★ IMP TERMINOLOGIES:

(1) Training and Testing
(2) Bias and Variance
(3) Overfitting and Underfitting
(4) Evaluation metric : Train / test accuracy, precision, recall, F1 score , Confusion matrix - TP, TN, FP, FN
(5) Complexity ⇒ No of parameters , categorical / numerical data, relation between features, size of dataset
(6) Curse of dimensionality : Dimension reduction by CNN, PCA , etc.

(Q) What are dif types of datasets used in ML?
(Q) How the validation & cross validation is dif from train and test accuracy?
(Q) Why is it imp to split data for ML algos to verify the performance?
(Q) What are the ethical considerations when one develops and deploys ML algos?
(Q) What are the issues related to privacy and data ethics?
(Q) What are the future trends of ML?
(Q) ~~Is~~ Does the ML domain impact industries and employment?
(Q) Bard Vs Gemini Pro Vs Gemini Ultra

- Reference book : Bishop (2nd one in syllabus)

> Objectives of ML:
- Process Data
- Analyze
- Conclude from data
- ~~Server~~ Sensible output

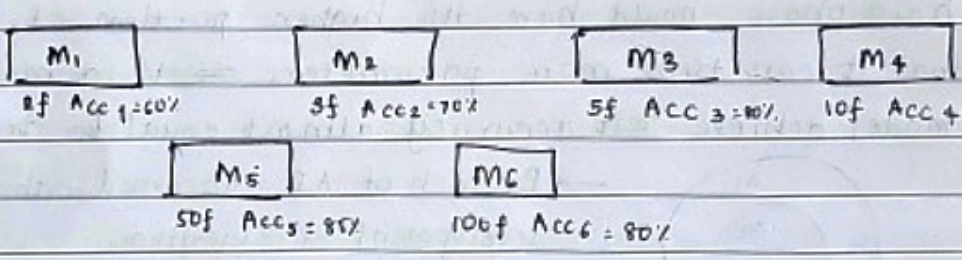
• Netflix uses ML for recommendations
• Facebook uses ML for targeted ads and auto tagging which

studies personal features using Deep Mind concept. (ML+NLP)

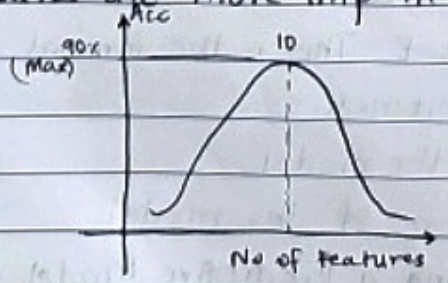
- Spam filtering: ML+NN Classification Problem
- Need for ML:
  - Increase in data generation
  - Improve decision making
  - Uncover patterns & trends.
  - Solve complex problems
- ML: Field of study that gives the computer capability to learn without being EXPLICITLY programmed. In ML you know both i/p and expected o/p. In traditional programming, ~~you ar~~ i/p generated output.
- Train phase and Test phase.
- Objective is to lessen error margin : Ideal o/p = actual o/p.
- Train phase should have the higher portion of samples so that it can tune more parameters ~~so t~~ to make the model achieve test accuracy almost equal to train accuracy.
- 

→ Branch of AI, concerned with design and development of algorithm.
→ ML is based on automation and acquired knowledge.
- ML: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves the experience E. This is the formal definition given by Samuel Tom M. Mitchell.
- Train phase ⇒ Building the model
  Test phase ⇒ Evaluation of the model
- ML process involves building a Predictive Model that can be used to find a soln for a Problem Statement.
  Define Objective → Data Gathering → Preparing data → Data Exploration → Building a model → Evaluation → Predictions

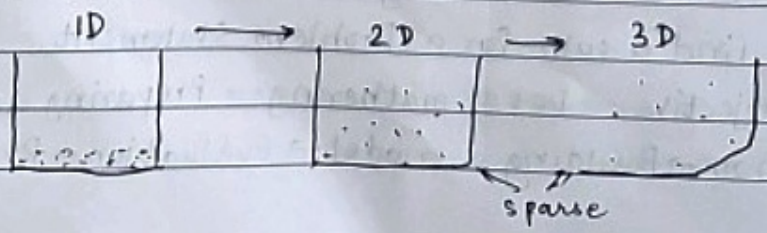
- Types of ML algos:
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
  - Supervised : Technique in which we teach ~~~~ or train the machine using data which is well-labelled.
  - Unsupervised. ~~Technique~~ Training of machine using info that is unlabeled and allowing the algo to act on that information without guidance.
  - Reinforcement: An agent is put in an environment and he learns to behave in this environment by performing certain actions & observing rewards for those actions.
  - Association Rule : Eg Market Basket Analysis
  - Curse of Dimensionality (COD)



Acc ⇒ Accuracy, f ⇒ features

- As features start increases beyond certain limit, accuracy will start decreasing rapidly. This is curse of dimensionality
- Overfitting occurs and model gets confused as to which features are more imp in prediction
- 

← Data having COD



- Data points get scattered as no of of features increases.
- Variance and bias increase.
- Underfitting: In a box of balls put an orange. If model is trained only on 2 f i.e, shape = spherical and color = any then orange will get identified as ball. This is underfitting.
- COD (result of) ⇒ Overfitting OR underfitting, not able to generalize (underlying pattern) ⇒ not utilize imp features.
- Objective of ML is keep the model as simple as possible but good prediction (not in cost of complexity and not to confuse the model). so COD generally misleads the model in predicting output.

– ML algo is a function.

$$y = f(x)$$

(y: dependant, x: independent variable)

– We get sweet pt (middle pt of over and underfitting) by regularization techniques

– When you give data (samples) for prediction to ML model, they try to find out underlying function $y=f(x)$ present in data.

– Parametric ML algo: Make assumptions about the function of data to give output. Eg: Linear regression. We assume relation between data is linear i.e, $y=mx+c$. So it is 1D with 2 parameters (m and c).

- No of parameters does not grow based on no of samples.
- Eg: Logistic regression, SVM, Naive Bayes, Simple NN.

– Non-parametric ML algo: No assumptions about the function of data. Eg: Decision Tree: Conditions on parameters are given wrt which decisions are made.

- No of parameters grow wrt no of samples
- Eg: Random forest, SVM with dif kernels, Complex NN, Xboot

Parametric	Non-Parametric
• Assumptions made about the function of dataset	• No assumptions about the function.
• No of parameter does not grow wrt no of samples	• No of parameters grow wrt no of samples
• Less data reqd to train the model.	• No assumption, so more data reqd for training.
• Due to assumptions, chances of mistakes, i.e. Underfitting.	• No assumption so we try to consider each data point which can lead to Overfitting
• Examples	• Examples.

- Linear Regression:

$$y = mx + c$$

DEP V, slope, IND V, intercept

slope ⇒ rate of change of y wrt x

intercept ⇒ value of y at x=0.

Eg.

square of Dev(x)	student score (x)	student score (y)	Mean ($\bar{x}$)	Mean ($\bar{y}$)	Dev(x) ($x_i - \bar{x}$)	Dev(y) ($y_i - \bar{y}$)	Prod of Dev(xy)	Sop of Dev
4	8	10	10	13	-2	-3	6	12
0	10	13			0	0	0	
4	12	16			+2	+3	6	
8								

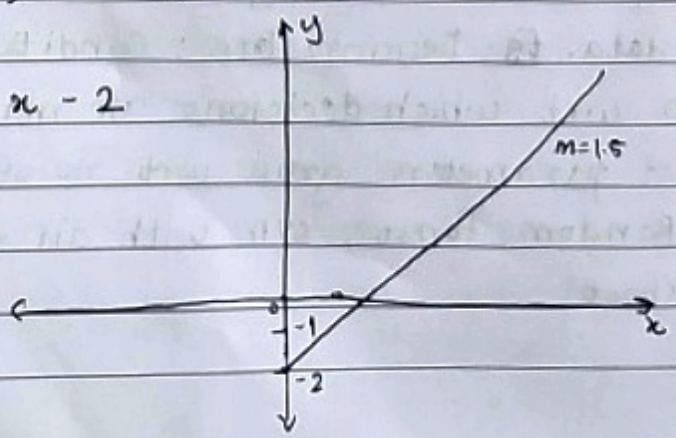
$$m = \frac{\text{Sum of product of Dev}}{\text{Sum of squares of Dev X}} = \frac{12}{8} = 1.5$$

$$b = \bar{y} - (m * \bar{x}) = 13 - (1.5 \times 10) = -2$$

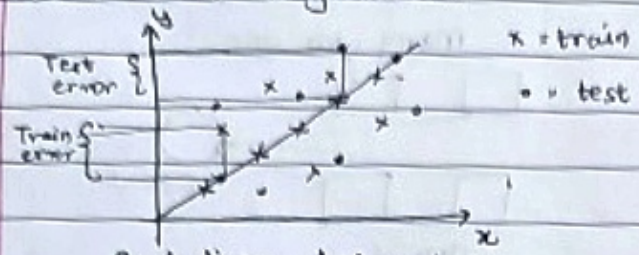
Eqn: $y = 1.5x - 2$

Eg: $x = 20$

$y = 30 - 2 = 28$



— Least error Regression.



Real-time dataset

It is the mathematical method used to find best fit line that represents relationship between dependent and independent variables in such a way that error is minimized

Eg:

Price of Tshirt(x)	Tshirt sold(y)	$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_{dev})(y_{dev})$	$y = mx + c$ (p)	Error y-p
2	4	-3.2	-4.2	13.44	3.3424	0.657
3	5	-2.2	-3.2	7.04	4.8604	0.139
5	7	-0.2	-1.2	0.24	7.8964	-0.896
7	10	1.8	1.8	3.24	10.9324	-0.932
9	15	3.8	6.8	25.84	13.9684	1.031
$\bar{x}=5.2$	$\bar{y}=8.2$			SOP = 49.8		$y_{actual} - y_{pred}$

$m = \frac{49.8}{32.8} = 1.518$

$b = 0.3064$

HW $X = 2\ 4\ 6\ 8$, $Y = 3\ 7\ 8\ 10$ } Both linear and least error.

