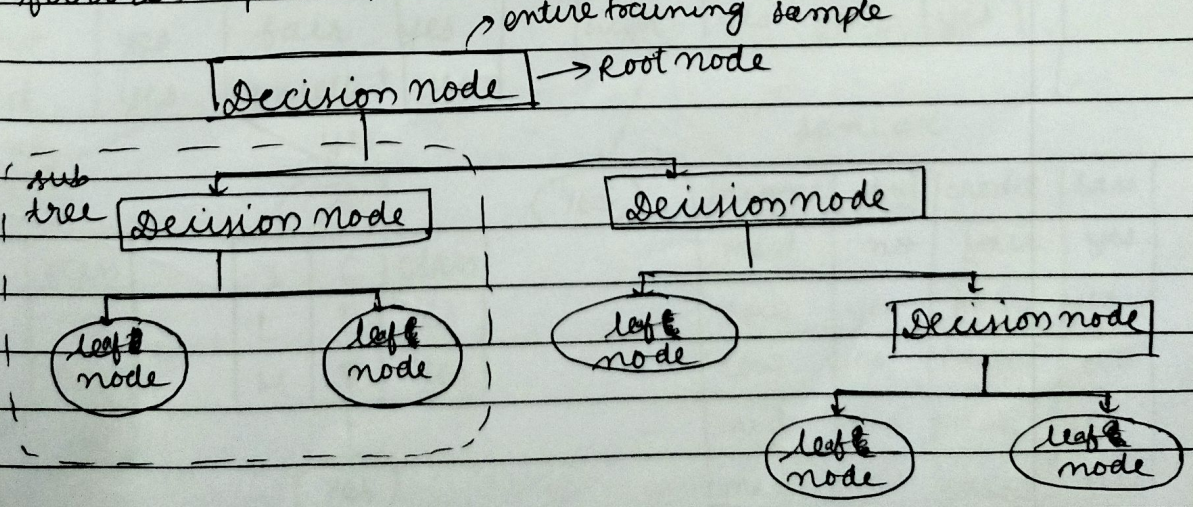
$= 7.6 \times 10^{-4}$

C2

Advantages of naive Bayes classifier

Decision Tree

- It is a supervised learning technique that can be used for both classification & regression
- It is a tree structured classifier where internal nodes represent features of dataset, branches represent the decision rules and, each leaf node represents the outcome
- The decision or the test are performed on the basis of feature on the given dataset.
- It is a graphical representation for getting all the possible solution to a problem based on given conditions
- In order to construct a decision tree we use the CART algorithm which stands for Classification and Regression Tree Algorithm. Ba
- Based on the outcome of a test on a feature it further splits the tree into subtrees.



Decision tree generation consists of two phases :-

1. Tree Construction

At start all the training examples are at the root

partition examples recursively based on selected attributes

2. Tree pruning

Identify & remove branches that reflect noise or outliers

The Testing Phase of Decision Tree :-

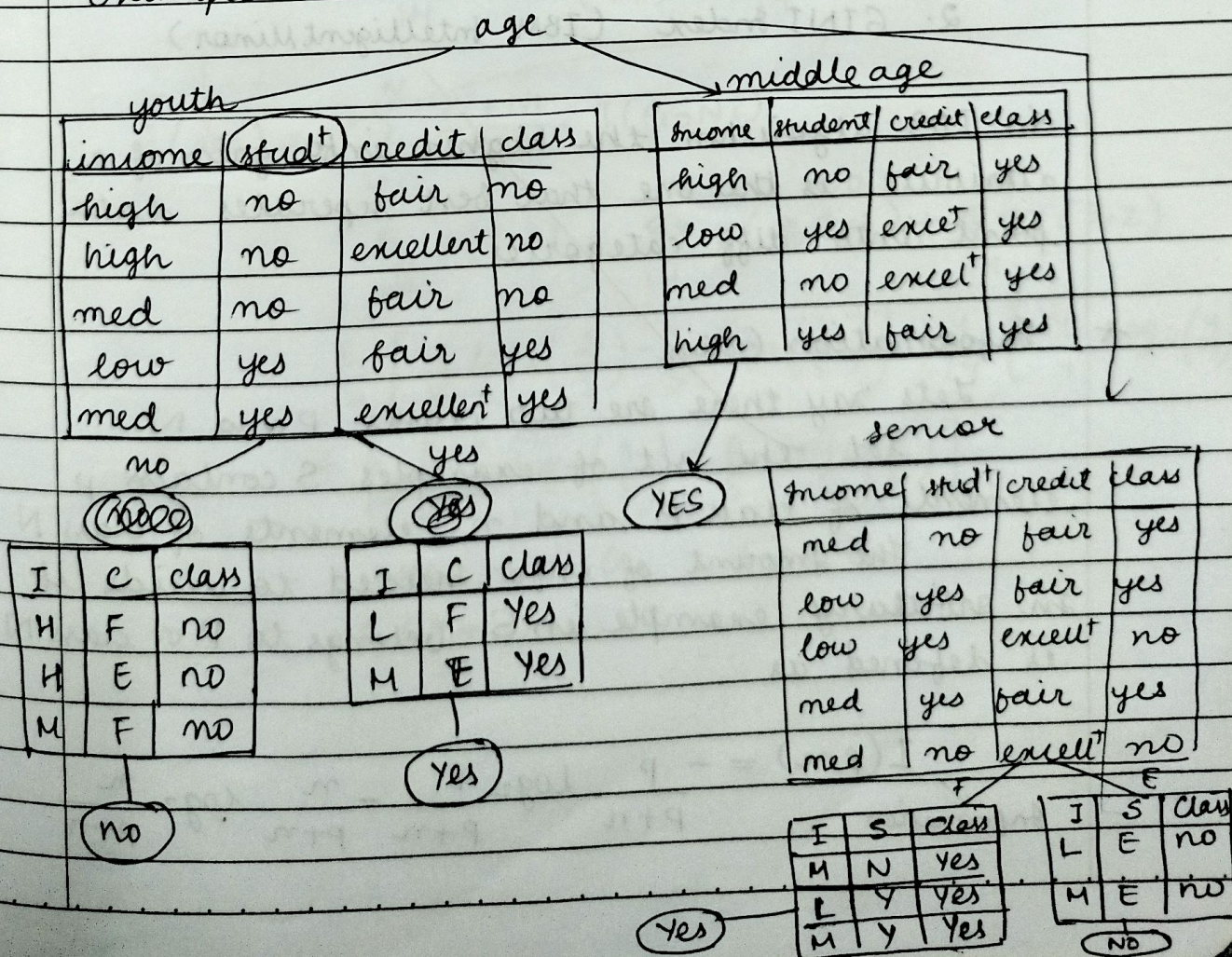
Classify an unknown sample by carried out the test on the attribute values of test sample against the decision tree.

Example

Algorithm for Decision Tree

1. At start all the training tuple are at the root
2. Tuple's are ~~partioned~~ partitioned recursively based on selected attributes
3. If all samples of a given node belong to same class → Label the class
4. If there are no remaining attributes for further partioning → Majority voting is employed for classifying the leaf
5. If there are no sample left, Label the class and terminate
6. Else go to step 2.

Example



Algorithm (Decision Tree)

- Attribute selection measure finds the best attribute in sequence to provide a splitting role that determine how the tuple at a given node can be a best split. There are two ways to find out the best feature for splitting :

1. Information gain

based on information gain, the decision tree are of two types :

* ID3 (Iterative Dichotomiser 3)
* C4.5

All attributes are assumed to be categorical. If continous values of attributes are there use preprocessing steps to convert into categorical.

2. GINI Index (IBM Intelligent Miner)

In ID3 algorithm the higher info gain of a attribute is the one that best seperates data point into diff categories

* Information Gain

Lets say there are two classes P and N

let the set of examples S contain p elements of class P and n elements of class N

The amount of info, needed to decide if an arbitrary example in S belongs to P or class N is defined as

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Information

Assume that using attribute A a set S will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$

If S_i contains P_i example of class P & N_i examples of class N, the entropy or expected information needed to classify objects in all subtree S_i is

$$E(A) = \sum_{i=1}^{v} \frac{P_i + N_i}{P+N} I(P_i, N_i)$$

(v: no. of categorical values of attribute)

The encoding info that would be gained by branching on A

$$Gain(A) = I(P,N) - E(A).$$

* Construct Decision Tree (ID3).

