git clone link → git init
git add .
git commit -m "first commit"
git branch -M main
git push -u origin main

→ make file in github
→ clone to folder
#git clone HTTPS link
or open folder and open git bash right click and paste
→ git init initialize every git repo in side git reposcity
→ git status
→ git add . This command add whatever we create into git repo
→ git commit -m "First commit adding .html"
→ Push this change to git hub so every person can see the change
→ git branch -M main
git push -u origin main

working dir → staging area (git add) → git commit → repository

Git & Github

⇒ untracked file: git does not know about ~~that~~ that files
→ we sent our project (which in our system) to git repository
→ we can't send direct to git
so first project to staging index
and then staging index to git
~~Git~~.
let we have to files in project index.html and contact.html
$ git add contact.html (tracked file) → sent contact.html to stage
git add . all file to stage
So at this point of time we have 1 file in stage and
to file at project and 0 files at git repo, index.html (untracked file)
$ git add index.html
2 file stage, 2 file project, 0 in git repo

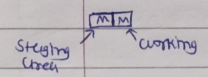
→ to commit file
$ git commit -m "First commit 7June23"
all the files push to git repo (not github)
$ ~~g~~ git commit -a -m "First commit" add & modified in one line code
→ If we doesn't want to track some file ~~that~~ then make new file .gitignore and write name of those files.

→ $ git log give history of all commit Author & Date
and also give SHA (incription algo) in every project its unique
$ git log --oneline give histry in one line
$ git show {SHAkey} show that version of file

$ git log -P -3 -5 -8
q for quite

⇒ $ git config --global user.name
user.email

$ ls $ ls -a give list in folder
$ mkdir make directory
$ cat {filename} can check what chang we made
$ git branch -m master
$ git diff compare working tree to staging area
$ git diff --stage compare staging area to last commit
$ touch west.html to make west.html file
$ git rm --cached west.html remove file from staging area (untracked)
$ git rm west.html delete file from staging area & working directory permanet delet
$ git status -s



$

$ git checkout {file name} match file to Previous commit
$ git checkout ~~file name~~ -f " all files " " "

⇒ git ignore
→ If we write file name inside .gitignore then git ignore that files any where in root node
→ *.html = ignore all html files
→ /filename = ignore from root node only (not root → cb → filename)
→ Folder name/ = ignor folder name
indicate that this is folder

http link = origin
$ git remote

⇒ branch
$ git branch gives all the branch
$ git branch slave make new branch name 'slave'
$ git checkout slave switched branch to 'slave'
$ git merge slave ~~merge~~ slave to master (:: run from master)
$ git branch --d slave delet slave
$ git branch -M master change branch name to master
$ git checkout -b New make 'new' branch ~~new~~ and switched to that branch

⇒ to Push your programs (Project) to github
new make file in github
$ git Clone {https link of github file}
add file into your computer
If you want to update your Changes in github then you need to Push that in git hub
$ git remote , $ git remote -V
O/P : origin ← origin is https link which we got from github
$ git Push origin master Push Project to github
name of branch that you want to Push

existing
can be any name
$ git remote add origin {httpslink}
$ git branch -M master first commit file then push
$ git push -u origin master

PAGE NO.:

★ ML Revision 27 July 23

⇒ $F_{beta} = (1+\beta)^2 \frac{PR}{(\beta^2 P)+R}$ FP Impo ↓β , FN βT Type1 , Type1

⇒ Maximum likelihood

TL Prob of correctly classified

Cross entropy take -log of max likelihood

⇒ Huber loss = $\begin{cases} \frac{1}{2}(y-\hat{y})^2 & , |y-\hat{y}| \leq \delta \text{ hyper parameter} \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2 & , \text{otherwise} \end{cases}$

⇒ Multi class Cross entropy

Categorical Cross entropy

$$\text{Loss} = -\sum_{j=1}^{c} y_{ij} \log(\hat{y}_{ij})$$

For i^{th} row ; y_{ij} ↑ actual ; $\hat{y}_{ij}$ ↑ pred

Sparse categorical cross entropy

instead of one hot encoding we do numbering to each op

Ex $y_1 \to 1$, $y_2 \to 2$, $y_3 \to 3$ and calculate only for that y

$$\text{Total loss} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} y_{ij} \log(\hat{y}_{ij})$$

After updating weights for every model For test data we get op

take soft max of that and max prob gives op

⇒ Entropy & gini impurity (For Purity check(s))

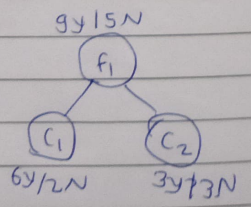
$H(S) = -P_+ \log P_+ - P_- \log P_-$

$GI = 1 - \sum_{i=1}^{p} (P)^2$

⇒ Ig (Feature selection)

$$\text{Gain} = H(S) - \sum \frac{|S_v|}{|S|} H(S_v)$$

Parent ; sample in child node ; Total sample



PAGE NO.:

PAGE NO.:

* DL Revision 28 July 23

⇒ ML use statistic technique for find relation bet^n i/p & o/p while DL follows logical structure called Neural Networks

⇒ $Z_1 = W_1 A_0 + b_1$

$A_1 = \sigma(Z_1)$

$Z_2 = W_2 A_1 + b_2$

$A_2 = \sigma(Z_2)$

⇒ Regularization & drop out layers apply for last layer If it work then try last & second last so on...