- Multivariable Regression ⇒ more than one independent variable.

Eg: Restaurant Delivery System:

y : Total travel time (hr) — DEP V

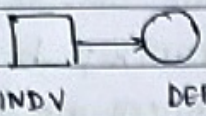
x_1 : Travel distance in miles } IND Vs

x_2 : No of deliveries

$\therefore y = f(x_1, x_2)$

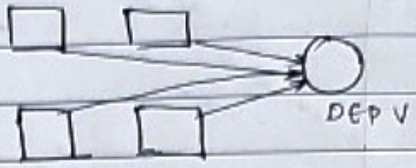
- Simple

One → One



Multivariable

Many to one



(Multicollineanity)

- Adding more no of variables (independent) in regression does not mean ~~ge~~ better performance, infact it can make the things coorse, i.e. overfitting.
- Adding more variables creates more relationships among them. IND Vs are not only potentially related to y, but also related to each other. ~~bd~~ This is called multicollinearity
- Ideally, all IND Vs should be corelated with DEP V and NOT with each other.

- For multivariable Regression:

(1) Simple Regression

(2) Scatter plots

(3) Correlation matrix.

- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n + \epsilon$ (↑ Error)

$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n$

- Interpretation of coeffs in multivariable Regression:

Eg: $y = 27 + 9x_1 + 12x_2$

→ Estimated change in y corresponding to one unit changein variable when all other variables remain constant.

(Q) Predict the sale of the product for which using IND V x_1 product 1 and IND V x_2 product 2.

P_1	P_2	Sale/week
x_1	x_2	y
1	4	1
2	5	6
3	8	8
4	2	12

bias due to overfitting

$$x = \begin{bmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{bmatrix}_{4\times 3} \qquad y = \begin{bmatrix} 1 \\ 6 \\ 8 \\ 12 \end{bmatrix}_{4\times 1}$$

$$\hat{a} = ((x^T x)^{-1} x^T) y \quad \text{where}$$

$$a = \begin{bmatrix} a_0 \\ a_1 \\ a_0 \end{bmatrix} \leftarrow \text{coeffs.}$$

1
16
25
64
4
109

$$x^T x = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{bmatrix}$$

$$(x^T x)^{-1} x^T = \begin{bmatrix} 3.153 & -0.59 & -0.3 \\ -0.59 & 0.205 & 0.016 \\ -0.3 & 0.016 & 0.055 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 1.36 & 0.47 & -1.02 & 0.19 \\ -0.319 & -0.098 & 0.1557 & 0.2622 \\ -0.065 & 0.0056 & 0.1857 & -0.125 \end{bmatrix}$$

$$\hat{a} = \begin{bmatrix} -1.619 \\ 3.4751 \\ -0.053 \end{bmatrix}$$

$$\hat{y} = a_0 + a_1 x_1 + a_2 x_2$$

$$\therefore \hat{y} = -1.619 + 3.475 x_1 - 0.053 x_2$$

> 2nd method of MVR

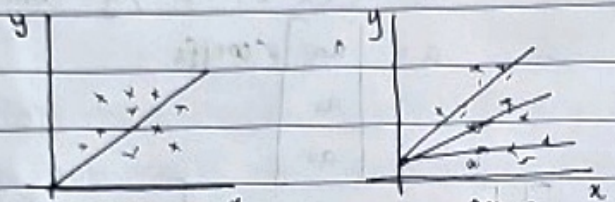
$y = a_0 + a_1x_1 + a_2x_2$

$a_0 = y - a_1x_1 - a_2x_2$

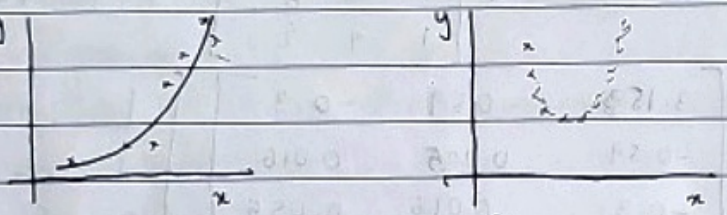
$a_1 = \dfrac{\Sigma(x_2)^2 * \Sigma(x_1y) - \Sigma(x_1x_2) * \Sigma(x_2y)}{\Sigma(x_1^2) * \Sigma(x_2^2) - (\Sigma(x_1x_2))^2}$

$a_2 = \dfrac{\Sigma(x_1)^2 * \Sigma(x_2y) - \Sigma(x_1x_2) * \Sigma(x_1y)}{\Sigma(x_1^2) * \Sigma(x_2^2) - (\Sigma(x_1x_2))^2}$

>



linear



$y = a_0 + \sum_{i=1}^{n} x_iy_i + FP$

non-linear

> ~~Polynomi~~

> $y = a_0 + \sum_{i=1}^{n} x_iy_i + FP$

linear — Polynomial function

> Non-linear to linear:

$y = ae^{bx}$

$\ln y = \ln(a) + \ln(e^{bx})$

$= \ln(a) + bx \ln(e)$ ($\ln(e) = 1$)

$\therefore \boxed{\ln y = \ln(a) + bx}$

> $y = a_0 + a_1x + a_2x^2$

$a = x^{-1}B$ $\quad X = \begin{bmatrix} n & \Sigma x_i & \Sigma x_i^2 \\ \Sigma x_i & \Sigma x_i^2 & \Sigma x_i^3 \\ \Sigma x_i^2 & \Sigma x_i^3 & \Sigma x_i^4 \end{bmatrix}$

$$B = \begin{bmatrix} \Sigma y_i \\ \Sigma x_i y_i \\ \Sigma x_i^2 y_i \end{bmatrix}$$

n = no of samples.

(Q)

X	Y	x_i^2	x_i^3	x_i^4	$x_i y_i$	$x_i^2 y_i$
1	1	1	1	1	1	1
2	4	4	8	16	8	16
3	9	9	27	81	27	81
4	15	16	64	256	60	240
$\Sigma x_i = 10$	$\Sigma y_i = 29$	30	100	354	96	338

$$X = \begin{bmatrix} 4 & 10 & 30 \\ 10 & 30 & 100 \\ 30 & 100 & 354 \end{bmatrix} \qquad B = \begin{bmatrix} 29 \\ 96 \\ 338 \end{bmatrix}$$

$$X^{-1}B = \begin{bmatrix} 7.75 & -6.75 & 1.25 \\ -6.75 & 6.45 & -1.25 \\ 1.25 & -1.25 & 0.25 \end{bmatrix} \begin{bmatrix} 29 \\ 96 \\ 338 \end{bmatrix} = \begin{bmatrix} 11215 \\ 36970 \\ 13022 \end{bmatrix}$$

$$= \begin{bmatrix} -0.75 \\ 0.95 \\ 0.75 \end{bmatrix}$$

$$y = -0.75 + 0.95x + 0.75x^2$$

> Model Selection: Depends on type of dataset and type of task. These are based on logical reasoning and comparison between ML models.

- Type of data:

(1) Supervised:

- Continuous, linear → SLR, MVR, SVM
- Continuous, non-linear → Poly^n R, Non-linear SVM
- Categorical → logistic R, decision tree, RF

(2) Unsupervised: K-means, H-Cluster.

- Type of Task:

(1) Regression: Linear or Non-linear

(2) Classification: Binary or Multi-~~task~~class

(3) Clustering

- Bias-Variance Trade off(?)

(1) Bias error } ~~reliable~~ reducible

(2) Variance error } i.e, can minimize

(3) Noise - unreliable, cannot minimize

- Bias is dif between ~~the~~ predicted and actual data points

$y = f(x) + e$

↳ irreducible error

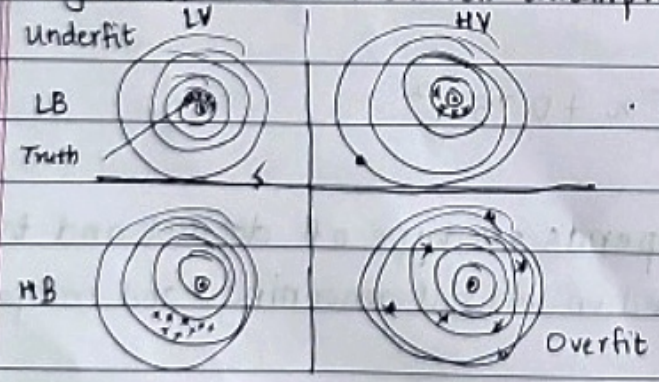
$f'(x) = p \quad A = f(x)$

Bias $f'(x) = E\ [f'(x) - f(x)]$

- Variance is when model takes into account the fluctuations in the data, i.e, noise as well.