$$I(yes, no) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.94$$

$$E(age) = \sum_{i=1}^{N} \frac{P_i + N_i}{P+N} I(P_i, N_i)$$

$$= \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2)$$

$$= \frac{5}{14}\left[\frac{-2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right] + \frac{4}{14}\left[\frac{-4}{4}\log_2\left(\frac{4}{4}\right)\right]$$

$$+ \frac{5}{14}\left[\frac{-3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{3}{5}\right)\right]$$

$$= 0.3467 + 0 + 0.3467$$

$$= 0.69$$

$$E(\text{income}) = \frac{4}{14} I(2,2) + \frac{6}{14} I(4,2) + \frac{4}{14} I(3,1)$$

$$= \frac{4}{14}\left[-\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right] + \frac{6}{14}\left[-\frac{4}{6}\log_2\left(\frac{4}{6}\right) - \frac{2}{6}\log_2\left(\frac{2}{6}\right)\right] + \frac{4}{14}\left[-\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right]$$

$$= 0.2857 + 0.3935 + 0.2317$$

$$= 0.91$$

$$E(\text{credit}) = \frac{8}{14} I(6,2) + \frac{6}{14} I(3,3)$$

$$= \frac{8}{14}\left[-\frac{6}{8}\log_2\left(\frac{6}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right)\right] + \frac{6}{14}\left[-\frac{3}{6}\log_2\left(\frac{3}{6}\right) - \frac{3}{6}\log_2\left(\frac{3}{6}\right)\right]$$

$$= 0.4635 + 0.4285$$

$$= 0.89$$

$$E(\text{student}) = \frac{7}{14} I(3,4) + \frac{7}{14} I(6,1)$$

$$= \frac{7}{14}\left[\frac{-3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{3}{7}\right)\right] + \frac{7}{14}\left[\frac{-6}{7}\log_2\left(\frac{6}{7}\right) - \frac{1}{7}\log_2\left(\frac{1}{7}\right)\right]$$

$$= 0.4926 + 0.2958$$

$$= 0.78$$

Gain (age) = 0.94 − 0.69 = 0.25

Gain (income) = 0.94 − 0.91 = 0.03

Gain (student) = 0.94 − 0.78 = 0.16

Gain (credit) = 0.94 − 0.89 = 0.05

Dataset (age)

youth

I	S	C	class
H	N	F	no
H	N	E	no
M	N	F	no
L	Y	F	yes
M	Y	E	yes

middle

I	S	C	class
H	N	F	yes
L	Y	E	yes
M	N	E	yes
H	Y	F	yes

yes

senior

I	S	C	class
M	N	F	yes
L	Y	F	yes
L	Y	E	no
M	Y	F	yes
M	N	E	no

$$I(yes, no) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.97$$

$$E(income) = \frac{2}{5} I(0,2) + \frac{2}{5} I(1,1) + \frac{1}{5} I(1,0)$$

$$= \frac{2}{5}\left[-\frac{2}{2}\log_2\left[\frac{2}{2}\right]\right] + \frac{2}{5}\left[\frac{-2}{2}\log_2\left(\frac{1}{2}\right)\right] + \frac{1}{5}\left[\frac{1}{1}\log_2(1)\right]$$

$$= 0 + \frac{2}{5}(+1) = \frac{2}{5} = 0.4$$

$$E(student) = \frac{3}{5} I(0,3) + \frac{2}{5} I(2,0)$$

$$= 0.$$

$$E(credit) = \frac{3}{5} I(1,2) + \frac{2}{5} I(1,1)$$

$$= 3/5\left[\frac{-1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right] + \frac{2}{5}\left[\frac{-2}{2}\log_2\left(\frac{1}{2}\right)\right]$$

$$= 0.55 + 0.4 = 0.95$$

Gain (Income) = $0.97 - 0.4 = 0.57$

Gain (student) = $0.97 - 0 = 0.97$

Gain (credit) = $0.97 - 0.95 = 0.02$

Senior

$$I(yes, no) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = 0.97$$

$$E(income) = \frac{3}{5} I(2,1) + \frac{2}{5} I(1,1)$$

$$= \frac{3}{5}\left[-\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right] + \frac{2}{5}\left[-\frac{2}{2}\log_2\left(\frac{1}{2}\right)\right]$$

$$= 0.55 + 0.4$$

$$= 0.95$$

$$E(student) = \frac{3}{5}\left[-\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right] + \frac{2}{5}\left[-\frac{2}{2}\log_2\left(\frac{1}{2}\right)\right]$$

$$= 0.95$$

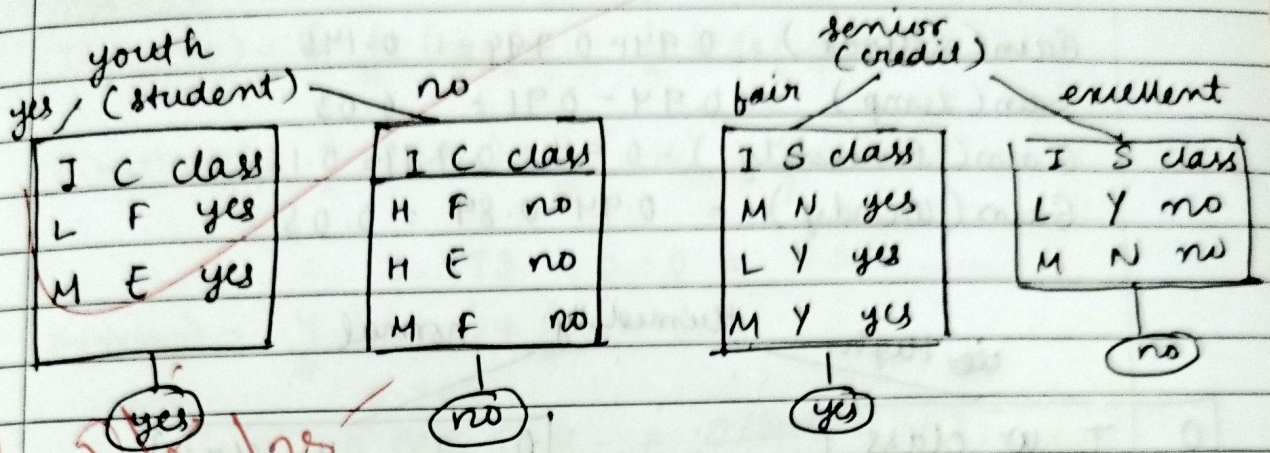
$$E(credit) = \frac{3}{5} I(3,0) + \frac{2}{5} I(0,2)$$

$$= 0$$

Gain (Income) = 0.02

Gain (student) = 0.02

Gain (credit) = 0.97



Outlook

$$I(yes, no) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right)$$

$$= 0.94$$

$$E(outlook) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2)$$

$$= 0.347 + 0 + 0.347$$

$$= 0.794$$

$$E(temp) = \frac{4}{14}I(2,2) + \frac{6}{14}I(4,2) + \frac{4}{14}I(3,1)$$

$$= 0.286 + 0.393 + 0.232$$

$$= 0.911$$

$$E(humidity) = \frac{7}{14}I(3,4) + \frac{7}{14}I(6,1)$$

$$= 0.493 + 0.296$$

$$= 0.789$$

$$E(windy) = \frac{8}{14}I(6,2) + \frac{6}{14}I(3,3)$$

$$= 0.464 + 0.428$$

$$= 0.892$$

Gain(outlook) = 0.94 - 0.794 = 0.146

Gain(temp) = 0.94 - 0.91 = 0.03

Gain(Humidity) = 0.94 - 0.789 = 0.151

Gain(windy) = 0.94 - 0.89 = 0.05

Humidity

~~yes~~ High

O	T	W	class
R	H	F	no
R	H	T	no
O	H	F	yes
S	M	F	yes
R	M	F	no
O	M	T	yes
S	M	T	no

normal

O	T	W	class
S	C	F	yes
S	C	T	no
O	C	T	yes
R	C	F	yes
S	M	F	yes
R	M	T	yes
O	H	F	yes

High

$$I(yes, no) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{4}{7}\right) = 0.98$$

$$E(Outlook) = \frac{2}{7}I(2,0) + \frac{3}{7}I(0,3) + \frac{2}{7}I(1,1)$$

$$= 0 + 0 + 0.286 = 0.286$$

$$E(Temp) = \frac{3}{7}I(1,2) + \frac{4}{7}I(2,2)$$

$$= 0.393 + 0.571 = 0.964$$

$$E(windy) = \frac{4}{7}I(2,2) + \frac{3}{7}I(1,2)$$

$$= 0.571 + 0.393 = 0.964$$

Gain(outlook) = 0.98 - 0.28 = 0.7

Gain(temp) = 0.98 - 0.96 = 0.02

Gain(windy) = 0.98 - 0.96 = 0.02

normal

$I(yes, no) = -\frac{6}{7}\log_2\left(\frac{6}{7}\right) - \frac{1}{7}\log_2\left(\frac{1}{7}\right) = 0.592$