CNN best result = $p \approx 0.4 - 0.5$

RNN = $p \approx 0.2 - 0.3$

ANN $p \approx 0.1 - 0.5$

⇒ This method is like RF we use NN instead of Trees

⇒ For every epoch remove p fraction of neuron in each HL randomly

⇒ For o.p activate all features and all neurons and multiply $(1-p)$ with every weight

dead neuron or dead activation prob

⇒ Relu $= \begin{cases} z, & z>0 \\ 0, & z<0 \end{cases}$, Leaky when $\alpha = 0.01$ in P Relu $= \begin{cases} \alpha, & x \geq 0 \\ \alpha x, & x<0 \end{cases}$

ELU $= \begin{cases} x, & \text{if } x \geq 0 \\ \alpha(e^x - 1) & x<0 \end{cases}$ selu $= \lambda$ ELU $\alpha \approx 1.6732...$

mean of o/p close to 0 $\mu = 0$ self normalizing $\lambda = 1.0507...$

o/p is $\sigma = 1$, $\mu = 0$

σ = vanishing grad prob tern = vanishing

$\mu \neq 0$ $\mu = 0$

swish = used when NN > 40 layers

$x\sigma$

soft plus = $\ln(1 + e^x)$

soft max $\frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_i}}$

PAGE NO.:

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

⇒ Translation covariance

After applying convolution layer feature which is find by CNN is dependent on location this is called translation variance

we can avoid this by using pooling. Pooling is a way to down sample feature map so features become independent of location and it called translation invariance

⇒ image → convolution + Relu + pooling + conv + Relu + pooling → flatten → Fully connected layers → o/p

For CNN Trainable parameters are only depends on filters size and ~~bias~~ number of filters

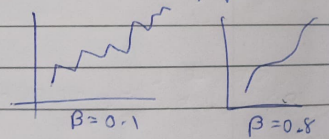
(3, 3, 3)

50 filter has (3×3×3×50) + 50 learnable parameters (every filter has bias value)

⇒ $V_t = \beta V_{t-1} + (1-\beta) S_t$ (S_t: a_t) $0 < \beta < 1$

V_t approximating avg over $\frac{1}{1-\beta}$

$V_n = (1-\beta)(\beta^{n-1} a_1 + \beta^{n-2} a_2 + .. a_n)$



⇒ For Deep Learning L2 (Ridge Regularization) gives best result

$w_{nr} = w_{old} - \eta \frac{\partial L}{\partial w_{old}}$

$L = L' + \frac{\lambda}{2} \sum_{i=1}^{n} \|w_i\|^2$ = (without reg)

$\frac{\partial L}{\partial w_{old}} = \frac{\partial L'}{\partial w_{old}} + \frac{\lambda}{2}(2 w_{old})$

$w_{new} = (1-\lambda\eta) w_{old} - \eta \frac{\partial L'}{\partial w_{old}}$

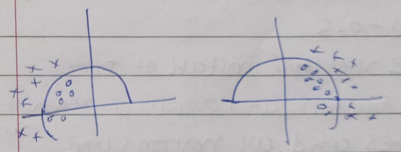
$1 - \lambda\eta < 1$

$0 < \lambda < \infty$

$0 < \eta < \infty$ (LR)

⇒ covariate shift

Input distribution and o/p distribution is diff but o/p & i/p relation are same (in figure given by same curve ∩)



⇒ Batch norm

⇒ change in activations due to the change in network parameter during training is called internal covariate shift.

⇒ In Batch normalization we convert OP of HL into gaussian distribution ($\sigma = 1, \mu = 0$) so internal covariate shift reduce and training will be fast and stable

⇒ If we dont use Batch norm LR should low and carefully ini param otherwise training will be unstable

* K means Clustering

⇒ From sklearn.cluster import KMeans

Km = KMeans (n_clusters = 5)

Km.fit_predict (df)

⇒ Km.inertia = gives WCSS = within cluster sum of square

⇒ Km.fit_predict(x) : fit and predict cluster for each rows in dataset X

* StackingClassifier

sklearn.ensemble.StackingClassifier

⇒ clf = StackingClassifier (
estimators = estimators, ← list of tuples
final_estimator = LogisticRegression(),
CV = 10

* Agglomerative Clustering

⇒ From sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n_cluster=5,
affinity='euclidean',
linkage='ward')

labels_ = cluster.fit_predict(data)

★ Desision Trees

Plot_tree

⇒ from sklearn.tree import DesitionTreeclassifier
clf = DecisionTreeclassifier()
clf.fit(x_train, y_train)

⇒ Plot_tree(clf)

★ Voting ensemble

From sklearn.esemble import VotingClassifier

⇒ estimators = [('lr', clf1), ('rf', clf2), ('knn', clf3)]
VC = VotingClassifier(estimators = estimators, voting='hard')
↑ list of tuples
can also assign weights
weights = [1, 5, 3]

★ Bagging

⇒ from sklearn.ensemble import BaggingClassifier
bag = BaggingClassifier(
base_estimator = DecisionTreeclassifier(),
taking 50% of data to every DT → n_estimators = 500,
sample with repla → max_sample = 0.5, max_features = 0.5
bootstrap = True, bootstrap_features = True
random_state = 42,
divide task in CPU → verbose = 1,
n_jobs = -1