Variance $f(x) = E\ [x^2] - E\ (x)^2$

- High bias → simple assumptions → Underfitting
- High variance → Too much assumption → Overfitting
- Underfit

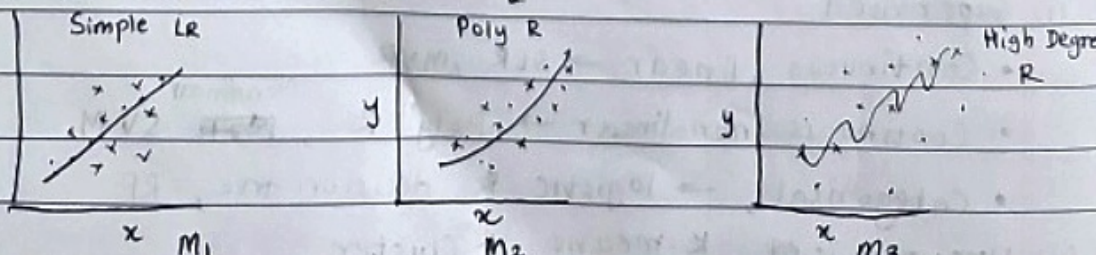


x - train

• - ~~test~~ truth value

Bullseye diagram

- Bias-Variance Trade-off

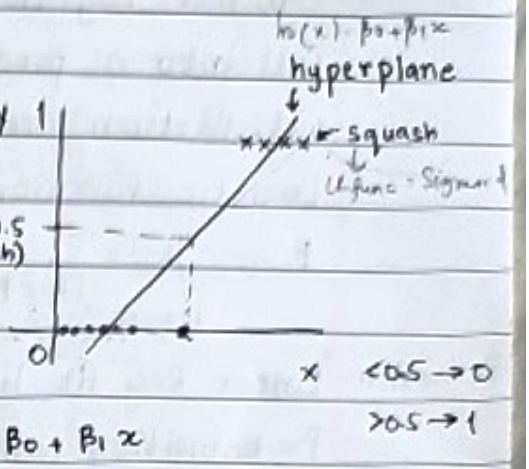


High bias Low variance | Moderate Bias & Variance | Low Bias High Variance

Total error = $Bias^2$ + Variance + IR error

X	y
No of hr study	Result P/F
3	fail
4	fail
6	pass
2	fail
5	pass

Categorical Binary P/F



Adding data points may lead to outliers leading to misinterpretation of data.

- When you apply Linear Regression on the categorical output of the data points, it suffers with two problems.
  (1) When you add more data pts, it faces the problem of outliers. Misclassification may occur.
  (2) Possibility of predicted data pts maybe greater than 1 or less than 0.

  In order to overcome these, we switch to Logistic Regression.
- hyperplane $h_\theta(x) = \beta_0 + \beta_1 x$
  squash ⇒ Logistic Regression function = $1/1+e^{-y}$ (Sigmoid)
- Logistic Regression is ~~an~~ the appropriate regression to conduct when dependent variable is categorical / binary. It is predictive analysis. It is used to describe data & explain relationship between one DEP binary 1 and one or more nominal / ordinal / interval / ratio level IND V.
- Binary → 0/1
- Multinomial logistic : Used to predict probability of one of three or more possible outcomes. Eg: Type of product a customer will buy, rating a customer will give a product. Eg: Political party a person will vote for.

> Logistic Regression is linear in nature as we use $y = \beta_0 + \beta_1 x$ to get the hyperplane. This hyperplane is in 'S' shape due to use of sigmoid function for calculating probability.

- Ordinal Logistic: Predicts probability of an outcome that fall into a predefined order such as level of customer satisfaction, severity of disease, or stages of cancer.
- Logistic function:

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

P ⇒ Probability which indicates a class. (as prob lies between 0,1)

Derivation: Linear Reg fit line : $y = \beta_0 + \beta_1 x$, we can't directly replace y by P. Probabilities can be predicted ~~by~~ using "ODDs" that is ratio of prob of success ~~upon~~ and prob of failure.

$$\therefore \frac{P}{1-P} = \beta_0 + \beta_1 x \quad \text{—(1)}$$

(reduces no of data pts ⇒ decrease in correlation)

$\frac{P}{1-P}$ ← always +ve $[0, \infty]$, but this is restricted range.

Take log to get high correlation.

$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x \quad \text{—(2)} \qquad \text{Range : } (-\infty, \infty)$$

But we are not interested in log, but exponent value to predict probability

$$\therefore e^{\ln(P/1-P)} = e^{(\beta_0 + \beta_1 x)}$$

$$\therefore \frac{P}{1-P} = e^{(\beta_0 + \beta_1 x)}$$

$$\therefore P = e^{(\beta_0 + \beta_1 x)} - P e^{(\beta_0 + \beta_1 x)}$$

$$P = P\left(\frac{e^{(\beta_0 + \beta_1 x)}}{P} - e^{(\beta_0 + \beta_1 x)}\right)$$

$$\therefore 1 = \frac{e^{(\beta_0 + \beta_1 x)}}{P} - e^{(\beta_0 + \beta_1 x)}$$

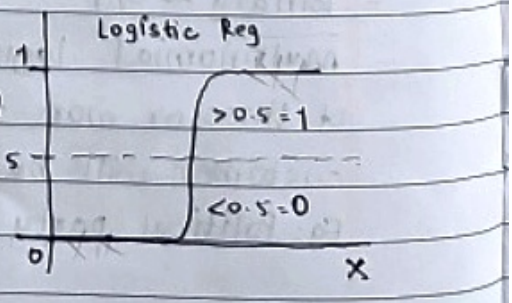
$$\therefore 1 - \frac{e^{(\beta_0 + \beta_1 x)}}{P} = -e^{(\beta_0 + \beta_1 x)}$$

$$\therefore P\left[1 + e^{(\beta_0 + \beta_1 x)}\right] = e^{(\beta_0 + \beta_1 x)}$$

$$\therefore P = \frac{e^{(\beta_0 + \beta_1 x)}}{1 + e^{(\beta_0 + \beta_1 x)}} \quad \text{—(3)}$$

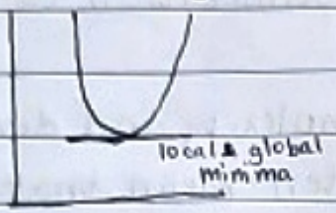
Divide by $e^{(\beta_0 + \beta_1 x)}$

$$\therefore \boxed{P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}}$$

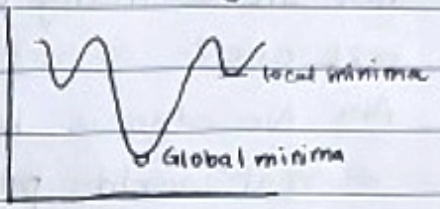


- You cannot classify points which lie exactly on threshold, but it is a very rare case so it is negligible.
- Cost function ⇒ Dif between predicted and actual

Linear Reg



Logistic Reg



Cost function

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

Cost function

$$\hat{y} = \frac{1}{1+e^{-x}}$$

~~log loss~~ : $\frac{1}{N} \sum_{i=1}^{N}$

- Log loss for logistic $= \frac{1}{N} \sum_{i=1}^{N} \underbrace{(y_i * \log(\hat{y_i}) + (1-y_i) * \log(1-\hat{y_i}))}_{\text{maximum likelihood estimation}}$

- On the given dataset apply logistic reg.

X (hrs ~~study~~)	Y (~~B~~ W/L)
29	0
15	0
33	1
28	1
40	1

(1) Calculate Probability of winning the game for player ~~to~~ who practices 28 hrs.

(2) Atleast how many hrs, player should practice that ~~me~~ makes him win the game with P > 85%.

Assume log ODDs log (odds) : $-64 + 2*hr$.

(1) $\therefore \log(odds) = -64 + 2*hr = \beta_0 + \beta_1 x$

$\beta_0 = -64$, $\beta_1 = 2$

$$P = \frac{1}{1+e^{-(-64+2x)}}$$ Put $x = 28$

$$= \frac{1}{1+e^{64-56}} = \frac{1}{1+e^{8}} = \text{~~0.0003.35~~ } 0.0003.35 \Rightarrow \boxed{0.033\%}$$

(2) $\frac{85}{100} < \frac{1}{1+e^{(64-2x)}} \Rightarrow 1+e^{(64-2x)} < \frac{100}{85}$

$\therefore e^{(64-2x)} < 0.17647 \Rightarrow 64-2x < -1.735$

$2x = 65.735$

$\therefore \boxed{x = 33 \text{ hrs}}$

∴ Player should practice for atleast 32.86 ~ 33 hrs so that he will win game with P of 85%.

- No Freelunch Rule:

There are so many algos for optimization so what is the best one?

Ans: No algo is best as complexity and diversity of real world problems often mean that some problems are easier to solve whereas others can be extremely difficult to solve. There is no one model, that works for every problem. Assumptions of great model for one problem may not hold for others. Thus, NFL states that no 2 algos are equivalent when their performance is average across all possible problems. Mathematically NFL theorem: If an algo performs well on certain class of problems, then it necessarily pays for that with degraded performance on the set of all remaining problems. In summary, there is no single best learning method.

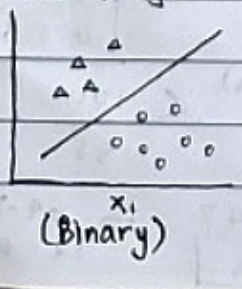
(Q) Take NN with many layers optimized by Back Propogation i.e deep learning. Choose correct option:

(1) Deep learning performs better than most other algos for real-world problems.

(2) Deep learning can fit everything.

(3) Deep learning performs better than other algos for all problems.

- Logistic Regression (Multiclass)



It is simple. Variables are categorical.
Confusion matrix → derivation is simple
Eg: Spam / not spam

- Multiclass → No of instances in dataset predicts no of classifiers.