$E(outlook) = \frac{3}{7} I(2,1) + \frac{2}{7}(2,0) + \frac{2}{7}(2,0)$

$= 0.393 + 0 + 0 = 0.393$

$E(temp) = \frac{4}{7} I(3,1) + \frac{3}{7}(3,0)$

$= 0.463$

$= 0.[illegible] + 0 = 0.463$

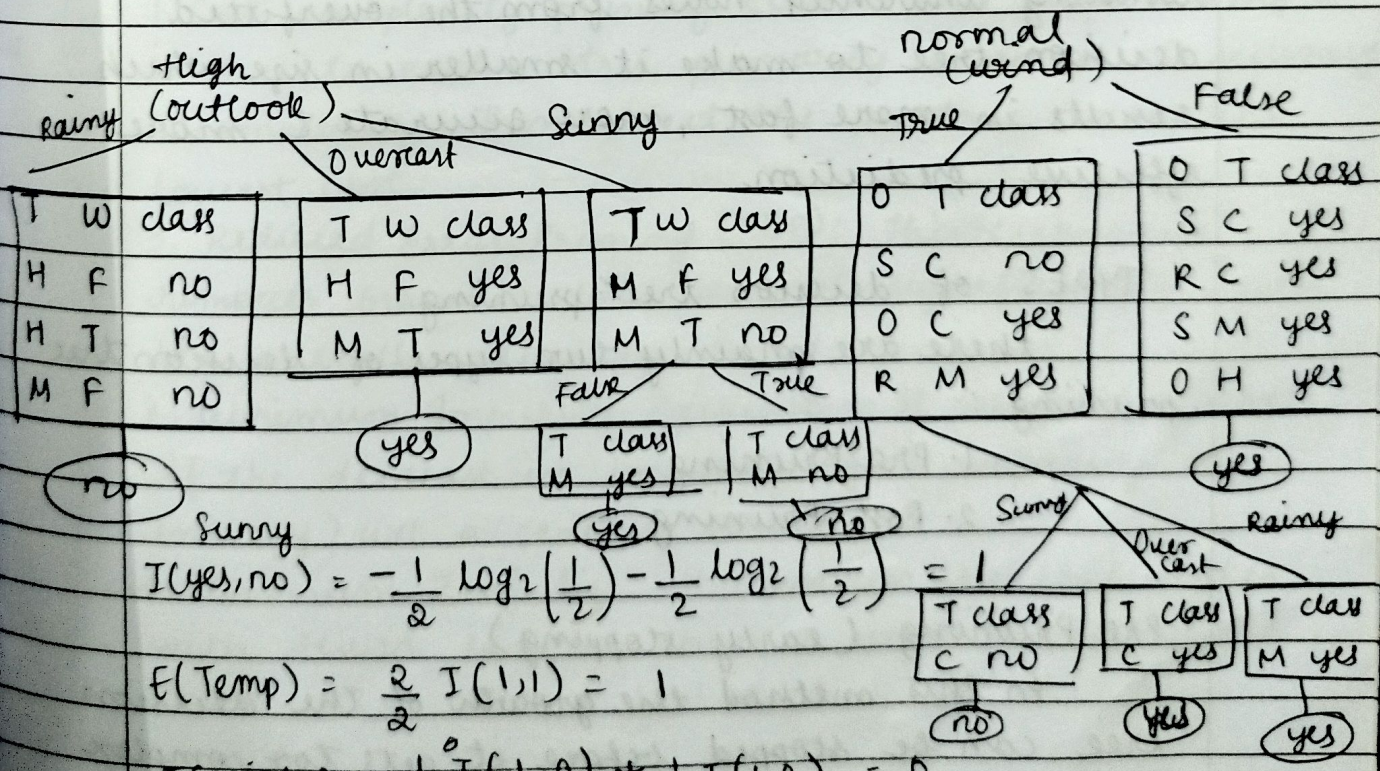
$E(windy) = \frac{4}{7} I(4,0) + \frac{3}{7}(2,1)$

$= 0 + 0.393 = 0.393$

Gain (outlook) = 0.592 − 0.393

Gain (temp) = 0.592 − 0.463

Gain (windy) = 0.592 − 0.[illegible]393



Sunny

$I(yes, no) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$

$E(Temp) = \frac{2}{2} I(1,1) = 1$

$E(windy) = \frac{1}{2} I(1,0) + \frac{1}{2} I(1,0) = 0$

Gain (Temp) = 0

Gain (windy) = 1

True

$$I(yes, no) = \frac{-2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) = 0.92$$

$$E(outlook) = \frac{1}{3}I(1,0) + \frac{1}{3}I(1,0) + \frac{1}{3}I(0,1) = 0$$

$$E(temp) = \frac{2}{3}I(1,1) + \frac{1}{3}I(1,0)$$

$$= 0.555$$

$$Gain(outlook) = 0.92 - 0 = 0.92$$

$$Gain(temp) = 0.92 - 0.55 = 0.37$$

Decision Tree Pruning

It is a technique used to prevent decision trees from overfitting the training ~~tree~~ data

It aims to simplify the decision tree by removing unwanted nodes from the overfitted decision tree to make it smaller in size which results in more fast, more accurate & more effective prediction

TYPES of decision tree pruning

there are mainly two types of decision tree pruning:

1. Pre Pruning
2. Post Pruning

Pre Pruning (early stopping)

In this method the growth of the decision tree can be stopped before it gets too complex while constructing the decision tree.

some common pre pruning techniques include:
- maximum depth (it limits the max level of depth in a decision tree).
- minimum samples per leaf (set a min threshold for the no. of samples in each leaf node)
- minimum samples per split (specify the minimal no. of samples needed to break a node)
- maximum features (it restricts the quantity of features considered for splitting)

POST Pruning (reducing nodes)

After the tree is fully grown post pruning involves removing branches or nodes to improve the models ability to generalise.

some common post pruning techniques include:
1. Cost complexity pruning (CCP): this method assigns a price to each subtree primarily based on its accuracy & complexity. Then select the subtree with the lowest cost.
2. Reduced Error Pruning (REP): This technique removes branches that do not significantly effect the overall accuracy.
3. Minimum Impurity Decrease: It prunes the nodes if the decrease in impurity (Gini impurity or entropy) wrt a certain threshold.
4. Minimum Leaf Size: It removes the leaf nodes with fewer samples than a specified threshold.

AUC - ROC Curve

It stands for Area under the Reciever Operating Characteristics curve.

It is a graphical representation of the performance of a binary classification model at various classification threshold.

It plots the true positive rate (TPR) vs the false positive rate (FPR) at different classification thresholds.

$$TPR = \frac{TP}{TP+FN} \qquad FPR = \frac{FP}{FP+TN}$$

A high AUC (close to 1) indicates the model is effective in classifying the samples between two class.

A low AUC (close to 0) suggests poor performance

In ROC curve the x axis typically represents false positive rate & y axis represents TPR.

Ex: Consider true levels of data points are [1, 0, 1, 0, 1, 1, 0, 0, 1, 0] predicted probabilities [0.8, 0.3, 0.6, 0.2, 0.7, 0.9, 0.4, 0.1, 0.95, 0.55]

Case 1: threshold = 0.5 , 0.7, 0.4, 0.2, 0.85

CaseI: 0.5 threshold.

predicted values:

[1, 1, 1, 1, 1, 1, 1, 1, 1, 0]

wrong hy
if value >= threshold then make it 1
if less than threshold then make it 0.

Confusion matrix

	1	0
1	5	0
0	4	1

$TPR = \frac{5}{5+0} = 1$ $FPR = \frac{4}{4+1} = \frac{4}{5} = 0.8$

Case II threshold = 0.7

predicted values : [1, 0, 0, 0, 1, 1, 0, 0, 1, 0] ✓

[1, 1, 0, 1, 1, 1, 1, 1, 1, 1] ✗

	1	0
1	4	1
0	5	0

$TPR = \frac{4}{1+4} = 0.8$ $FPR = \frac{5}{5} = 1$

Case III threshold = 0.4

predicted values

[1, 1, 1, 1, 1, 1, 0, 1, 1, 0]

	1	0
1	5	0
0	3	2

$TPR = \frac{5}{5} = 1$ $FPR = \frac{3}{3+2} = 0.6$

Case IV threshold = 0.2

predicted values

[1, 0, 1, 0, 1, 1, 0, 1, 1, 0]

	1	0
1	5	0
0	1	4

$TPR = \frac{5}{5} = 1$ $FPR = \frac{1}{5} = 0.2$

Case V threshold = 0.85

predicted values

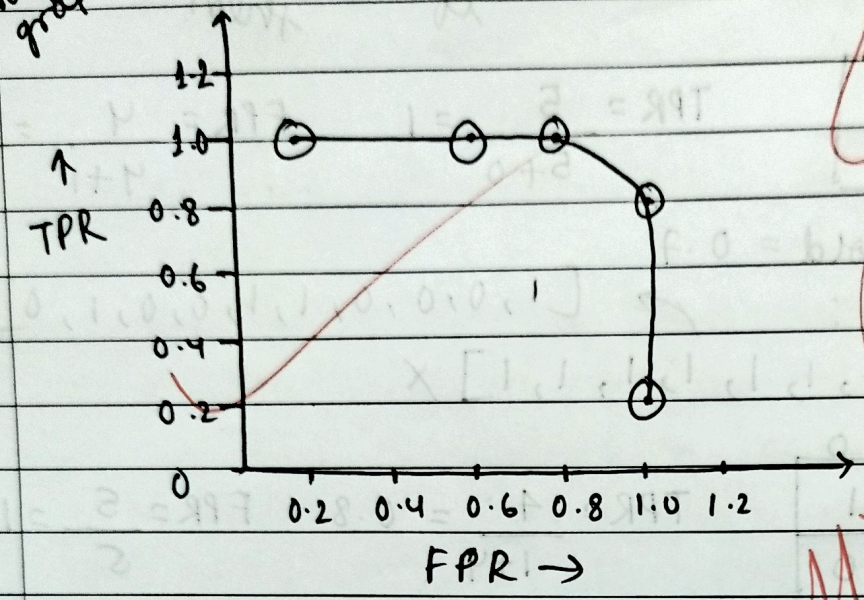
[0, 1, 0, 1, 0, 1, 1, 1, 0, 1]

	1	0
1	1	4
0	5	0

$TPR = \frac{1}{5} = 0.2$ $FPR = \frac{5}{5} = 1$

URBAN EDGE
COLORFUL YOU

(wrong graph)



28/01/25

Continous valued Attribute in DT

If any of the attribute of a dataset is continous then follow the below steps to calculate info gain in DT.