⇒ bag.estimators_samples_
list of array each array contain index number of which assign to that DT

⇒ bag.estimators_features_

★ Random Forest

⇒ rf = RandomForestClassifier(max_features = 2,
How many DT → n_estimator=100,
max_features={5, auto, sqrt, log2, 0.3 (30%), None}
bootstrap=True, ← Repla
100 Rows → max_sample = 100,
DT hyper parameters:
max_depth = 5,
criterion = {gini, entropy},
min_sample_split = {5 (min), 0.4 (%)},
max_leaf_nodes =
min_impurity_decrease

★ Aduboost

no of model
clfa = AduboostClassifier(n_estimators = 1500, learning_rate = 0.1)
base_estimator, default = DT with height=1
algorithm = 'SAMME.R' (default), 'SAMME'

★ Date time

pd.to_datetime(data['date'])

data['date'].dt.year / dt.day / dt.dayofweek

-dt.month / month_name() / dt.day_name()

-dt.week / dt.quarter / dt.

★ Missing indicator

⇒ mi = MissingIndicator() → (n,1) ⇒ new col with col1 is missing or not

mi.fit(x_train['col1'])

mi.transform(x_train['col1']) → op = array of (n,1) with T/F

[[False],
[True],
]

⇒ simple imputer also contain missing indicator so no need to use seperately

SimpleImputer(add_indicator=True)

⇒ Knn = KNNImputer(n_neighbors = 3, weights = 'distance' / 'uniform')

SI:-

⇒ Simple imputer(strategy = 'mean' / 'median' / 'most_frequent' / 'constant' / add_indicator) — for arbitrary value imputation

⇒ si = simple imputer(strategy='constant', fill_value = 1000)

(MICE) iterative imputer ⇒ ite = IterativeImputer() tol = 10^{-3} max_iter initial_strategy= add indicator

xtrain = ite.fit(xtrain)

★ PCA

from sklearn.decomposition import PCA

pca = PCA(n_components = 100)

pca.fit_transform(x_train)

pca.transform(x_test)

pca.explained_variance_ = Eigen values

pca.components_ = Eigen vectors

pca.explained_variance_ratio_

★ Polynomial logistic regression (non linear logistic regression)

⇒ from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree=3, include_bias=False)

x_trf = poly.fit_transform(x)

after this x → $x_1, x_1^2, x_1^3, x_1x_2, x_2^2, x_2^3$ ... (Features)

and now we can apply LogisticRegression on this modified x_trf

Pandas Profiling

⇒ !pip install pandas-profiling
from pandas_profiling import ProfileReport
Prof = ProfileReport(df)
Prof.to_file(output_file='output.html')

★ one hot encoding using pandas

⇒ pd.get_dummies(df, columns=['fuel','owner'], drop_first=True)

★ one hot encoding using sklearn

⇒ from sklearn.~~model selection~~ preprocessing import onehotencoder
ohe = OneHotEncoder(drop='first', sparse=False, dtype=np.int32)
ohe.
sparse=F gives dense array, T gives sparse matrix
ohe.fit(X_train[['col1','col2']])
trf = FunctionTransformer(func=np.log1p) / pt=PowerTransformer(method='box-cox')

⇒ SimpleImputer()
OrdinalEncoder(categories=[list]) (too high / too low)
OneHotEncoder(drop='first', ...)
si.transform(series) gives np array
ohe.ohe.

np.concatenate((arr1, arr2, arr3), axis=1)

★ Column transfer (make_column_transformer)

⇒ From sklearn.compose import ColumnTransformer
transformer = ColumnTransformer(transformers=[
('name1', SimpleImputer(), ['Fever']),
('name2', OrdinalEncoder(categories=[list]), ['cough'])
], remainder='passthrough')
list of tuples
list or [1] or [1,2,5] or column

transformer.fit_transform(X_train) & transform on test

name1 → trf.named_transformers_

⇒

From sklearn import set_config
set_config(display='diagram') gives diagram of pipeline

★ Pipeline (list of tuples)

Pipe = Pipeline([('trf1', trf1), ('trf2', trf2)])

Pipe = make_pipeline(trf1, trf2, trf3)
Pipe.fit(X_train, y_train) ← If model is inside pipeline; other case fit_trans ← train, fit ← tst
Pipe.predict(X_test)
Pipe.name_steps['trf1'].transformers_[0][1].statistics_

⇒ cross_val_score(Pipe, X_train, y_train, cv=5, scoring='accuracy').mean()
★ params = {'trf5__max_depth': [1, 2, 3, 4, 5, None]} ← dict (can have more than one item)
grid = GridSearchCV(Pipe, params, cv=5, scoring='accuracy')
grid.fit(X_train, y_train)
grid.best_score
grid.best_params_ / grid.cv_results_

RandomSearchCV take random from params (It is not give best parameter maybe some many)

bi = Binarizer(copy=False, threshold=n)
bi.fit(-) ★ ⇒ bi.transform(-)
copy=F for change in original column itself
convert continuous values to binary 0 or 1
value < Threshold 0, > Then 1 by default threshold=1

⇒ kbin = ~~KBinDisc~~ KBinsDiscretizer(n_bins=10, encode='ordinal' or 'onehot', ~~strat~~ strategy='quantile')

kbin.n_bins_
.bin_edges_
new column with age lable can get by
1-10
10-20 ...
pd.cut(x=X_train['age'], bins=list of bin_edges_)