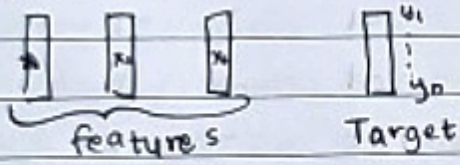
3 classes

This is complex. Confusion matrix is complex. Variables are categorical.
Eg: Classification of fruits

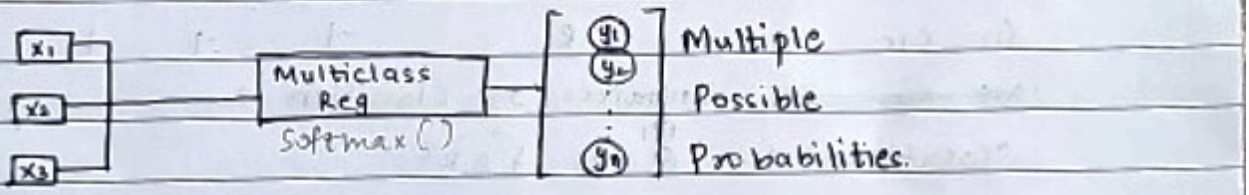
for one to all (rest) ⇒ no of classifiers = N
one to one ⇒ $\frac{N*(N-1)}{2}$ classifiers
} Both can be used

N → No of instances

- Multiclass



features Target ⇒ k possible target classes



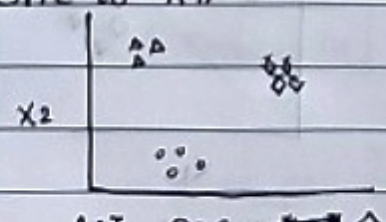
LR^n : $y = mx + c$

$Z_{i(k+1)} = W_{(k \times m)} \cdot X_{(m \times 1)} + B_{(k+1)}$ ← Bias

wt. associated with feature

$$\begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \vdots & & & \\ w_{n,0} & \dots & & w_{n,n} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} B_0 \\ B_1 \\ \vdots \\ B_n \end{bmatrix}$$

- One to All



Instances = 3 ⇒ Classifiers = 3

Classifier 1 : T vs {C, D} = y_1

2 : C vs {T, D} = y_2

3 : D vs {T, C} = y_3

*

- Then we apply softmax function to classify new test data pts into T, C, D classes.

  $y_1 \Rightarrow P_1$
  $y_2 \Rightarrow P_2$
  $y_3 \Rightarrow P_3$

  } We will focus on the Probabilities and assign class which has majority +ve count.

- $h_\theta^{(i)}(x) = P(y=i \mid x;\theta)$ $(i = 1,2,2)$

(Q)

Features			Classes	y_1	y_2	y_3
x_1	x_2	x_3	G	1	-1	-1
x_4	x_5	x_8	B	-1	1	-1
x_7	x_6	x_{12}	R	-1	-1	1
x_{10}	x_9	x_{14}	G	1	-1	-1
x_{13}	x_{11}	x_{16}	B	-1	1	-1
x_{17}	x_{15}	x_{21}	R	-1	-1	1
~~x_{18}~~	~~x_{20}~~	~~x_{19}~~				

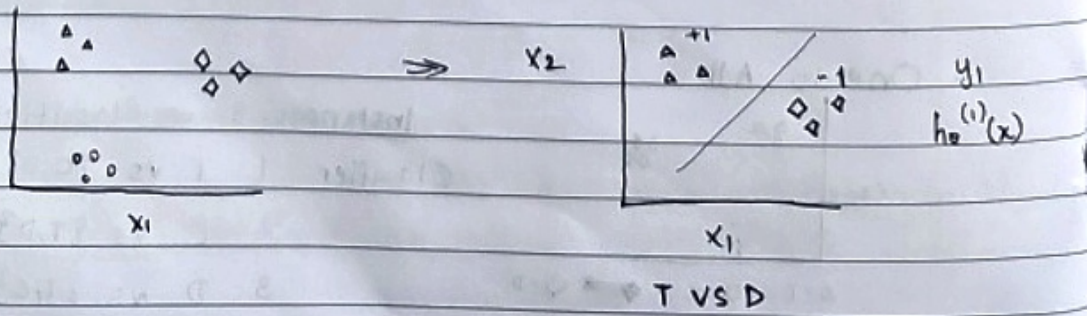
Instances = 3 Classifiers = 3.

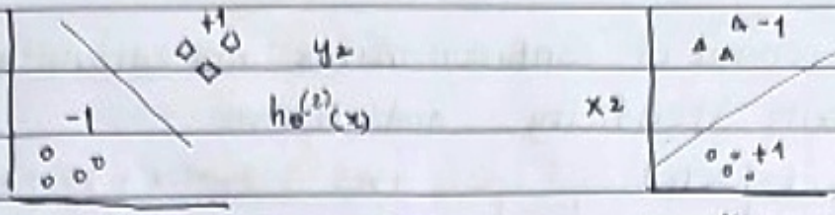
Classifier: 1 : G (+1) vs {B, R} (-1, -1)
2 : B vs {G, R}
3 : R vs {G, B}

Softmax function will give probs P_1, P_2, P_3.

$P_1 = 0.8$, $P_2 = 0.7$, $P_3 = -0.9$ lets say this is the case for one data pt, then ~~that~~ we take highest +ve $\Rightarrow P_1$ implying it belongs to G class.

- One to One





(Q) Derive the cost function of multiclass Logistic Reg & compare with Binary logistic Reg cost function.

(Q) Compare.

Binary Logistic	Multiclass Logistic
• Sigmoid / logistic function is used to get the probabilities ~~that for~~ back from response variable y.	• Softmax function is used to get probabilities back from response variable y.
• Prob of +ve class is given as: $P = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$ Prob of -ve class is (1-P).	• Prob of class i, given k classes, is: $\sigma(x)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$ for $i = 1,\dots,k$ & $z = (z_1 \dots z_k) \in R^k$
• $y = \beta_0 + \beta_1 * \text{feature\_1} + \beta_2 * \text{feature\_2} + \dots$ One β is learnt per feature.	• $y_i = \beta_0 + \beta_1 * f\_1 + \beta_2 * f_2 t \dots$ One β is learnt per feature per class. Every class has its own hyper plane.
• Eg: Pass/Fail, spam/not spam	• Eg: Fruits, colors.

(Q)

	Expected	
predicted	45 TP	8 FP
	15 FN	32 TN

⇒ 60 samples → +ve, 40 samples → -ve } actual

Total 100 samples.

(Q) Apply concept of confusion matrix and calculate accuracy, sensitivity, specificity and F1 score.

	Expected			
	C_1	C_2	C_3	C_4
C_1	52	3	7	2
Predicted C_2	2	28	2	0
C_3	5	2	25	12
C_4	1	1	9	40

+ves = 52 + 28 + 25 + 40 = 145

Missclassified = 3+7+2+2+2+5+2+12+1+1+9 = 46

Total = 191 samples

Accuracy = $\frac{145}{191} \times 100 = 75.9\%$

Accuracy = $\frac{TP + TN}{All}$

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

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

F1 score = $2 \frac{Precision \times Recall}{Precision + Recall}$

~~Specificity~~ Precision $C_1 = \frac{52}{52+3+7+2} = 0.8125$

$C_2 = \frac{28}{28+2+2} = 0.875$

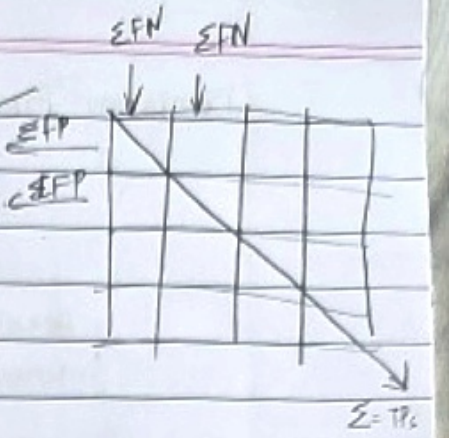
$C_3 = \frac{25}{25+5+2+12} = 0.568$

$C_4 = \frac{40}{40+9+1+1} = 0.784$

Recall $C_1 = \frac{52}{60} = 0.867$ $C_2 = \frac{28}{34} = 0.823$

$C_3 = \frac{25}{43} = 0.582$ $C_4 = \frac{40}{54} = 0.74$

$C_{1F1} = \frac{2(0.8125 \times 0.867)}{(0.8125 + 0.867)} = 0.8388$



$C_2F_1 = 2\frac{(0.875 \times 0.823)}{(0.875 + 0.823)} = 0.848$

$C_3F_1 = 2\frac{(0.568 \times 0.582)}{(0.568 + 0.582)} = 0.575$

$C_4F_1 = 2\frac{(0.784 \times 0.74)}{(0.784 + 0.74)} = 0.7614$

$$\text{Precision} = \frac{TP_1 + TP_2 + TP_3 + TP_4}{TP_1 + FP_1 + TP_2 + FP_2 + TP_3 + FP_3 + TP_4 + FP_4}$$

$$= \frac{52 + 28 + 25 + 40}{52 + 12 + 28 + 4 + 25 + 19 + 40 + 11} = 0.759$$

$$\text{Recall} = \frac{TP_1 + TP_2 + TP_3 + TP_4}{TP_1 + FN_1 + TP_2 + FN_2 + \cdots}$$

$$= \frac{52 + 28 + 25 + 40}{52 + 8 + 28 + 6 + 25 + 18 + 40 + 14} = 0.759$$

$$F1_{score} = \frac{2(0.759 \times 0.759)}{2(0.759)} = 0.759$$

(8) What factors can contribute to the popularity of logistic ~~regis~~ Reg?

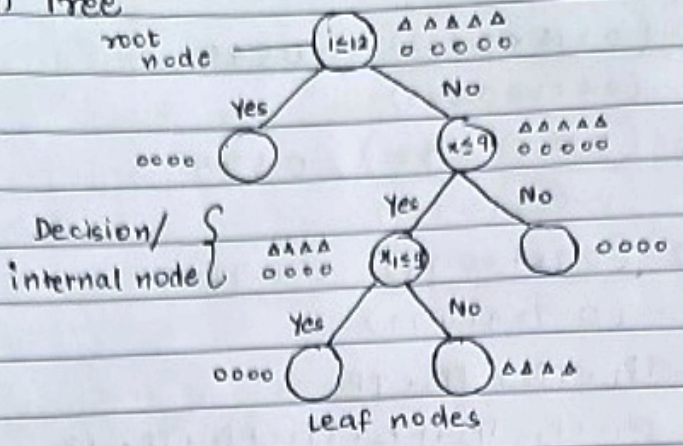
(9) Is the decision boundary linear or non-linear in case of Logistic Reg model?