Example

Lets say the temp attribute is the continous value.

Temp:	72	60	48	40	90	80
class:	Y	Y	N	N	N	Y

step 1: sort the attribute values in ascending order.

Temp	40	48	60	72	80	90
class	N	N	Y	Y	Y	N

step 2: determine thresholds by averaging or finding mean by consecutive values where there is a change in classification.

threshold 1: $\frac{(48+60)}{2} = 54$ threshold 2: $\frac{80+90}{2} = 85$

step 3: consider threshold 1 i.e. 54 & proceed to calculate gain value

Temp →	C_1	C_1	C_2	C_2	C_2	C_2
class →	N	N	Y	Y	Y	N

$$I(3,3) = -\frac{3}{6}\log_2\left(\frac{3}{6}\right) - \frac{3}{6}\log_2\left(\frac{3}{6}\right)$$

$$= +1$$

$$E(temp) = \frac{2}{6} I(0,2) + \frac{4}{6} I(3,1)$$

$$= 0 + \frac{2}{3}\left[-\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right]$$

$$= 0.54$$

$$Gain(temp) = 1 - 0.54 = 0.46$$

entropy, E(whole dataset) = I(yes, no)

for threshold 2

temp	C_1	C_1	C_1	C_1	C_1	C_2
class	N	N	Y	Y	Y	N

$I(3,3) = 1$

$$E(\text{temp}) = \frac{5}{6} I(3,2) + \frac{1}{6} I(0,1)$$

$$= 0.81$$

$$E(\text{Gain}) = 1 - 0.81$$

$$= 0.19$$

step 4: Choose which has max information gain. Hence, we choose threshold 1

Tid	Refund	Marital	Taxine	Cheat
1	Yes	Single	125k	no
2	No	Married	100k	no
3	No	Single	70k	no
4	Yes	Married	120k	no
5	No	Divorced	95k	Yes
6	No	Married	60k	no
7	Yes	Divorced	220k	no
8	No	Single	85k	Yes
9	No	Married	75k	no
10	No	Single	90k	Yes

$$I(yes,no) = -\frac{3}{10}\log_2\left(\frac{3}{10}\right) - \frac{7}{10}\log_2\left(\frac{7}{10}\right)$$

$$= 0.881$$

$$E(Refund) = \frac{3}{10} I(0,3) + \frac{7}{10} I(3,4)$$

$$= 0 + \frac{7}{10} 0.295 = 0.295$$

$$E(Marital) = \frac{4}{10} I(2,2) + \frac{4}{10} I(0,4) + \frac{2}{10} I(1,1)$$

$$= 0.4 + 0 + 0.2 = 0.6$$

Taxable	60K	70K	75K	85K	90K	95K	100K	120K	125K	220K
class	no	no	no	yes	yes	yes	no	no	no	no

$$\text{Threshold 1} = \frac{75+85}{2} = 80 \qquad \text{Threshold 2} = \frac{95+100}{2} = 97.5$$

Threshold 1

Taxable	c_1	c_1	c_1	c_2	c_2	c_2	c_2	c_2	c_2	c_2
class	no	no	no	yes	yes	yes	no	no	no	no

$$E(\text{taxable}) = \frac{3}{10} I(0,3) + \frac{7}{10} I(3,4) = 0 + 0.295 = 0.295$$

Threshold 2

Taxable	c_1	c_1	c_1	c_1	c_1	c_1	c_2	c_2	c_2	c_2
class	no	no	no	yes	yes	yes	no	no	no	no

$$E(\text{taxable}) = \frac{9}{10} I(3,3) + \frac{1}{10} I(1,0) = 0.6 + 0 = 0.6$$

We choose threshold 1.

Gain (Refund) = 0.586

Gain (Marital) = 0.281

Gain (taxable) = 0.586

Refund

yes → no

M	T	class
S	125	no
M	120	no
D	220	no

no

M	T	class
M	100	no
S	70	no
D	95	yes
M	60	no
S	85	yes
M	75	no
S	90	yes

$$I(yes,no) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{4}{7}\right) = 0.985$$

$$E(Marital) = \frac{3}{7}I(0,3) + \frac{3}{7}I(2,1) + \frac{1}{7}(1,0) = 0.394$$

=

Taxfree	60	70	75	85	90	95	100
class	no	no	no	yes	yes	yes	no

threshold 1 = $\frac{75+85}{2} = 80$ threshold 2 = $\frac{95+100}{2} = 97.5$

thresh 1:

Taxfree	C_1	C_1	C_1	C_2	C_2	C_2	C_2
class	no	no	no	yes	yes	yes	no

$$E(Taxfree) = \frac{3}{7}I(0,3) + \frac{4}{7}I(3,1) = 0.464$$

thresh 2:

Taxfree	C_1	C_1	C_1	C_1	C_1	C_1	C_2
class	no	no	no	yes	yes	yes	no

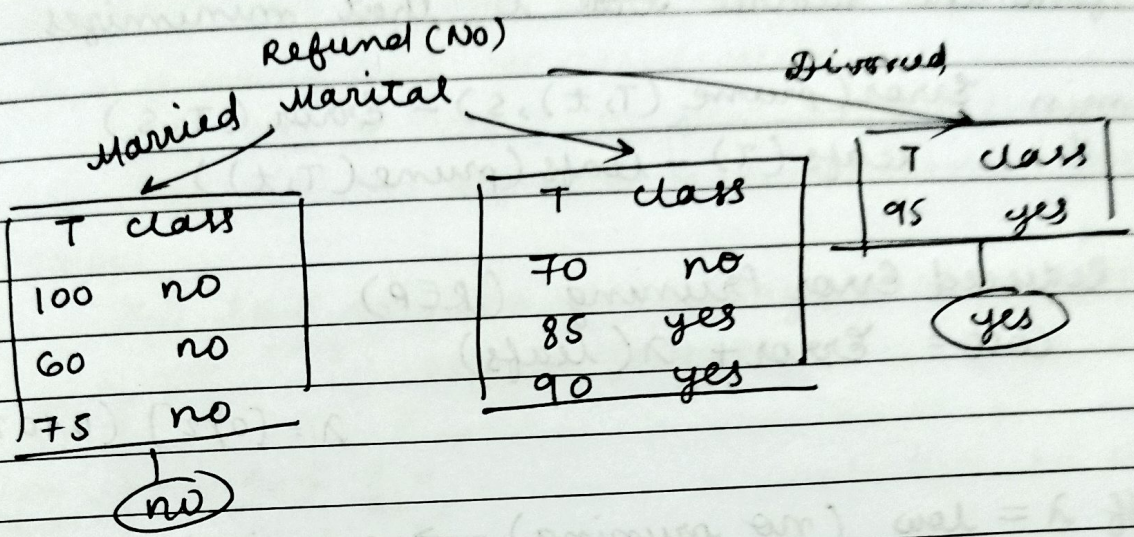
$$E(Taxfree) = \frac{6}{7}I(3,3) + \frac{1}{7}(0,1)$$