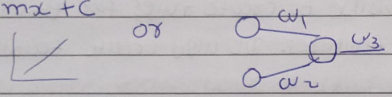
interview questions

3) Diff bet^n Parametric & non Parametric ML algo

i) Parametric ML algo: we make assumption of nature of the function Ex Linear regression

→ and it has fix number of parameter It does not depends on row of data linear reg, Linear SVM, Naive Bayes (assumption all independent comms one independent)

Ex $y = mx + C$



or

simple neural network

ii) Non parametric ML algo: No assumption

→ No fix para grow with number of row

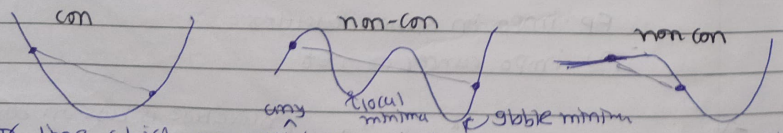
Ex Tress Other SVM with diff Kernel

KNN, DT, RF, xGBoost, Ada...

Complex neural networks

Para	Non-Para
1) simple func	1) complex
2) less data	2) more data
3) underfit	3) overfit

2) what is difference bet^n convex and non convex loss fun? what happens when we have a non convex function?



⇒ For line which connect two points has no part of function above that line

⇒ Strictly convex fun has only one minima

non-convex has more than one minima

⇒ when we have non convex function then some time solution stuck in local minima

6) give example where ~~mean~~ median is more impo than mean?

Ex
salary 3L, 5L, 8L, 2Cr
This kind of situation take median

⇒ when data has outliers take median

7) what do you mean by unreasonable effectiveness of data in ML?

⇒ as data increasses algo does not matter simple and oreate algo also give same performance in compare to complex and powerful algo.

8) what is Lazy Learning algorithum?
How it is diff from eager learning?
why is KNN A Lazy learning ML algo?

two type → Lazy: generalization of training data in theory, delayed until query is made by system.
→ eager: where system try to generalize the training data before receiving queries.

⇒ Lazy Learning Store training data because it need during prediction
Prediction can slow

⇒ eager learning not store data, fast prediction

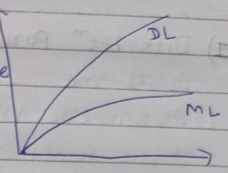
9) what is semi supervised ML?

types of ML → supervised
→ unsupervised
→ Reinforcement
→ semi supervised

⇒ supervised has labeld op
⇒ unsupervised has no labeled op
⇒ semi supervised
Ex google Photos cluster different Photos in ~~one~~ group and we need to label only ~~those~~ group.
each photo in group has same label

3) when use Deep Learning?

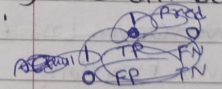
Favour ⇒ when need high performence
⇒ have large numbers of data
⇒ If problem is very complex and we don't have domain knowledge about it then use DL algo
Ex cat-Dog classification
⇒ it is not possible to create rules for this problem and ~~find~~ create features. DL algo create features & rules



against

⇒ Data is small then use ML
⇒ If you dont have cost (Hardware) to GPU
⇒ Interpretibility or explainability
Ex block account using DL can't explain why block account

4) give diff between FP & ~~FN~~ FN give example which when important



FP impo in spam classifier
FN impo cancer or not

5) Naive Bayes why "naive"? (naive base is base on bayes theorem)

⇒ Naive Bayes algo consider that data has no relationship.

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

$P(y|x) = P(x \cap y)$ (ignore denominator ∵ Prob)

$= P(x_1 \cap x_2 \cap y) = P(x_1 | x_2 \cap y)\, P(x_2 \cap y) = P(x_1|y)\, P(x_2|y)$

assumption that x_1 not depends on x_2 ; ignore $P(y)$

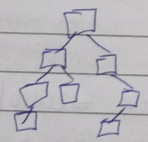
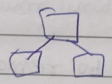
PAGE NO.:

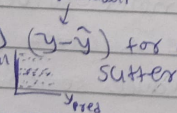
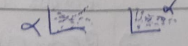
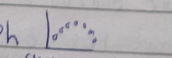
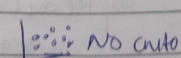
14) what is 'No Free Lunch' in ML?
⇒ without any assuption while making ML model, we can't say which model gives good result on perticuler dataset.

15) difference bet^n structured & unstructured data?

data
- struc: ⇒ tubular data; ⇒ organized; ⇒ easy to apply ML algo
- unstruc (qualitative): ⇒ non tubular Ex social media post, text file, sound file, video; ⇒ unorganized; ⇒ apply DL for unstructure data; ⇒ use in NLP, text mining, computer vision

17) difference bet^n Bagging VS Boosting

	take ML algo which Low Bias high variance	ML algo which High Bias low variance
1) type of model use	Decision Tree (Fully grown) DT 	Decision stump 
2) Sequential VS Parallel Learning	Bagging → Parallel; Boosting → sequential	
3) weightage of base learner	Bagging → same weightage	Boosting → different "

18) Assumptions of ~~ten~~ Linear regression
1) Linear Relationship bet^n input & op
2) no multi collinearity
3) Normality of Residuel ⇒ $(y-\hat{y})$ for every point distributed normaly (residual = $y-\hat{y}$)  scatter is uniform in X axis 
4) Homoscedasticity ⇒
5) No Autocorrelation of errors
There is no Patten on above graph  auto correlation  No auto correlation

PAGE NO.:

16) what is OOB error and how is it useful?
OOB = out of bag Evaluation
take ex of RF some samples are not selected for training (≈37%) we can test over model on this data (as validation set)

11) when you prefer DT over RF?
⇒ explainability
⇒ low computation power
⇒ If we know some feature is very important and training should contain those feature while training...