(9) What is the impact of outliers on the Logistic Reg?

(8) Which algorithm is better in case of outliers present in dataset from logistic, linear reg and SVM?

4

- Decision Tree



~~Steps:~~

Algo

CART (Classification & Regression Tree) → Gini Index

ID3 ← requires high computational power with unstable tree structure when no of nodes increase. → Entropy & info gain

- Steps:

(1) Compute the entropy for dataset that is called as Entropy(S).

(2) For every attribute/feature:

(i) Calculate entropy for all other values: Entropy(A)

(ii) Take avg information entropy for current attribute.

(iii) Calculate gain for current attribute.

(3) Pick up the highest gain attribute.

(4) Repeat until we get the tree we desire

- Formula:

(1) Calculate entropy (amt of uncertainty in dataset)

$$\text{Entropy} = \frac{-p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

(2) Calculate avg info

$$I(\text{attribute}) = \sum \frac{pi+ni}{p+n}\,\text{Entropy}(A)$$

(3) Calculate info gain

Gain = Entropy(S) - I(attribute)

i.e, dif in entropy before and after splitting dataset into decision trees.

Eg:

Sr No	Outlook	Temp	Humidity	Wind	Play Tennis
1	Sunny	hot	High	Weak	N
2	Sunny	hot	High	Strong	N
3	Overcast	hot	High	Weak	Y
4	Rainy	Mild	High	Weak	Y
5	Rainy	Cool	Normal	Weak	Y
6	Rainy	Cool	Normal	strong	N
7	Overcast	Cool	Normal	strong	Y
8	Sunny	Mild	High	Weak	N
9	Sunny	Cool	Normal	Weak	Y
10	Rainy	Mild	Normal	Weak	Y
11	Sunny	Mild	Normal	Strong	Y
12	Overcast	Mild	High	Strong	Y
13	Overcast	Hot	Normal	Weak	Y
14	Rainy	Mild	High	Strong	N

Compute and draw decision tree. Decide ~~the~~ if player plays tennis, based on weather conditions.

→ $p = 9 \quad n = 5$ (+ve and -ve's)

Entropy(S) = $-\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.94$

Now split attributes : Eg: Outlook will be split into sunny, overcast, rainy.

Entropy (Sunny) = $-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.971$
$(p=2, n=3)$

Entropy (Rainy) = $-\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = 0.971$
$(p=3, n=2)$

Entropy (Overcast) = $-\frac{4}{4}\log_2\left(\frac{4}{4}\right) = 0$
$(p=4, n=0)$

I (Outlook) = $\frac{5}{14}(0.971) + \frac{5}{14}(0.971) + 0 = 0.693$

Gain : $0.94 - 0.693 = \boxed{0.247}$

For Temp,

Entropy (Hot) $= -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1$
$(p=2, n=2)$

Entropy (Mild) $= -\frac{4}{6}\log_2\left(\frac{4}{6}\right) - \frac{2}{6}\log_2\left(\frac{2}{6}\right) = 0.918$
$(p=4, n=2)$

Entropy (cool) $= -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) = 0.811$
$(p=3, n=1)$

$I(\text{Temp}) = \frac{4}{14}(1) + \frac{6}{14}(0.918) + \frac{4}{14}(0.811) = 0.911$

Gain: $0.94 - 0.911 = \boxed{0.029}$

For Humidity:

Entropy (High) = 0.985

Entropy (Normal) = 0.59

I (Humidity) = 0.788

Gain = $\boxed{0.152}$

For Wind:

Entropy (Strong) = 1

Entropy (Weak) = 0.811

I (Wind) = 0.892

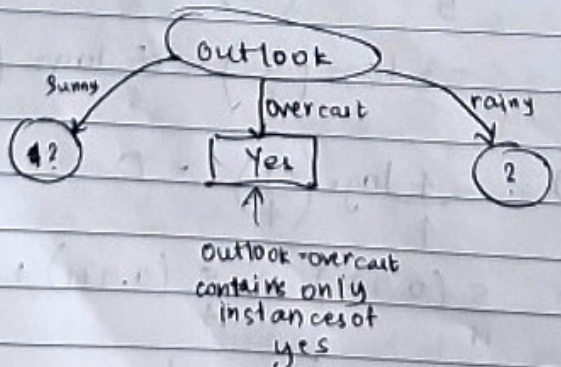
Gain = $\boxed{0.048}$

Gains in Ascending order:

Gain (Temp) < Gain (Wind) < Gain (Humidity) < Gain (Outlook)

0.029 0.048 0.152 0.247

Choose Outlook as root node.



Outlook-overcast contains only instances of yes

? ≡ Further Splitting necessary

- Now split datasets according to Sunny and Rainy

For Sunny:

Sr No	Temp	Humid	Wind	Play
1	hot	High	weak	N
2	hot	High	Strong	N
3	Mild	High	Weak	N
4	Cool	Normal	Weak	Y
5	Mild	Normal	Strong	Y

Entropy (S) = 0.971 (Only Sunny)

Entropy (hot) = 0

Entropy (cool) = 0

Entropy (mild) = 1

I(Temp) = 0.4 Gain = 0.571 (0.971 - 0.4)

Entropy (high) = 0

Entropy (normal) = 0

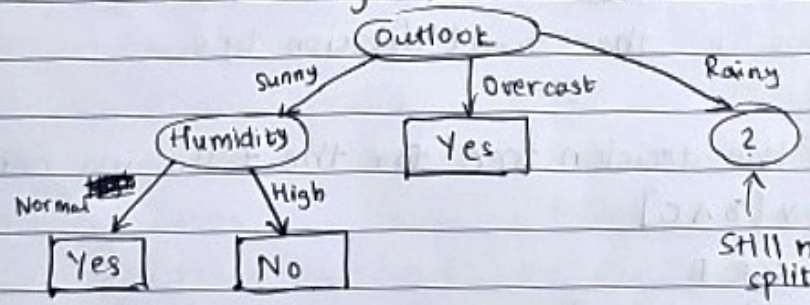
I(Humid) = 0 Gain = 0.971

Entropy (weak) =

Entropy (strong) =

I(Wind) = 0.951 Gain = 0.02

So we take humidity as next root



For rainy:

Sr No	Temp	Humid	Wind	Play
1	Mild	High	Weak	Y
2	Cool	Normal	Weak	Y
3	Cool	Normal	Strong	N
4	Mild	Normal	Weak	Y
5	Mild	High	Strong	N

Entropy(S) = 0.971

Entropy (Mild) = $-\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) = 0.918$
(p=2, n=1)

Entropy (Cool) = $2 \times \left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = 1$
(p=1, n=1)

I (Temp) = $\frac{2}{5}(1) + \frac{3}{5}(0.918) = 0.9508$

Gain = [0.02]

Entropy (High) = 1
(p=1, n=1)

I (Humid) = 0.9508

Gain = [0.02]

Entropy (Normal) = 0.918
(p=2, n=1)

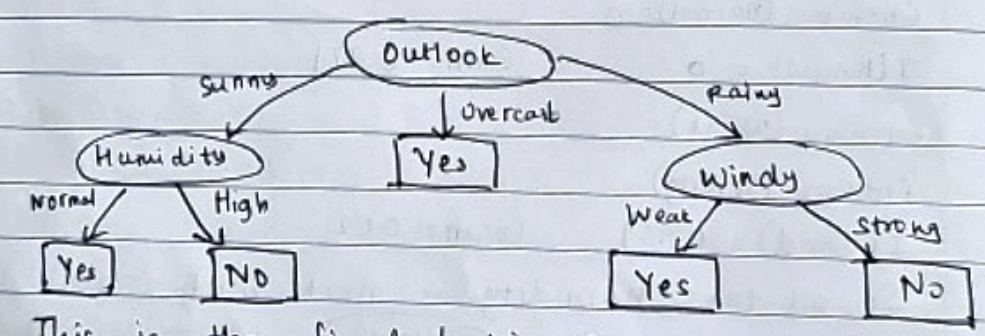
Entropy (Weak) = 0

I (Wind) = 0

Entropy (Strong) = 0

Gain = [0.971]

So take Windy as outlook



This is the final decision tree.

(Q) Draw the decision tree for the following condition.

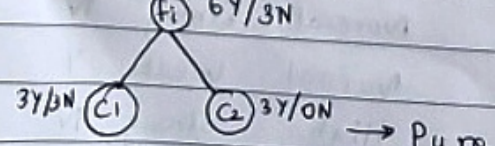
(1) $A \vee [B \wedge C]$

(2) A XOR B

(3) $[A \wedge B] \vee [C \wedge D]$

- Entropy based Decision tree:

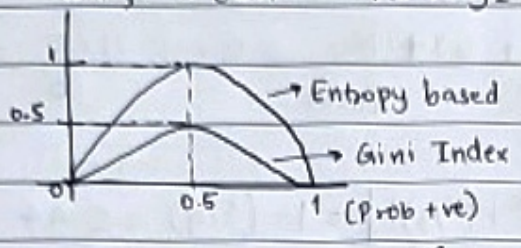
$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$ ($P_+ \Rightarrow +ve$, $P_- \Rightarrow -ve$)



→ Pure split

for C_2, $H(C_2) = -\frac{3}{3}\log_2(3/3) - 0 = 0$

C_1, $H(C_1) = -1/2 \log_2(1/2) - 1/2 \log_2(1/2) = 1$



- Gini Index : $GI = 1 - \sum_{i=1}^{n} (P)^2$ → +ve & -ve $= 1 - [(P_+)^2 + (P_-)^2]$

For C_1, $GI(C_1) = 1 - [(1/2)^2 + (1/2)^2] = 1 - 1/2 = 1/2$

Gini Index is computationally efficient as compared to the entropy based Decision tree.