$$= 0.857$$

threshold 1 will be chosen

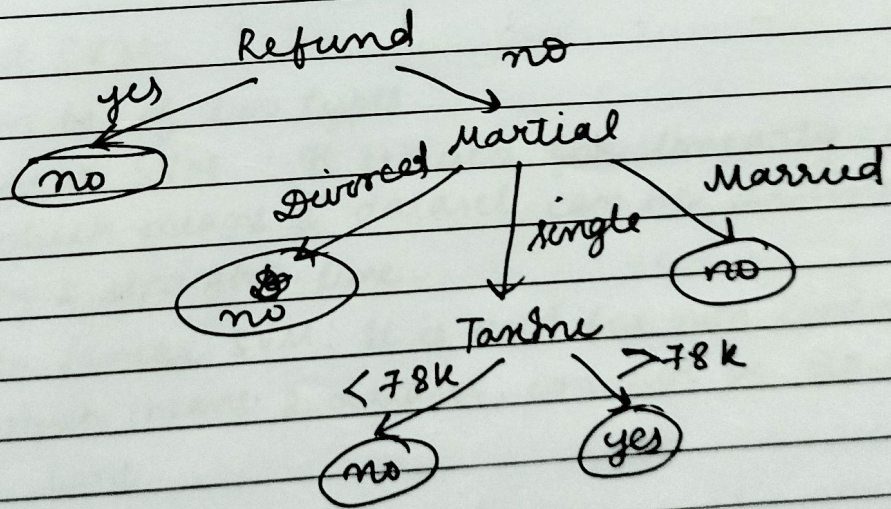
Gain(Marital) = 0.985 - 0.394 = 0.591

Gain(TaxInc) = 0.985 - 0.464 = 0.521



TaxInc	70	85	90	=	C_1	C_2	C_2
class	no	yes	yes		no	yes	yes

threshold 1: $\frac{70+85}{2} = 77.5$



Post Pruning

Cost Complexity Pruning (CCP)

find the subtree tree t that minimizes

$$\min_t \frac{Error(prune(T,t),S) - Error(T,S)}{leafs(T) - leafs(prune(T,t))}$$

Reduced Error Pruning (REP)

$$Cost = Error + \lambda(leafs)$$

$\lambda = [0, 2]$ (0 to 2)

If λ = low (no pruning) → overfitting

If λ = high (pruning) → good regularization, good generalised model

If λ = very high → underfitting

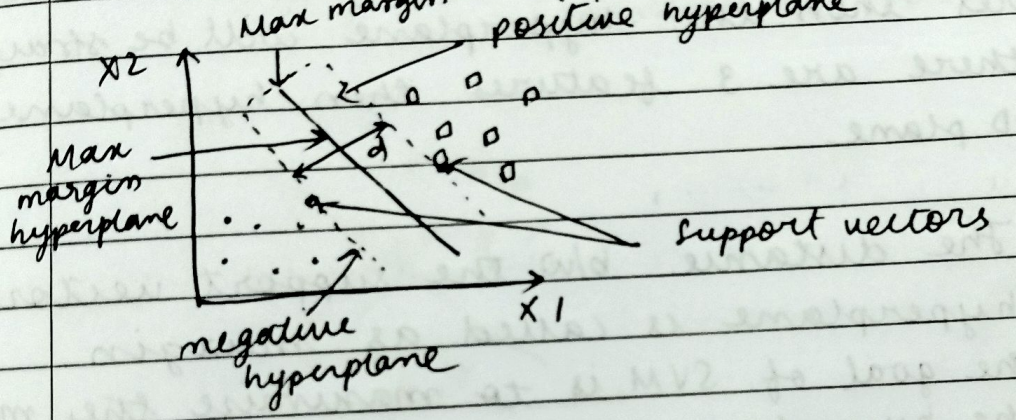
URBAN EDGE COLORFUL YOU

Support Vector Machine (SVM)

It is one of the most popular supervised learning algorithm used for both classification & regression. However it is generally used for classification.

The goal of SVM is to find the best line or decision boundary that can segregate n dimensional space into classes.

This best decision boundary is cased hyperplane



TYPES of SVM:

It can be of two types:

1. Linear SVM: It is used for linearly seperable data which means a dataset can be classified into two classes by a straight line.

2. non Linear SVM: It is used for non linearly seperable data which means a dataset can not be classified by straight line.

Terminologies of SVM:

Support vectors: The data points or vectors that are closest to the hyperplane & which affect the position of hyperplane are termed as support vector.

Hyperplane: There can be multiple decision boundaries possible to draw for segregating the classes in n dimensional space but a best decision boundary is required to classify the datapoints. This best boundary is known as hyperplane of SVM.

The dimension of the hyperplane depends on no. of features in the dataset. If there are two features then the hyperplane will be straight line. If there are 3 features then hyperplane will be 2-D plane

Margin: The distance b/w the support vectors and the hyperplane is called as margin

The goal of SVM is to maximize the margin

The hyperplane with the max. margin is called optimal hyperplane

Use of dot product in SVM:

Consider a random point x we need to classify whether x belongs to positive class (right side of hyperplane) or it belongs to negative class (left side of hyperplane). Consider the point x is a vector and then make a perpendicular line to the hyperplane i.e., w as $\vec{w}$. Lets say the distance of $\vec{w}$ from origin to the hyperplane is c.

Make a projection of $\vec{x}$ on $\vec{w}$. If the dot product of $\vec{x}$ & $\vec{w}$ > c then the point x lies on right side of hyperplane. Thus belongs to +ve class.

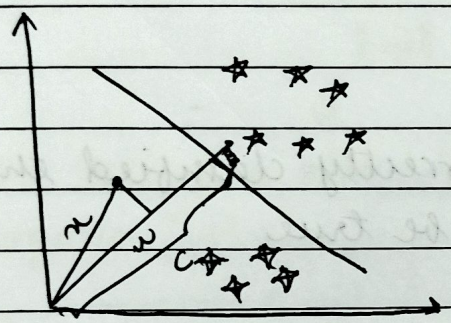
If its is less than c, it lies on the left side of hyperplane. Thus belongs to -ve class. If it is equals to c then it lies on the hyperplane itself.

Mathematically,

$\vec{x} \cdot \vec{w} = c$ (point lies on decision boundary)

$\vec{x} \cdot \vec{w} > c$ (positive samples)

$\vec{x} \cdot \vec{w} < c$ (negative samples)



Margin is SVM

To classify a point as -ve or +ve a decision rule need to be defined as

$$\vec{x} \cdot \vec{w} - c \geq 0$$

putting -c as b, we get

$$\vec{x} \cdot \vec{w} + b \geq 0$$

hence

$$y = \begin{cases} +1 & \vec{x} \cdot \vec{w} + b \geq 0 \\ -1 & \vec{x} \cdot \vec{w} + b < 0 \end{cases}$$

We need the value of ~~$\vec{w}$ & b~~ (w, b) such that margin has max distance. Lets say dist is d.

To calculate d, assume L1 is $w \cdot x + b = 1$

L2 is $w \cdot x + b = -1$

URBAN EDGE COLORFUL YOU

To calculate the distance 'd' in such a way that no positive or negative point can cross the margin line

For all the negative class points the eqn is:

$$\vec{w}.\vec{x} + b \leq -1$$

For all positive class points

$$\vec{w}.\vec{x} + b \geq +1$$

Considering two constraints into one

we assume that for negative class $y = -1$ & +ve class have $y = 1$

So,

$$y_i(\vec{w}.\vec{x} + b) \geq 1$$

for every point to be correctly classified the above condition should always be true

To maximize the 'd' such that $y_i(\vec{w}.\vec{x} + b) \geq 1$ holds true. To do that consider two support vectors 1 from negative class, i.e. x_1 & 1 from positive class i.e., x_2.

The distance b/w these two vectors x_1 & x_2 will be $(x_2 - x_1)$ vector.

To find out the shortest distance b/w these two points take a vector 'w' ⊥r to the hyperplane & find the projection of $(x_2 - x_1)$ vector on w

$$\frac{(x_2 - x_1).\vec{w}}{\|w\|} = \frac{x_2.\vec{w} - x_1.\vec{w}}{\|w\|} \quad \text{---(1)}$$

for +ve point $y = 1$

$\Rightarrow 1 \times (\vec{w} \cdot x_2 + b) = 1$

$\Rightarrow \vec{w} \cdot x_2 = 1 - b$ ——— (2)

for -ve point $y = -1$

$-1(\vec{w} \cdot x_1 + b) = 1$

$\Rightarrow \vec{w} \cdot x_1 = -b - 1$ ——— (3)

now substituting values of 2 & 3 in 1 we get

$$\frac{(1-b) - (-b-1)}{\|w\|} = \frac{1-b+b+1}{\|w\|}$$

$$= \frac{2}{\|w\|} = d.$$

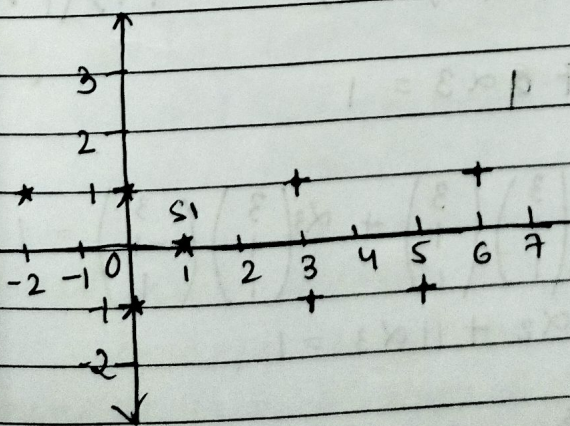
$$\text{argmax}\,(w^*, b^*)\ \frac{2}{\|w\|} \text{ such that } y_i(\vec{w} \cdot \vec{x} + b) \geq 1$$

↳ eqn to maximize

* Determine the ~~exam~~ eqn of hyperplane that divides the datapoints into two classes

positively labeled datapoints [(3,1), (3,-1), (6,1), (5,-1)]

negatively [(1,0), (-2,1), (0,-1), (0,1)]



→ support vector of -ve class

S1 (1,0)

If same dist we get two points, consider both

S2 (3,1) } S.V of +ve

S3 (3,-1) } class.