12) why logistic regression not logistic classification?
logistic regression is ~~linear~~ ~~re~~ linear regression with σ function. and we set some threshold If o/p > threshold then →1
⇒ logistic regression called ~~logistic~~ regression because It's give continous value so..

13) what is online ml?
and In what kind of scenario it ~~are~~ useful?
⇒ is technige in which train model on server (online).
⇒ This is called incremental learning
$m \rightarrow m' \rightarrow m'' \rightarrow m'''$
we partially train model not every model support but only some algo like SVM, SGD regression support.
advantage: less costly, faster training
⇒ Ex chatbots.
→ swift key → Keycord Android
→ youtube → once you watch video feed below that video change
→ concept drift Ex recommendation sys → sales, festivals, season

⇒ since we don't have all house data we can't find $f(x)$ but find $f'(x)$ which is close to $f(x)$

$f(x) - f'(x) =$ reducible error

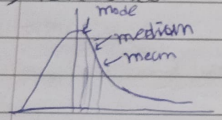
This error reduce as we get more data or using different models

21) ★ For which case outliers affect more?

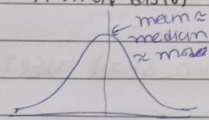
⇒ outliers are mostly affect on weight base algo like linear, Logistic, Adaboost, Deep learning

⇒ not much effect on tree base algo like RF, XGBoost, Gradient boosting

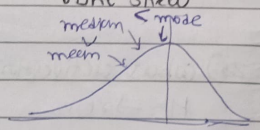
22) left skew



normal distri



right skew



Positive skew

Ex: wealth distri

symetric distri

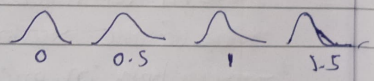
IRIS dataset

Negative skew

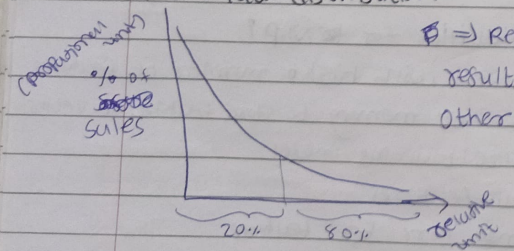
life span of human being

mode = most frequent value

$$\text{Skew} = \frac{n}{(n-1)(n-2)} \sum \left[\frac{(x-\bar{x})}{s}\right]^3$$



23) Power law distribution



Ex: ⇒ Relative change in one quantity results in proportional change in other quantity

1) 80% sales are coming from 20% of overall Product
2) 80% of windows crash due to 20% bugs
3) 80% data scientist use 20% software

⇒ because of multicollinearity can't explain importance of feature in output

Ex: some invention done by scientist only two scientist is from same background math so it is very difficult to find contribution of those two.

⇒ some time we want to find relation (Impossible) between feature and op so we use linear regression

⇒ Is multicollinearity always bad?

If want find feature impo then yes

If only prediction then NO, for only pred use better algo then linear regression

type of multicolin

Structural

⇒ If we create new feature and it contains multi colli then it's called structural multi col

data bias

⇒ data has multi coli

How to remove multi coli

→ increase data ⇒ remove one corre col ⇒ Lasso/Ridge regn

⇒ PCA

19) ★ Does multicollinearity affects all ML algo?

mostly impact on parumentric algo

20) ★ Reducible vs irreducible errors

error/noise

⇒ data has some noise so we add bias in $y = mx + b$ even if we have x can't predict y so we add bias this error can't be reduce so it's called irreducible error

⇒ Reducible error: $y = f(x) + \text{error (bias)}$ (b)

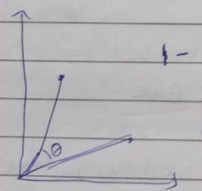
let we has 1000 house data (sample data) for prediction price but we want to create formula for population (Total houses)

27) what is assumption for SVM?
⇒ No assumption

28) Decision Tree
⇒ No assumption
⇒ Feature scaling not required (∵ DT is rule base approach instead of distance calculations)
⇒ can handle continous and categorical variable
⇒ use for Classi & Regression
⇒ No linear parameter not affect performance of DT. (DT is not curve base algo)
⇒ can handle missing value automatically (outliers not involve in split of DT)
In some paper ★ ⇒ usually robust to outliers and can handle automatically.
⇒ less Training time compared to RF (∵ train only one tree not Forest)

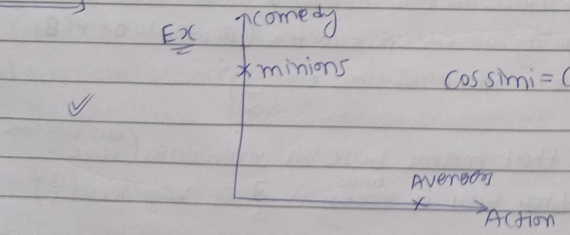
Disadv: over fitting, High variance, Unstable: Adding a new data point can lead to regenerate overall tree
Not suitable for large dataset