(Q) Make a decision tree according to Gini Index.

	Weekend	Weather	Parent	Money	Decision
1	ω_1	Sunny	Yes	Rich	Cinema
2	ω_2	Sunny	No	Rich	Tennis
3	ω_3	Windy	Yes	Rich	Cinema
4	ω_4	Rainy	Yes	Poor	Cinema
5	ω_5	Rainy	No	Rich	Stay In
6	ω_6	Rainy	Yes	Poor	Cinema
7	ω_7	Windy	No	Poor	Cinema
8	ω_8	Windy	No	Rich	Shopping
9	ω_9	Windy	Yes	Rich	Cinema
10	ω_{10}	Sunny	No	Rich	Tennis

$GI = 1 - \sum_{i=1}^{n} (P)^2 = 1 - (0.6^2 + 0.2^2 + 0.1^2 + 0.1^2) = 0.58$

For Money, Poor = 03, Rich = 07

$GI(Poor) = 1 - (3/3)^2 = 0$

$GI(Rich) = 1 - [(3/7)^2 + (2/7)^2 + (1/7)^2 + (1/7)^2] = 0.694$

Weighted avg $= GI * \frac{Poor}{Total} + GI * \frac{Rich}{Total} = 0 \times \frac{3}{10} + 0.694 \times \frac{7}{10}$

$= 0.486$

For parent,

GI(Yes) = $1 - (5/5)^2 = 0$

GI(No) = $1 - [(1/5)^2 + (2/5)^2 + (1/5)^2 + (1/5)^2] = 1 - 0.28 = 0.72$

Weighted avg = $GI * \frac{Yes}{Total} + GI * \frac{No}{Total} = 0 + \frac{0.72 \times 5}{10} = 0.36$

For weather,

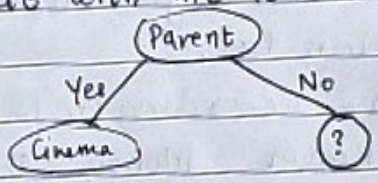
GI(Sunny) = $1 - [(1/3)^2 + (2/3)^2] = 1 - (5/9) = 0.44$

GI(windy) = $1 - (1^2 + 3^2/4^2) = 1 - 10/16 = 0.375$

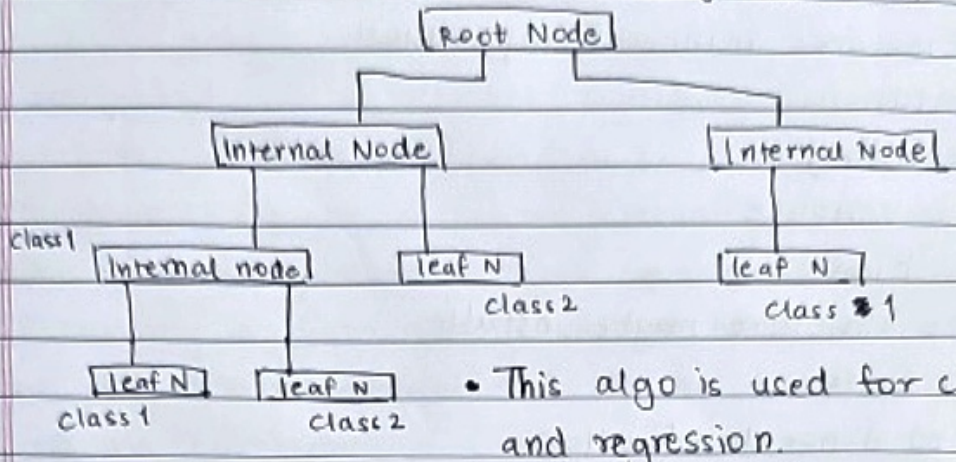
GI(rainy) = 0.44

Weighted avg = $0.44 * x\ 2x\ 3/10 + 0.375 \times 4/10 = 0.416$

The attribute with the lowest Gini index will be considered as root



- CAR Tree (Classification And Regression Tree)



- This algo is used for classification and regression.
- The best split point of each input is taken.
- Based on the best split point of each input in step 1, the new "Best" split point is identified.
- Split the chosen input according to the best split point.
- Continue the splitting until a stopping rule is satisfied or no further desirable splitting is available.

- CART for classification:

- It works by recursively splitting the training data into smaller and smaller subsets based on certain criteria. The goal is to split data in such a way that minimizes the impurity within each subset. For classification, CART uses Gini impurity.

- CART for Regression:

- Regression is an algo where target variable is continuous and tree is used to predict its value. CART for regression is a decision tree that creates a tree like structure to predict continuous target variable. It works by splitting training data recursively into smaller datasets based on specific criteria. The objective is to split the data in a way that minimizes residual reduction in each subset.

- Advantages:

(1) Results ~~and~~ are simplistic
(2) CART can implicitly perform feature selection.

(3) Outliers have no meaningful effect on CART.
(4) It requires minimal supervision

- Limitations

(1) Overfitting
(2) High variance
(3) low bias
(4) Tree structure maybe unstable.

- Use Cases

(1) Blood donor classification
(2) Environmental and Ecological data
(3) Financial applications.

(Q)

CGPA	Interactive	Pract Knowledge	Comm Skills	Job Offer
≥9	Yes	V Good	Good	Y
≥8	N	good	moderate	Y
≥9	N	avg	poor	N
<8	N	avg	good	N
≥8	Y	good	mod	Y
≥9	Y	good	mod	Y
<8	Y	good	poor	N
≥9	N	V Good	good	Y
≥8	Y	Good	good	Y
≥8	Y	avg	good	Y

Y=7, N=3 (According to Job offer)

$GI = 1 - (7/10)^2 - (3/10)^2 \Rightarrow \boxed{GI(T) = 0.42}$ Imp

CGPA	Y	N
≥9	3	1
≥8	4	0
<8	0	2

$GI(T, CGPA \in (\geq 9, \geq 8)) = 1 - (7/8)^2 - (1/8)^2 = 0.2187$

$GI(T, CGPA \in (<8)) = 1 - (2/2)^2 = 0$

$GI(T, CGPA) = \frac{|S_1|}{|T|} GI(S_1) + \frac{|S_2|}{|T|} GI(S_2)$

$= 8/10 \times 0.2187 + 0 \times 2/10$

$\boxed{GI(T, CGPA) = 0.17496}$

Now, $GI(T, CGPA \in \{\geq 9, < 8\}) = 1 - (3/6)^2 - (3/6)^2 = 0.5$

$GI(T, CGPA \in \{\geq 8\}) = 1 - (4/4)^2 = 0$

$GI(T, A\text{-}CGPA) = \frac{6}{10} \times 0.5 + 0 \times \frac{4}{10} = \boxed{0.3}$

Now, $GI(T, CGPA \in \{\geq 8, < 8\}) = 0.44$

$\{\geq 9\} = 0.375$

$\therefore GI(T, A\text{-}CGPA) = \boxed{0.414}$

So, first GI is minimum. Subtract from GI(T).

$\therefore \Delta Gini = GI(T) - GI(T, CGPA) = 0.42 - 0.1755 = \boxed{0.244}$ — (1)

Now wrt interactive

Interactive	Y	N
Yes	5	1
No	2	2

$GI(T, Interactive \in \{Yes\}) = 1 - (5/6)^2 - (1/6)^2 = 0.27$

$GI(T, \{No\}) = 1 - (2/4)^2 - (2/4)^2 = 0.5$

$GI(T, A\text{-}interactive) = \frac{6}{10} \times 0.27 + \frac{4}{10} \times 0.5 = \boxed{0.362}$

$\Delta Gini(Inter) = 0.42 - 0.362 = \boxed{0.058}$ — (2)

Now wrt Pract knowledge

Prac	Y	N
v good	2	0
good	4	1
avg	1	2

$GI(T, \{Vgood, good\}) = 1 - (6/7)^2 - (1/7)^2 = 0.245$

$GI(T, \{avg\}) = 1 - (1/3)^2 - (2/3)^2 = 0.44$

$GI(T, A\text{-}Prac) = \frac{7}{10} \times 0.245 + \frac{3}{10} \times 0.44 = \boxed{0.3035}$

$GI(T, \{G, avg\}) = 0.4688$

$GI(T, \{Vgood\}) = 0$

$\Big\} GI(T, A\text{-}Prac) = \frac{8}{10} \times 0.4688 = \boxed{0.375}$

$GI(T, \{Vg, avg\}) = 1 - (3/5)^2 - (2/5)^2 = 0.48$

$GI(T, \{G\}) = 0.32$

$\Big\} GI = \boxed{0.4}$

$\Delta Gini(Prac) = 0.42 - 0.3035 = \boxed{0.116}$ — (3)

Now wrt Comm skills,

Comm	Y	N
G	4	1
M	3	0
P	0	2

$GI(T, \{M, P\}) = 1 - (3/5)^2 - (2/5)^2 = 0.48$

$GI(T, \{G\}) = 1 - (4/5)^2 - (1/5)^2 = 0.32$

$GI(T) = \boxed{0.4}$

GI (T, {G,M}) = $1-(7/8)^2-(1/8)^2 = 0.218$ } GI(T) = 0.1744

GI (T, {P}) = 0

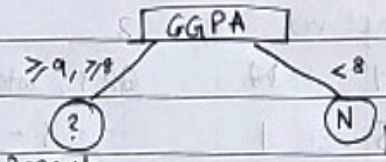
GI (T, {G,P}) = $1-(4/7)^2-(3/7)^2 = 0.489$ } GI(T) = 0.342

GI (T, {M}) = 0

Lowest is 0.1744

Δ Gini (Comm) = 0.42 − 0.1744 = 0.2456 — (4)

Now taking ~~too~~ highest ΔGini as root. Note: Comm and CGPA are almost same so take any.