URBAN EDGE
COLORFUL YOU

$S1 = \begin{pmatrix}1\\0\end{pmatrix}$ $S2 = \begin{pmatrix}3\\1\end{pmatrix}$ $S3 = \begin{pmatrix}3\\-1\end{pmatrix}$

aggument 1 in all 3 support vectors as bias

$\bar{S1} = \begin{pmatrix}1\\0\\1\end{pmatrix}$ $\bar{S2} = \begin{pmatrix}3\\1\\1\end{pmatrix}$ $\bar{S3} = \begin{pmatrix}3\\-1\\1\end{pmatrix}$

Lets take three parameters $\alpha1, \alpha2$ & $\alpha3$ for S1, S2, S3
Calculate the weight vector, we need to go for the following eqs

$$\alpha1\,\bar{S1}\cdot\bar{S1} + \alpha2\,\bar{S2}\cdot\bar{S1} + \alpha3\,\bar{S3}\cdot\bar{S1} = -1$$

$$\alpha1\,\bar{S1}\,\bar{S2} + \alpha2\,\bar{S2}\cdot\bar{S2} + \alpha3\,\bar{S3}\cdot\bar{S2} = 1$$

$$\alpha1\,\bar{S1}\,\bar{S3} + \alpha2\,\bar{S2}\cdot\bar{S3} + \alpha3\,\bar{S3}\cdot\bar{S3} = 1$$

$$\alpha1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha2\begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha3\begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} = -1$$

$$= 2\alpha1 + 4\alpha2 + 4\alpha3 = -1$$

$$\alpha1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha2\begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha3\begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} = 1$$

$$= 4\alpha1 + 11\alpha2 + 9\alpha3 = 1$$

$$\alpha1\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha2\begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha3\begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} = 1$$

$$= 4\alpha1 + 9\alpha2 + 11\alpha3 = 1$$

$\alpha1 = -3.5$, $\alpha2 = 0.75$, $\alpha3 = 0.75$

$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 = 0$

1) $-\frac{9}{4}x_1 + \frac{1}{2}x_2 + \frac{1}{2} = 0$ 2) $-\frac{9}{4}x_1 + \frac{x_2}{2} = -\frac{1}{2}$

1) $\frac{9}{2}x_1 - \frac{1}{2}x_2 = 1$ 2) $4.5\,x_1 - x_2 = 1$

$\bar{w} = \sum \alpha_i \bar{s_i} = \alpha_1 \bar{s_1} + \alpha_2 \bar{s_2} + \alpha_3 \bar{s_3}$

1) $\bar{w} = -3.5\begin{pmatrix}1\\0\\1\end{pmatrix} + 0.75\begin{pmatrix}3\\1\\1\end{pmatrix} + 0.75\begin{pmatrix}3\\-1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\-2\end{pmatrix}$

equating it with hyperplane offset.

$y = wx + b$ where $w = \begin{pmatrix}1\\0\end{pmatrix}$ $b = -2$

1) $y = x - 2$.

if $x = \begin{pmatrix}5\\1\end{pmatrix}$ we get, $y = \begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}5\\1\end{pmatrix} - 2 = 5 - 2 = 3$

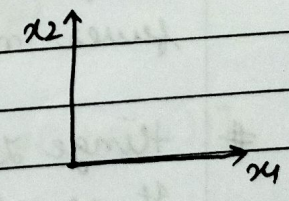
positive class.

If $w = \begin{pmatrix}1\\0\\-2\end{pmatrix}$ then $w = \begin{pmatrix}1\\0\end{pmatrix}$ $b = -2$

If $w = \begin{pmatrix}1\\0\end{pmatrix}$ line is parallel to x_2 axis

$\begin{pmatrix}0\\1\end{pmatrix}$ line is parallel to x_1 axis

$\begin{pmatrix}1\\1\end{pmatrix}$ line will make 45° wrt x_1 & x_2



Hard Margin linear SVM classifier

if data points are exactly linearly seperable then it is a problem of hard margin linear SVM. The derivation that you have made i.e. exactly hard margin linear SVM.

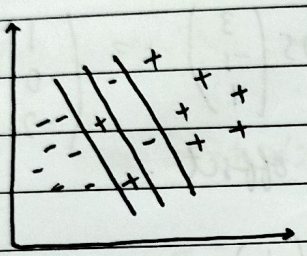
maximize $\text{argmax}(w^*, b^*) = \frac{2}{\|w\|}$ to convert this into

(if $f(x)$ is maximize, $1/f(x)$ is for minimize)

minimize $\text{argmax}(w^*, b^*) = \frac{\|w\|}{2}$

to simplify, minimize $\frac{\|w\|^2}{2}$ subject to $y_i(w^T x_i + b) \geq 1$

For soft margin linear SVM classifier



If the datapoints are almost linearly seperable thats why hard margin is not possible to draw. As some +ve points are on the side of -ve points & vice versa

regularization constant → C, slack variable → ξ_i

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \text{ subject to}$$

$$y_i(w^T x_i + b) \geq 1 - \xi_i \text{ & } \xi_i \geq 0 \text{ for } i = 1, 2, \ldots, m$$

here 'C' is the regularization term that balances margin maximization & the penalty for misclassification. A higher C value implies a stricter penalty for margin violation leading to a smaller margin but fewer misclassification.

Hinge Loss : It is a common loss funt in SVM. It penalizes misclassified points or margin violation and it is often combined with a regularization term in the objective funt.

When data contains outliers or is not perfectly seperable, SVM uses soft margin technique. This

method introduces a slack variable for each data point to allow some misclassification while balancing b/w maximizing the margin & minimizing violation.
summation of slack variable → hinge loss.

Logistic Regression

Loss fun^{n} $L = \frac{1}{N} \sum y_i \log \alpha_i + (1-y_i) \log (1-\alpha_i)$

$z_i = W x_i$

$\alpha_i = \sigma(z_i)$

$$P(y=1 | X) = h_\theta(X) = \frac{1}{1+e^{-\theta^T X}} = \frac{1}{1+e^{-z}}$$

where X: feature vector θ: weight vector

Likelihood function:

$$L(\theta) = \prod_{i=1}^{m} P(y_i | x_i, \theta)$$

$$\Rightarrow L(\theta) = \prod h_\theta(x_i)^{y_i} (1-h_\theta(x_i))^{(1-y_i)}$$

$$l(\theta) = \sum_{i=1}^{n} [y_i \log h_\theta(x_i) + (1-y_i) \log (1-h_\theta(x_i))]$$

Cost fun^{n}:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y_i \log h_\theta(x_i) + (1-y_i) \log (1-h_\theta(x_i))]$$

minimizing $J(\theta)$

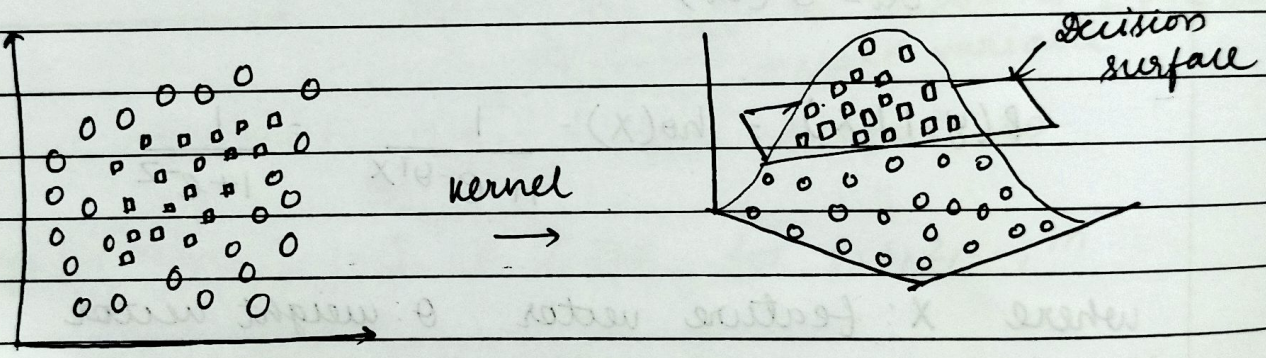
$$\frac{\partial J}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) x_{ij}$$

~~Derivation for Logistic Reg~~

$$\hat{y} = \begin{cases} 1, & h_\theta(x) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Non Linear SVM

When the datapoints are not linearly seperable we can use non linear SVM with kernel function to make it seperable with a decision hyperplane.



The kernel fun$^+$ allows the SVM to solve both linear and non linear problems by mapping data into a higher dimensional space where a hyperplane can seperate the data into classes.