29) cosine similarity and cosine distance



1 - cosine similarity = cosine distance

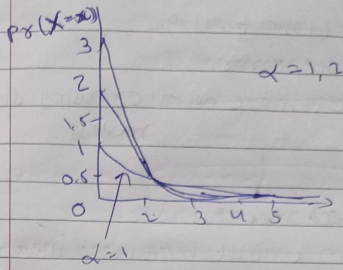
Cosine similarity = Cosθ



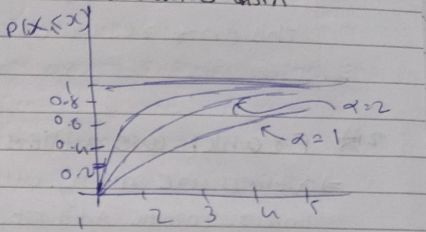
Cos simi = Cosθ = Cos 90 = 0

Log normal distribution: is continuous prob distribution of random variable whose logarithm is normal distribution

⇒ Pareto distribution is type of power law distribution



Cumulative distri



24) when to use Entropy and when gini impurity?

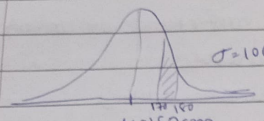
25) what activation fun should use?
HL → relu
output depends on problem For classification = binary = σ
multi = softmax
Regression = linear
⇒ HL when we use Relu use He for weight ini in HL
& use Xavier glorot on OP layer for of softmax
both in one model

26) why use Naive Bayes Classifier for NLP?
⇒ because it work very well with high number of features
algo like word2vec take large memory to store feature in vectors.
⇒ converge fast when we are training model
⇒ work well with categorical features
Disadv: assume there is not corr bet^n features

Adv: Can handle missing value data ($\because \sum_{i=1}^{n} P(x_i/y) = 1$)
⇒ robust to outliers
Just skip x which we don't know
(∵ take all prob for that feature)

let take example of normal (gausian distribution)

⇒ Normal dist is continous distribution



we know distri (parameters σ, μ)

⇒ chance of perticular event is occure is called probability

P(170 to 180) = Area under 170 to 180

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{z^2}{2}} \qquad z = \frac{x-\mu}{\sigma}$$

P(170 to 180) = 0.02

⇒ Now likelihood is

so let we take one point and its 100cm then what is likelihood that it follows normal dist μ=150cm & σ=10cm

$$L(\mu,\sigma \mid x=100) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

μ=150, σ=10, x=100

very small → $= 1.47 \times 10^{-7}$

$L(\mu,\sigma \mid x=130) = 0.005$

$L(\mu,\sigma \mid x=140) = 0.024$

$L(\mu,\sigma \mid x=200) = 1.49 \times 10^{-7}$

Probability = mesurement of chance that certain event occure out of all possible event $0 \le P \le 1$

Likelihood = in statistical context, likelihood is a function that measures the plausibility of a particular parameter value given some observed data. It quantifies how well a specific outcome support specific parameter values.

It quantifes how good one's model is, given set of data that has been observed

Probability discribe outcome while Likelihood describe models

30) Data Engineer Vs Data Analyst Vs Data scientist

Data warehouse } Data Engineer : ensure that data will be store in much more efficient way

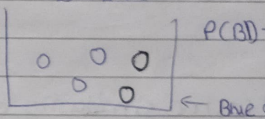
Data scientist
- Exploratory Data Analysis } Data Analyst: perform some EDA and statistical analysis on data and understand some more information from data → visualize data using power BI, Tableau
- using DL or ML →
  - Model creation
  - model deployment

31) Probability vs Likelihood

Parameter → event

event → Parameter

B = black, Bl = blue



⇒ Probability of Ball is Blue is = 3/5 — Bernauli distribution

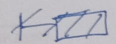
⇒ we take 5 Ball and let Those all are Blue

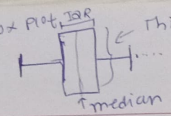
Bl Bl Bl Bl Bl

Base on this observe data or event we have to find the possibility of give parameter (P(Bl) = 3/5 or P(B))

So possibility = $\left(\frac{3}{5}\right)^5$

that means base on given data (all ball are Blue) probability of blue ball is 3/5 is very low $\left(\frac{3}{5}\right)^5$





PAGE NO.:

34) Time series cross validation (using rolling window approach)

Fold 1 1, 2, 3	1
validation 4	2
Fold 2 1, 2, 3, 4	1, 2
" 5	3
Fold 3 1, 2, 3, 4, 5	1, 2, 3
" 6	4

walk forward cross validation

35) why you take σ not σ^2?

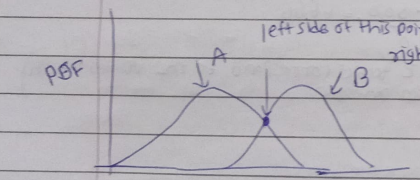
⇒ because σ^2 change the unit from original data so we prefer σ (SD) instead of variance

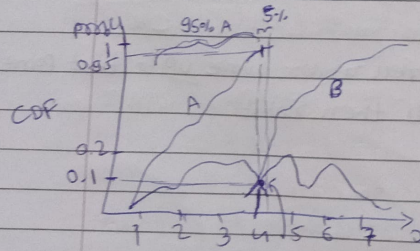
36) why variance prefer instead of mean abs deviation?