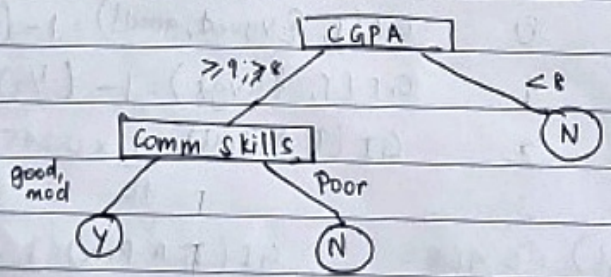


Repeat process after removing <8.

Attribute	GI	Gini
Interactive	0.056	0.1624
Pract.	0.125	0.0934
Comms	0	0.2184

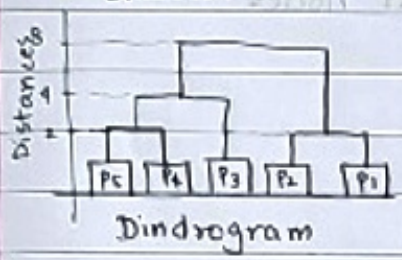
} direct answers calculated by mam

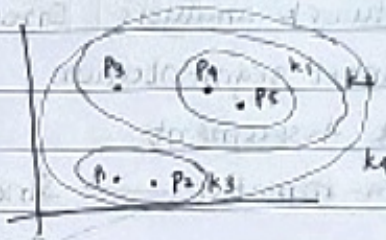
So take comm as root,



- Hierarchical Clustering:

→ Nearest neighbour → Assign cluster
→ Many cluster → combine all





Above is an example of Bottom to Top (One data pt to entire cluster)

Hierarchical
- Aggromative → Bottom Top approach
- Devisive (not in syllabus) → Top Bottom Approach (One cluster broken into data pts)

- Merits:

(1) It is simple to implement & easy to interpret
(2) It is robust, means no predefined cluster k, so given hierarchical structure is reliable & accurate
(3) Algo is flexible. It is suitable to any type of data.
(4) This algo is versatile, it can be used for both supervised & unsupervised learning.

- Demerits:

(1) It is very sensitive to outliers, means if such outliers are present, hierarchical structure may not be accurate.
(2) Computationally expensive, no of datapoints are more then it is time consuming
(3) Not suitable for multidimensional data.
(4) Cannot handle categorical variables as it cannot convert categorical data to numerical.

Teacher's Signature:.....

- Usecases:

(1) Image Segmentation
(2) Text Analysis : Text mining & classification
(3) Recommendation system
(4) Network analysis: Grouping similar nodes
(5) Market segmentation
(6) Rick Assessment

- Aggromative ← Single link
  Complete link

(8) Show the hierarchical clustering with single link on given dataset.

→

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0					
P_2	3	0				
P_3	8	9	0			
P_4	9	8	1	0		
P_5	5	10	6	7	0	
P_6	4	9	7	8	(2)	0

If these pts are given, just ignore. Focus on lower triangle.

Bcoz of bottom up approach, see last row, i.e P_6. It has minimum distance from P_5, so make a cluster $P_{5,6}$.

	P_1	P_2	P_3	P_4	$P_{5,6}$
P_1	0				
P_2	3	0			
P_3	8	9	0		
P_4	9	8	(1)	0	
$P_{5,6}$	4	9	6	7	0

Taking min distance of points from cluster.
Eg: $P_1 P_5 = 5$, $P_1 P_6 = 4$
So $P_1 (P_{5,6}) = 4$.

Now see row P_4. Minimum is $P_4 P_3$ so merge.

Teacher's Signature:.....

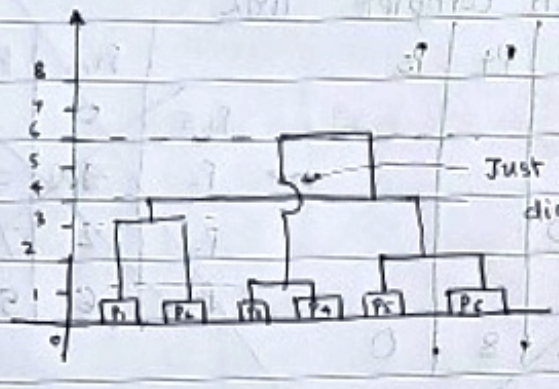
	P_1	P_2	$P_{3,4}$	$P_{5,6}$
P_1	0			
P_2	(3)	0		
$P_{3,4}$	8	8	0	
$P_{5,6}$	4	9	6	0

→

	P_1P_2	P_3P_4	P_5P_6
P_1P_2	0		
P_3P_4	8	0	
P_5P_6	(4)	6	0

So now merge $P_1P_2P_5P_6$

	$P_{1,2,5,6}$	$P_{3,4}$
$P_{1,2,5,6}$	0	
$P_{3,4}$	6	0



Just for understanding (for clear diagram, draw P_3P_4 on other side)

Q. Apply the aggromative algo and represent the ~~dend~~ dindrogram on given dataset with single link.

	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	3	7	0		
P_4	6	5	9	0	
P_5	11	10	(2)	8	0

→

	P_1	P_2	$P_{3,5}$	P_4
P_1	0			
P_2	9	0		
$P_{3,5}$	3	7	0	
P_4	6	(5)	8	0

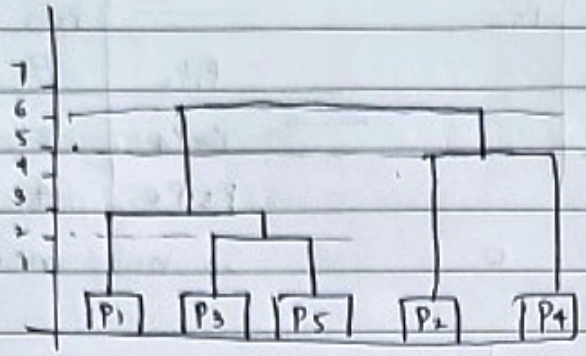
→

	P_1	$P_{2,4}$	$P_{3,5}$
P_1	0		
$P_{2,4}$	6	0	
$P_{3,5}$	(3)	7	0

→

	$P_{1,3,5}$	$P_{2,4}$
$P_{1,3,5}$	0	
$P_{2,4}$	(6)	0

Teacher's Signature:.....



- In complete link, consider maximums instead of minimums in same process as above.

(Q) Same problem with complete link

	P1	P2	P3	P4	P5
P1	0				
P2	9	0			
P3	3	7	0		
P4	6	5	9	0	
P5	(11)	10	2	8	0

→

	P1,5	P2	P3	P4
P1,5	0			
P2	~~9~~ 10	0		
P3	2	7	0	
P4	6	5	(9)	0

→

	P1,5	P2	P3,4
P1,5	0		
P2	9	0	
P3,4	2	(5)	0

→

	P1,5	P2,3,4
P1,5	0	
P2,3,4		0

→

	P1,5	P2	P3	P4
P1,5	0			
P2	10	0		
P3	3	7	0	
P4	8	5	(9)	0

→

	P1,5	P2	P3,4
P1,5	0		
P2	10	0	
P3,4	(8)	7	0

→

	P1,3,4,5	P2
P1,3,4,5	0	
P2	(10)	0

Teacher's Signature:.....

- For complete link, consider same clusters, i.e, you have to merge pts with min distance between them. But, while merging (making changes in table) consider max distances.

(Q) Same with complete.

	P_1	P_2	$P_{3,5}$	P_4
P_1	0			
P_2	9	0		
$P_{3,5}$	11	10	0	
P_4	6	5	9	0

→

	P_1	$P_{2,4}$	$P_{3,5}$
P_1	0		
$P_{2,4}$	9	0	
$P_{3,5}$	11	10	0

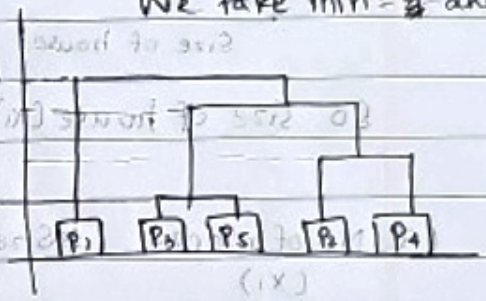
We take min = 5 and merge.

→

	$P_{1,3,5}$	$P_{2,4}$
$P_{1,3,5}$	0	
$P_{2,4}$	10	0

→

	P_1	$P_{2,4,5,3}$
P_1	0	
$P_{2,4,3,5}$	11	0



- Reducing Curse of Dimensionality can be done in 2 ways:

(1) Feature selection: Take only imp features highly correlated with target

(2) Feature extraction: Transform the given feature set to a new, improved feature set.

- Advantages of Dimension Reduction:

- Reduces COD
- Improves performance of model
- Visualize the data, i.e, understanding the data.

- $cov(X,Y) = \sum_{i=1}^{N} \frac{(x-\bar{x})(y-\bar{y})}{N-1}$ is used to quantify relationship.

~~when cov~~

$cov(x,y) = 1$ linear ↗ ~ 0 ⇒ No relationship

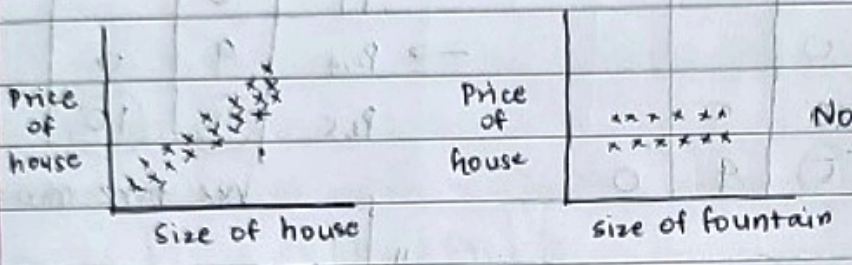
$= -1$ linear ↘

Teacher's Signature:.....