Diff. types of Kernel Functions

- Polynomial kernel
- Sigmoid Kernel
- RBF Kernel
- Bessel function kernel

} non linear

Linear kernel Function : It is the simplest kernel that assumes that data is linearly seperable. It compules the dot product between two vectors.

$$K(x,y) = x^T y$$

$x = [1,2]$ $y = [3,4]$

$$K(x,y) = (1\times3) + 2(4) = 11$$

Polynomial :

$$K(x,y) = (x^T y + c)^d$$ → degree of polynomial

Radial Basis function (RBF) (Gaussian kernel)

$$K(x,y) = \exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right)$$

$\|x-y\|^2$ is sq. euclidean dist b/w vectors x & y

σ parameter controlling the spread.

$x : [1,2]$ $y [3,4]$ $\sigma = 1$ $\|x-y\| = \sqrt{(1-3)^2+(2-4)^2}$

$$K(x,y) = \exp\left(\frac{-8}{2}\right) \quad e^{-2} = 0.018$$

Sigmoid Kernel

$$K(x,y) = \tanh(\alpha x^T y + c)$$

α : slope parameter

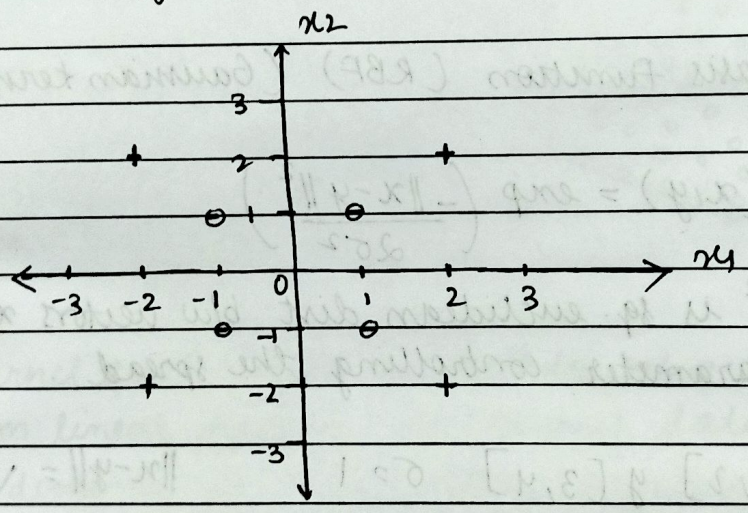
c : constant

$x = [1,2]$ $y = [3,4]$ $c = 1$ $\alpha = 0.5$

$$K(x,y) = \tanh(0.5(11)+1)$$
$$\tanh(6.5) = 0.99$$

Numerical example of non linear SVM.

Consider the positive labelled data points are:
$(2,2), (2,-2), (-2,2), (-2,-2)$
the negative labelled data points are:
$(1,1), (1,-1), (-1,1), (-1,-1)$.
Find out eqn of decision hyperplane that can be sole to classify these data points into two classes.



From the graph it is observed that a decision hyperplane can not be drawn to seperate these points. So we need to apply non linear SVM using a kernel fun^t that can transform the data points from one feature space to another feature space.

$$\phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{cases} \left.\begin{matrix} 4 - x_2 + |x_1 - x_2| \\ 4 - x_1 + |x_1 - x_2| \end{matrix}\right\} & \text{if } \sqrt{x_1^2 + x_2^2} \geq 2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \text{otherwise} \end{cases}$$

Transform the positive labelled datapoints using the kernel function.

$\phi\begin{pmatrix} x_1=2 \\ x_2=2 \end{pmatrix}$, $\sqrt{2^2+2^2} \geq 2$ $\begin{pmatrix} 4-2+|2-2| \\ 4-2+|2-2| \end{pmatrix}$

So,

$\phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$

$\phi\begin{pmatrix} 2 \\ -2 \end{pmatrix}$ $\sqrt{2^2+(-2)^2} \geq 2$

so,

$\phi\begin{pmatrix} 2 \\ -2 \end{pmatrix} = \begin{matrix} 4-(-2)+|2+2| \\ 4-2+|2+2| \end{matrix} = \begin{matrix} 6+4=10 \\ 2+4=6 \end{matrix} = \begin{pmatrix} 10 \\ 6 \end{pmatrix}$

$\phi\begin{pmatrix} -2 \\ 2 \end{pmatrix}$ $\sqrt{(2)^2+2^2} \geq 2$

$\phi\begin{pmatrix} -2 \\ 2 \end{pmatrix} = \begin{matrix} 4-2+|-2-2| \\ 4-(-2)+|-2-2| \end{matrix} = \begin{pmatrix} 6 \\ 10 \end{pmatrix}$

$\phi\begin{pmatrix} -2 \\ -2 \end{pmatrix}$ $\sqrt{(2)^2+(-2)^2} \geq 2$

$\phi\begin{pmatrix} -2 \\ -2 \end{pmatrix} = \begin{matrix} -4-(-2)+|-2+2| \\ 4-(-2)+|-2+2| \end{matrix} = \begin{pmatrix} 6 \\ 6 \end{pmatrix}$.

(2,2), (10,6), (6,10), (6,6)

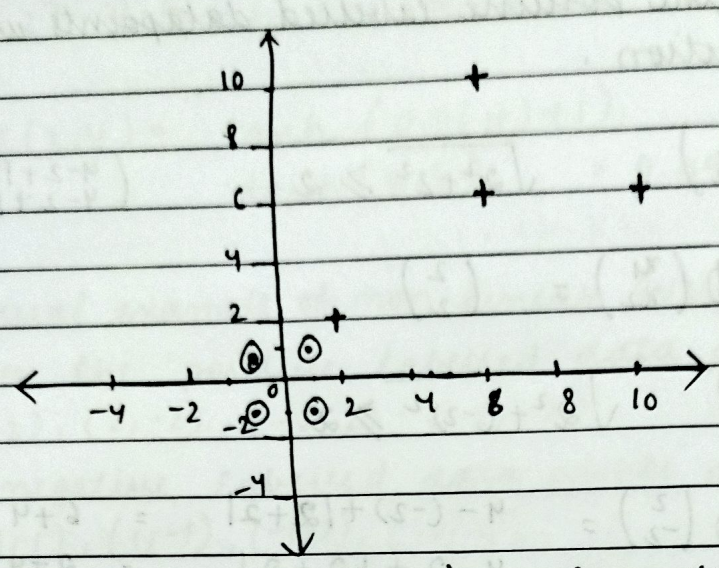
Transform the negatively labelled data points.

$\phi\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ = $\sqrt{(1)^2+(1)^2} \not\geq 2$

so,

$\phi\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

(1,1), (1,-1), (-1,1), (-1,-1).



$S_1 = (1,1)$ $S_2 = (2,2)$

from the graph it is obs that data points are linearly seperable. Apply linear SVM choose support vector of both class. Aggument bias to support vector.

$$\bar{S}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \bar{S}_2 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

$$\alpha_1 \bar{S}_1 \bar{S}_1 + \alpha_2 \bar{S}_2 \bar{S}_1 = -1$$
$$\alpha_1 \bar{S}_1 \bar{S}_2 + \alpha_2 \bar{S}_2 \bar{S}_2 = 1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -1$$

$$\Rightarrow \quad 3\alpha_1 + 5\alpha_2 = -1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} = 1$$

$$\Rightarrow \quad 5\alpha_1 + 9\alpha_2 = 1$$

1) $5\left(\frac{-1-5\alpha_2}{3}\right) + 9\alpha_2 = 1$

1) $-5 - 25\alpha_2 + 27\alpha_2 = 3$

2) $2\alpha_2 = 8$

3) $\alpha_2 = 4$ 3) $\alpha_1 = \frac{-1-20}{3} = -7$

$\alpha_2 = 4 \quad \alpha_1 = -7$

$$\bar{w} = \sum \alpha_i \bar{s}_i = \alpha_1 \bar{s}_1 + \alpha_2 \bar{s}_2$$

$$= -7\begin{pmatrix}1\\1\\1\end{pmatrix} + 4\begin{pmatrix}2\\2\\1\end{pmatrix} = \begin{pmatrix}1\\1\\-3\end{pmatrix}$$

$y = wx + b$ where $w = (1, 1) \quad b = -3$.

$$y = 2x - 3$$

PRIMAL DUAL PROBLEM

If it convex non linear quadratic optimization problem with inequality constraint then primal dual method is used by introducing langranian multiplier i.e.