⇒ variance give smooth function

⇒ when outliers are more then use MAD

37) what is importance of PDF & CDF





If $x < \theta$ then A ← 95% correct

else If $x \geq \theta$ then B

B 10% wrong

90% correct.

PAGE NO.:

32) How handle imbalance dataset?

(↑ number of sample which is less) (vis versa) decrease more

⇒ over sampling, under sampling

⇒ class weights: give weight to increase importance of minority class

⇒ use evaluation matrix instead of accuracy

⇒ use Ensemble method Ex RF, Adaboost, Bagging

⇒ cost sensitive Learning:

Some algo have inbuild options for handling imbalance data.

For Ex svm allows for adding class weights to penalize misclassifying the minority class.

⇒ Data augmentation in minority class

⇒ collect more data

⇒ use stratified cross validation for find accuracy

33) How to make model robust to outliers?

⇒ remove outliers using z score, IQR, visual inspection using box Plot

⇒ Data transformation (using fun transformers)

⇒ use loss function which less sensitive to outliers Ex MAE instead of MSE

⇒ Ensemble techniques reduce effect of outliers

⇒ Data segmentation: If data contains mix outliers rich and outliers free regions build separate model for each segment.

⇒ Use median instead of mean

⇒ Normalize / scale Data

⇒ Cross-validation: Employ robust cross-validation like K-fold cross validation to evaluate the model performance and ensure that generalize well.

40) How to handle Imbalance dataset in deep learning?

⇒ use weighted neural networks

weights = {0:1, 1:550} If 0 is 550 time more in o/p then assign 1 to 0 & 550 to 1

model.fit(x_train, y_train, class_weight=weights, epochs=10)

41) How to do Hyper parameter tuning in ML & DL?

⇒ ML ⇒ GridSearchCV, RandomSearchCV (Hyper parameters tuning using iterations)

dict = {'C': [0.1, 0.5, 1, 10]
'G': ['linear', 'rbf']
}

grid = GridSearchCV(estimator=SVM, dict, cv=5, scoring='accuracy')
grid.fit(x_train, y_train)
grid.predict(x_test)

n_jobs = -1 to make process faster

42) How can you make process faster while using hyper parameter tuning?

using n_job = -1

43) when use normalization and when standardization

⇒ Normalization scale data from range 0 to 1

→ use normalization when features are in diff range and you want to bring them all to a comparable range.

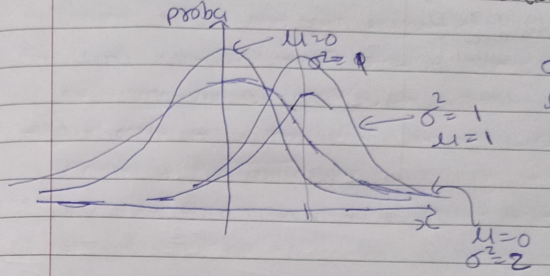
→ It specially useful when algo relies on the magnitude of features and works well with features that have meaningful minimum & maximum

⇒ Stan = $\mu=0, \sigma^2=1$

→ use standardization when data has varying distributions and you want to make sure all features have comparable mean & σ^2

→ Stan is commonly use ML algo (SVM, K-neurest neigh) relies on the distance bet^n data points. as it centers the data at zero and give equal impo of +ve & -ve deviations from mean.

38) How σ and μ affect distribution graph.



σ^2 = spread of data

μ = centrality of model

For $\sigma^2=1$ $\mu=0$

-1σ to 1σ 68%.

-2σ to 2σ 95% (68+27)

-3σ to 3σ 99% (68+27+4)

39) Imagine a group of 300 students took a test. Rohit scored 700 out of 1000. The avg score was 500 and standard deviation was 120. assuming test score follow normal distribution find out how well rohit performed in comparision to his peers.

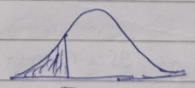
$\sigma = 120$

$\mu = 500$

z for rohit is: $z = \frac{x-\mu}{\sigma}$

$z = \frac{700-500}{120} = 1.666$

From z table (-ve score in z table correspond to the values which are less then mean)



$z = 0.9515$

z score gives area under curve up to that point

so 95% student score less then rohit

Q5) In what case use SGDRegression over linear regressior

⇒ linear regressior gives exact value but it use matrix inversion so it is very costly computation $(X^TX)^{-1}X^Ty$

⇒ SGD Regressior is useful when data is very large

⇒ Choice is dependion size of data and computational resources avaliable.

Q6) Is Ridge Regression use Matrix inversion?