- Correlation → Pearson correlation

$= \frac{cov(x,y)}{\sigma_x * \sigma_y}$ = -1 to 1

(Q) Size of house (x_1) Size of fountain (x_2) Price of house (y)



So size of house (x_1) is the selected feature, without loss of info.

(Q) No of rooms (x_1) Size of room (x_2) Price of house (y)

Here both features are imp, so apply feature extraction, i.e, dimensionality reduction. Reduce to 1 feature.

2f → old f set

⇓

Transform

⇓

D1 → New f set.

⇓

Size of house

⇓

Predict y without loss

- Principal Component Analysis:

(1) Calculate mean for each

$\bar{X}_i = \frac{1}{N}(X_{i1} + X_{i2} + \ldots + X_{iN})$

(2) Calculate Cov

$$\text{Cov}(X_i, X_j) = \frac{1}{N-1} \sum_{k=1}^{N} (X_{ik} - \bar{x}_i)(X_{jk} - \bar{x}_j)$$

(3) Make cov. matrix

(4) Calculate eigenvalues and eigen vectors of cov matrix.

$\det(S - \lambda I) = 0$ Polynomial eqn of degree n has n real roots which are eigenvalues of S.

(Q)

F	Ex 1	Ex 2	Ex 3	Ex 4
x_1	4	8	13	7
x_2	11	4	5	14

Derive new feature set from given data using PCA.

(1) Calculate mean

$\bar{x}_1 = 8$

$\bar{x}_2 = 8.5$

(2) Calculate cov

$$cov(x_1, x_2) = \frac{1}{N-1} \sum_{k=1}^{N} (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)$$

$$= \frac{1}{3}\left[(4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)\right]$$

$$= -11$$

$$cov(x_1, x_1) = \frac{1}{3}\left[(4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\right]$$

$$= 14$$

$$cov(x_2, x_2) = \frac{1}{3}\left[(11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2\right]$$

$$= 23$$

(3) $S = \begin{bmatrix} cov(x_1,x_1) & cov(x_1,x_2) \\ cov(x_2,x_1) & cov(x_2,x_2) \end{bmatrix} = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$

(4) Eigenvalues

$$(S - \lambda I) = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$= \begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix}$$

$$\det(S - \lambda I) = (14-\lambda)(23-\lambda) - (11 \times 11)$$

$$= \lambda^2 - 37\lambda + 201 = 0$$

$\lambda = 30.3848, 6.615$

$\lambda_1 \quad \lambda_2$

Teacher's Signature:

(5) Compute eigenvectors

$$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 0 \end{bmatrix} = (S - \lambda I)(U)$$

$$\therefore \begin{bmatrix} (14-\lambda)u_1 - 11u_2 \\ -11u_1 + (23-\lambda)u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$(14-\lambda)u_1 - 11u_2 = 0$$

$$-11u_1 + (23-\lambda)u_2 = 0$$

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda} = t$$

$$\therefore u_1 = 11t \;, \quad u_2 = (14-\lambda)t$$

$$u_1 = \begin{bmatrix} 11 \\ 14-\lambda \end{bmatrix}$$

$$\|u_1\| = \sqrt{11^2 + (14-\lambda)^2} = 19.7346$$

$$e_1 = \begin{bmatrix} 11/\|u_1\| \\ (14-\lambda_1)/\|u_1\| \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.83 \\ \end{bmatrix}$$

(5) Compute first PC1

$$e_1^T * \begin{bmatrix} X_{1k} - \bar{X}_1 \\ X_{2k} - \bar{X}_2 \end{bmatrix} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} X_{11} - \bar{X}_1 \\ X_{21} - \bar{X}_2 \end{bmatrix}$$

$$= 0.5574\,(4-8.5) - 0.8303\,(11-8.5)$$

$$= -4.30535$$

F	Ex1	Ex2	Ex3	Ex4
PC1	-4.30535	3.73	5.69	-5.1238

Teacher's Signature:.....

(Q) Apply the concept of PCA and calculate eigenvalues.

X	Y
2.5	2.4
0.5	2.2
2.2	1.9
1.9	2.2
3.1	3.0

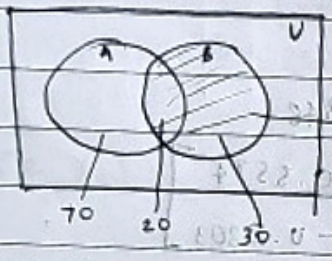
$\bar{X} = 2.04$

$\bar{Y} = 2.34$

$cov(X,X) = \frac{1}{4}[3.752] = 0.938$

$cov(Y,Y) = \frac{1}{4}[1.158] = 0.2895$

- Naive Baye's Theorem



shrinking

$P(B) = 3/10$

$P(A \cap B) = 2/10$

$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{0.2}{0.3} = 0.67$

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

$P(A \cap B) = P(B)\,P(A|B) = P(A)\,P(B|A)$

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

$x_1, x_2, \ldots, x_n$ } i/p $\{y\}$ = o/p ⇒ Target

$f_1, f_2, \ldots, f_n$ }

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

A B

Teacher's Signature:.....

$$= \frac{P(y)\ \pi_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)\ldots P(x_n)}$$

$$\therefore P(y|x_1,\ldots x_n) \propto P(y)\ \pi_{i=1}^{n} P(x_i|y)$$

$$y = \underset{\text{highest probability}}{\arg\max}\ P(y)\ \pi_{i=1}^{n} P(x_i|y)$$

(Q)

No	Outlook	Temp	Humidity	Play Tennis
1	Sunny	~~High~~ Hot	High	N
2	Sunny	~~High~~ Hot	High	N
3	Overcast	~~High~~ Hot	High	Y
4	Rain	Mild	High	Y
5	Rain	Cool	Normal	Y
6	Rain	Cool	Normal	N
7	Overcast	Cool	Normal	Y
8	Sunny	Mild	High	N
9	Sunny	Cool	Normal	Y
10	Rain	Mild	Normal	Y
11	Sunny	Mild	Normal	Y
12	Overcast	Mild	High	Y
13	Overcast	~~Not High~~ Hot	Normal	Y
14	Rain	Mild	High	N

From the given dataset, what is probability of person to play tennis if climate is sunny and hot?

→ ~~P(sunny) = 2/5~~ Outlook

	Y	N	P(Y)	P(N)
Sunny	2	3	2/9	3/5
~~Rain~~ Overcast	4	0	4/9	0/5
~~Overcast~~ Rain	3	2	3/9	2/5
Total	9	5	100%	100%

Teacher's Signature:.....

Temp:

	Y	N	P(Y)	P(N)
Hot	2	2	2/9	2/5
mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Play Tennis?		P(y) & P(N)
Yes	9	9/14
No	5	5/14
Total	14	100%

$$P(Y \mid \text{sunny, hot}) = \frac{P(\text{sunny} \mid \text{yes})\, P(\text{hot} \mid \text{yes})\, P(y)}{P(\text{sunny})\, P(\text{hot})}$$

$$= \frac{2}{9} \times \frac{2}{9} \times \frac{9}{14} = \frac{14}{45}$$

$$\frac{5}{14} \times \frac{4}{14}$$

Denominator is always constant so neglect.

$P(y \mid \text{sunny, hot}) = 0.0317$

$P(N \mid \text{sunny, hot}) = 0.0857$

Normalize,

$$\frac{P(y)}{P(y) + P(N)} = \frac{0.0317}{0.0317 + 0.0857} = 0.2656 \sim \boxed{0.27} \quad P(y)$$

$$P(N) = 1 - 0.27 = \boxed{0.73}$$

- ~~Naive Bay~~

Teacher's Signature:.....

- Naive Baye's Classifier: How to handle zero probability.

(a)	TEXT	Category
1	A great game	Sports
2	The election was over	Non sports
3	Very clean match	Sports
4	AClean but forgettable game	~~Non~~ sports
5	It was a close election	Non Sports

Classify the given sentences ("A very close game") using Naive Bayes Classifier.

Total no of words in sports = 11

Total no of words in non-sports = 9

Total no of unique words = 14

So we need to consider probability of each word in given sentence.

Sports:

P(A) = 2/11 P(very) = 1/11 P(close) = 0/11 P(game) = 2/11

So ~~#~~ P(close) = 0, this zero probability is handled by Laplace's smoothing:

$$\theta_i = \frac{x_i + \alpha}{N + \alpha d}$$

θ_i = P(word)

x_i = Word count

~~α is 0~~ N = Total no of words

d = No of unique words

α is > 0 but always $\alpha = 1$ constant.

	P(W/sports)	P(W/Non sports)
P(A)	$\frac{2+1}{11+14} = \frac{3}{25}$	$\frac{2}{23}$
P(very)	$\frac{1+1}{11+14} = \frac{2}{25}$	$\frac{1}{23}$
P(close)	$\frac{0+1}{11+14} = \frac{1}{25}$	$\frac{2}{23}$
P(game)	$\frac{2+1}{11+14} = \frac{3}{25}$	$\frac{1}{23}$

P(W/Non sports) $= \frac{2 \times 1 \times 2 \times 1}{(23)^4} = \frac{4}{(23)^4} = 0.46 \times 10^{-5}$

$P(W/sports) = \frac{3 \times 2 \times 1 \times 3}{25 \times 25 \times 25 \times 25} = 0.46 \times 10^{-5}$

Teacher's Signature:.....