$\alpha_i \geq 0$ for each constraint

$$L(w, w_0, \alpha) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + w_0))$$

To solve this find out derivative of this equation w.r.t w and w_0 & set them to 0.

$$\frac{\partial L}{\partial w} = w + \frac{d}{dw}\left[\sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i w^T x_i - \sum_{i=1}^{n} \alpha_i w_0 y_i\right]$$

$$\Rightarrow w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\frac{\partial L}{w_0} = 0 + \frac{\partial}{\partial w_0}\left[\sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i w^T x_i - \sum_{i=1}^{n} \alpha_i w_0 y_i\right]$$

1) $-\sum_{i=1}^{n} \alpha_i y_i = 0$

2 $\sum_{i=1}^{n} \alpha_i y_i = 0$

Now substitute w in langrenian. Lets calculate

$$\|w\|^2 = w^T w$$

$$= \left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T \left(\sum_{j=1}^{n} \alpha_j y_j x_j\right)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j^T$$

$$w^T x_i = \sum_{j=1}^{N} (\alpha_j y_j x_j)^T x_i$$

$$= \sum_{j=1}^{n} \alpha_i y_j x_j^T x_i$$

$$L(w, w_0, \alpha) = \frac{1}{2}\left[\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j^T\right] + \sum_{x=1}^{n} \alpha_i \left[1 - y_i \left\{ \sum_{j=1}^{n} \alpha_i y_i x_j^T x_i \right\} + w_0\right]$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j^T - \sum_{i=1}^{n} \alpha_i y_i w_0$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j^T$$

now we have to optimize by maximizing

$$\max_{\alpha} L(w, w_i) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j^T$$

such that $\alpha_i >= 0 \quad \forall i$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

The above problem is quadratic programming problem & can be solved using sequential optimization method developed by Platt.

PRODUCTION OF CLASS OF POINT

If x_j is support vector then $w^T x_j + w_0 = 1$ if $y_j = 1$

$$w^T x_j + w_0 = 1 \quad \text{if} \quad y_j = 1 \quad \text{---(1)}$$

$$w^T x_j + w_0 = -1 \quad y_j = (-1) \quad \text{---(2)}$$

Combining both we get,

$$w^T x_j + w_0 = y_j$$

$$w_0 = y_j - w^T x_j$$

$$w_0 = y_j - \sum_{i=1}^{n} \alpha_i y_i x_i^T x_j$$

(+ve class) $w^T x + w_0 > 0$

(-ve class) $w^T x + w_0 < 0$

} Put w^T & w_0 value in this.

Q Using SVM find the hyperplane with max margin for the following data

x_1	x_2	class
2	2	−1
4	5	+1
7	4	+1

Soln.

$f(x) = \vec{w} \cdot \vec{x} + b$ i = no. of support vectors

$\vec{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$

$\phi(\alpha) = \frac{1}{2} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j^T$

$= \sum_{i=1}^{3} \alpha_i - \frac{1}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} \alpha_i \alpha_j y_i y_j x_i x_j^T$

$\alpha_1 + \alpha_2 + \alpha_3 - 2\alpha_2 - 2\alpha_3 - 2\alpha_2 + \alpha_3$

$\sum_{i=1}^{3} \alpha_i y_i = 0$

1) $\alpha_1 y_1 + \alpha_2 y_2 + \alpha_3 y_3 = 0$ ($y_1 \rightarrow$ −ve class (−1)) ($y_2 \rightarrow$ +ve class (+1))

2) $-\alpha_1 + \alpha_2 + \alpha_3 = 0$

$\vec{x_1} \cdot \vec{x_1} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \end{pmatrix} = (2\ 2) \begin{pmatrix} 2 \\ 2 \end{pmatrix} = 8$

$\vec{x_1} \cdot \vec{x_2} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} (4\ 5) = 18$

$\vec{x_1} \cdot \vec{x_3} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} (7\ 4) = 22$

$\vec{x_2} \cdot \vec{x_1} = \begin{pmatrix} 4 \\ 5 \end{pmatrix} (2\ 2) = 18$

$\vec{x_2} \cdot \vec{x_2} = \begin{pmatrix} 4 \\ 5 \end{pmatrix} (4\ 5) = 41$

$\vec{x_2} \cdot \vec{x_3} = \begin{pmatrix} 4 \\ 5 \end{pmatrix} (7\ 4) = 48$

$\vec{x_3} \cdot \vec{x_1} = \begin{pmatrix} 7 \\ 4 \end{pmatrix} (2\ 2) = 22$

$\vec{x_3} \cdot \vec{x_2} = \begin{pmatrix} 7 \\ 4 \end{pmatrix} (4\ 5) = 48$

$\vec{x_3} \cdot \vec{x_3} = \begin{pmatrix} 7 \\ 4 \end{pmatrix} (7\ 4) = 65$

$\phi(x) = 0 = \frac{1}{2}(8+18+22+18+41+48+22$

$$\phi(\alpha) = \frac{-1}{2}\begin{bmatrix} 8\alpha_1^2 - 18\alpha_1\alpha_2 - 22\alpha_1\alpha_3 - 18\alpha_2\alpha_1 + 41\alpha_2^2 + 48\alpha_2\alpha_3 \\ -22\alpha_3\alpha_1 + 48\alpha_2\alpha_3 + 65\alpha_3^2 \end{bmatrix} + 2\alpha_1 + 2\alpha_3$$

$$= 2\alpha_1 - \frac{1}{2}[8\alpha_1^2 - 36\alpha_1\alpha_2 - 44\alpha_1\alpha_3 + 41\alpha_2^2 + 96\alpha_2\alpha_3 + 65\alpha_3^2]$$

$$= 2\alpha_2 + 2\alpha_3 - \frac{1}{2}[8\alpha_1^2 - 36(\alpha_2+\alpha_3)\alpha_2 - 44(\alpha_2+\alpha_3)\alpha_3 + 41\alpha_2^2 + 96\alpha_2\alpha_3 + 65\alpha_3^2]$$

$$= 2\alpha_2 + 2\alpha_3 - \frac{1}{2}[8\alpha_1^2 - 36\alpha_2^2 - 36\alpha_2\alpha_3 - 44\alpha_2\alpha_3 - 44\alpha_3^2 + 41\alpha_2^2 + 96\alpha_2\alpha_3 + 65\alpha_3^2]$$

$$= 2\alpha_2 + 2\alpha_3 - \frac{1}{2}[8\alpha_2^2 + 8\alpha_3^2 + 16\alpha_2\alpha_3 - 36\alpha_2^2 - 44\alpha_3^2 + 41\alpha_2^2 + 16\alpha_2\alpha_3 + 65\alpha_3^2]$$

$$= 2\alpha_2 + 2\alpha_3 - \frac{1}{2}[13\alpha_2^2 + 29\alpha_3^2 + 32\alpha_2\alpha_3]$$

• for $\phi(\alpha)$ to be maximum. we must have derivative wrt $\alpha_2 = 0$ & $\alpha_3 = 0$.

$$\frac{\partial\phi(\alpha)}{\partial\alpha_2} = 2 - \frac{1}{2}[26\alpha_2 + 32\alpha_3] = 0$$

$$\Rightarrow 26\alpha_2 + 32\alpha_3 = 4$$
$$\Rightarrow 13\alpha_2 + 16\alpha_3 = 2.$$

$$\frac{\partial\phi(\alpha)}{\partial x_3} = 2 - \frac{1}{2}[58\alpha_3 + 32\alpha_2] = 0$$

$$\Rightarrow 58\alpha_3 + 32\alpha_2 = 4$$
$$\Rightarrow 29\alpha_3 + 16\alpha_2 = 2$$

$\alpha_2 = 0.215$

$\alpha_3 = -0.05495$

(A)

URBAN EDGE
COLORFUL YOU

$\alpha_1 = \alpha_2 + \alpha_3 = 0.215 - 0.049$

$= 0.165$

$$w = \sum_{i=1}^{3} \alpha_i y_i x_i$$

$$= 0.165(-1)(2,2) + 0.215(1)(4,5) + -0.05(1)(7,4)$$

$$= \begin{pmatrix} -0.33 \\ -0.33 \end{pmatrix} + \begin{pmatrix} 0.86 \\ 1.075 \end{pmatrix} + \begin{pmatrix} 0.35 \\ 0.2 \end{pmatrix}$$

$$= \begin{pmatrix} 0.88 \\ 0.945 \end{pmatrix}$$

$$f(x) = \begin{pmatrix} 0.88 \\ 0.945 \end{pmatrix} \vec{x} + b$$

$$b = -\frac{1}{2}\left(\min_{y_i=+1}(\vec{w}\cdot\vec{x}_i) + \max_{y_i=-1}(\vec{w}\cdot\vec{x}_i)\right)$$

$$b = -\frac{1}{2}\left(\min\left(\begin{pmatrix} 0.88 \\ 0.945 \end{pmatrix}\begin{pmatrix} 4 \\ 5 \end{pmatrix}, \begin{pmatrix} 0.88 \\ 0.945 \end{pmatrix}\begin{pmatrix} 7 \\ 4 \end{pmatrix}\right) + \max\left(\begin{pmatrix} 0.88 \\ 0.945 \end{pmatrix}\right)\right)$$

$$= -\frac{1}{2}(11.895) = -5.94$$

$$\boxed{f(x) = 0.88x_1 + 0.945x_2 - 5.94}$$