No,

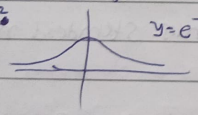
$$\beta = (X^TX + \lambda I)^{-1}X^Ty$$

while the matrix inversion appears in the eqn for optimal β modern implementation of ridge regression in libraries like Scikit-learn we more effient numerical methods, such as SVD (singular value decomposition) to solve this problem without explicity performing matrix inversion

Q7) Kernal Function (linear, rbf, Poly, sigmoid, Precomputed)

RBF: Radial Bassis function:

$y = e^{-(x+b)^2}$



$y = e^{-x^2}$ ← x is vector sqwr of all features

x_1, x_2	z

$z = e^{-(x_1^2 + x_2^2)}$

⇒ In Summary norma use when you want feature [0,1] & stand use when you want $\sigma^2=1$, $\mu=0$

⇒ Stand can be more robust since it less influence by extreme value compare to norma

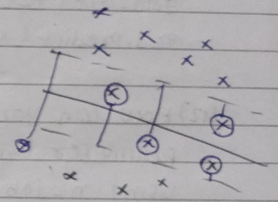
⇒ Although it is not fix which use when some algo might perform better with specific scaling method

It's good to try both and select best

Q8) what is diff bet^n hard & soft margin?

hard margin = $\underset{w^*, b^*}{\arg\max} \frac{2}{\|w\|}$ such that $y_i(w^Tx_i+b) \geq 0$

$= \underset{}{\arg\min} \frac{\|w\|}{2}$



Soft margin: allow outliers in model

$\underset{(w^*,b^*)}{\arg\min} \frac{\|w\|}{2} + C\sum_{i=1}^{n} \varepsilon_i$ ← hing loss

↑ margin error + ↑ classification error

HyperParam

→ If hyperparameter C↑ give more focus to reduce misclassified point

→ C↓ more focus to give Large margin with some misclassified points

$\varepsilon_i = \text{Loss} = \max[0, 1-y_i(w^Tx_i+b)]$

Same Ancology as Logistic regression

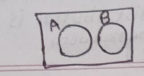
Loss Term → logistic loss + $\lambda\|w\|$ ← regularization Term

$C \propto \frac{1}{\lambda}$

similare to hing loss ε

$\left(\underset{\min}{\arg} \frac{\|w\|^2}{2}\right)$ similar to

Mutually Exclusive events: A & B never occur together
A, B $P(A \cap B) = 0$
$P(A|B) = \frac{P(A \cap B)}{P(B)} = 0$
Head on 1^{st} toss given T in 1^{st} toss



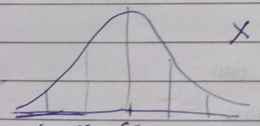
Independent event
If event does not depend on 2^{nd} event

51) why normal distribution is very important?
⇒ Commonly in Nature many natural phenomina follow a normal distribution such as height of people, weights of objects IQ score of population. and many more
⇒ Normal distribution provide a convenient way to model and analyse such data.
⇒ Now we have enough study and information about normal distribution so we want dataset to be normaly distributed.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Area of $e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} = \sigma\sqrt{2\pi}$ so we ~~divid~~ by $\sigma\sqrt{2\pi}$ to get Area = 1

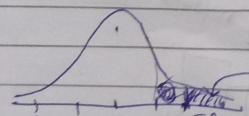
52) Suppose the heights of adult males in certain population follow a normal distn which $\mu = 68$ inches & $\sigma = 3$ inch what is prob that randomly selected adult male from the population is taller than 72 inches?



$X \sim N(68, 3)$

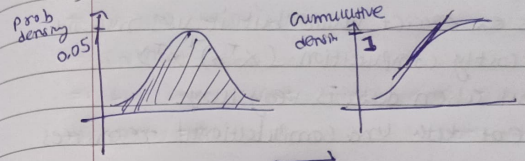
$Z = \frac{72-68}{3} = 1.33$

⇓ convert to standard normal distn



$= 1 - \text{Area}(-\infty \text{ to } 1.33)$
$= 1 - 0.90824$
≈ 9.2

48) How to find CdF to PDF?



integration →
← differentiation

49) what is advantage of standard Normal distribution
⇒ using standard normal distribution allow to compare different distribution with each other

Normal distri $X \sim (\mu, \sigma)$
standard Normal dist $Z \sim (0, 1)$ $\quad \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$

How to convert Normal distri to standard normal distri.
convert each column to $Z = \frac{X-\mu}{\sigma}$

so final data has standard normal distri ($\mu = 0, \sigma = 1$)

50) F1 score (F1 is for binary & multi class classification)
F1 is used evaluate the performance of binary classification model. it combine precision and recall both
⇒ combine both P, R in one and use when imbalance bet^n classes.
⇒ F1 = 1 means perfect P&R
F1 = 0 means poor P&R

T-test work good with smaller sample size

PAGE NO.:

59) what is concept drift?
⇒ when dependent variable change after some time of build model
data drift: independent " " "

QQ plot: A normal Prob Plot Plots the Observed data against the expected values of normal distri. If data fall along a straight line the distribution is likely to be normal
(quantile -quantile plot) [x, y] data, Theoretical distri

PAGE NO.:

53) How to find data is normaly distri or not?
⇒ Visual inspection
⇒ QQ plot
⇒ statistical test like shapiro-wilk, Anderson-Darling test
⇒ Skew = 0 Kolmogorov-Smirnov test

51) Does QQ plot only detect normal distribution?
⇒ No, it can detect any use y_quntile diff

55) How to check distribution is Pareto?

$$y = \frac{\alpha x_m^{\alpha}}{x^{\alpha+1}}$$

take log both side and check graph If it is ⌊ then distribution y is pareto

⇒ OR we can ~~com~~ check by using QQ plot

56) Proof of CLT
NO, Proff its exist in nature

57) diff bet^n parameter & statistics.
⇒ Parameter characteristic of Population
statistic " " sample

58) diff bet^n bar graph and histogram