① what is central Limit theorem?

→ central limit theorm is important for statistics because it allow us to safely assume that sampling distribution of sample mean will be follow gaussion distribution in most case.

This means that we can take advantage of statistical technique that assume a normal distribution

suppose $[S_1, S_2, S_3 \dots\dots\dots S_m]$ → samples

$\downarrow$

$\bar{x}_i = [\bar{x}_1, \bar{x}_2, \bar{x}_3 \dots\dots\dots \bar{x}_n]$ → sample distribution of sample mean

and μ → mean σ = variance and n = no of popution

the $\boxed{\bar{x}_i \sim N(\mu, \sigma/\sqrt{n})}$ → variance for sample mean

; Sample Mock Interview Question:

1. what is optimization equation of GBDT ?

Ans:

Input: training set $\{(x_i, y_i)\}_{i=1}^n$, a differentiable loss function $L(y, F(x))$, number of iterations M.

Algorithm:

1. Initialize model with a constant value:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma).$$

2. For m = 1 to M:
   1. Compute so-called *pseudo-residuals*:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n.$$

   2. Fit a base learner (e.g. tree) $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^n$.
   3. Compute multiplier γ_m by solving the following one-dimensional optimization problem:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right).$$

   4. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

additive model

$$F(M) = h_0(x) + \gamma_1 h_1(x) + \gamma_2 h_2(x) \cdots + \gamma_K h_K(x)$$

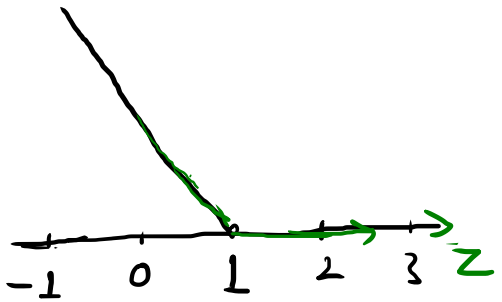
where optimization problem will be

$$\gamma^* = \underset{\gamma}{\operatorname{argmin}} \left(L(y_i, F_{m-1}(x) + \gamma_m h_m(x)) \right)$$

Loss can be any phenomenon choice.

2. what is formulation for hinge Loss.

Ans:



hinge loss = max(0, 1-z)

means if $z \geq 1$ then 0

else 1 - z

3. what is train complexity of KNN?

Ans:- training of KNN does not cost much because at training time, our data point are stored with some datastructe way so that at the time of testing. It will take less access time.

train_time_complexity = $o(nd)$

4. what is Test time of brute force KNN

Ans:

test time complexity = $O(k * n * d)$

5. what is test time complexity of kd tree KNN ?

Ans: Test_time_complexity = $O(k . \log n)$ = $O(2^d . k . \log n)$

6. How will we regularize KNN Model ?

→ in KNN, K = no of neighbour can be worked to penalize the the model. but there is no explicit term for regularization

7. Which of these model are preferable when we have low complexity power ?
   a. SVM
   b. KNN
   c. Linear Regressions
   d. XGboost

(a) SVM → $O(n^2)$ [train] & $O(kd)$ [runtime]

(b) KNN → $O(k.n.d)$ $O(d)$ [test]

(c) LR → $O(nd)$ [train]

(d) XGboost → $O(n \log n . d . M)$ [train] & $O(depth*M)$ (runtime)

8. what is Laplace smoothing

Ans. Laplace smoothing is a smoothing technique which handles the problem of zero probability in naive bayes

$$P(w|y=1) = \frac{\#\text{of event w and } y=1 + \alpha}{\text{Total No of even } y=1 + k\alpha}$$

9. How will we regularize Naive bayes?

→ Laplase term α will be worked as regularizer the Naive baiyes so we need to hypertune the α.

10: Can we solve dimensionality reduction with SGD?

Ans: Yes

11. which of these will be doing more computation GD and SGD?

Ans: SGD often converges much faster than GD but error functis not as minimized as GD

→ SGD will take more compution than GD.

12. If A is a matrix of size (3,3) and B is a matrix of size (3,4) how many numbers of multiplications that can happen in the operations A*B ?

Ans = $3 \times 3 \times 4 = 36$

13. What is optimization equation of Logistic regression?

Ans:

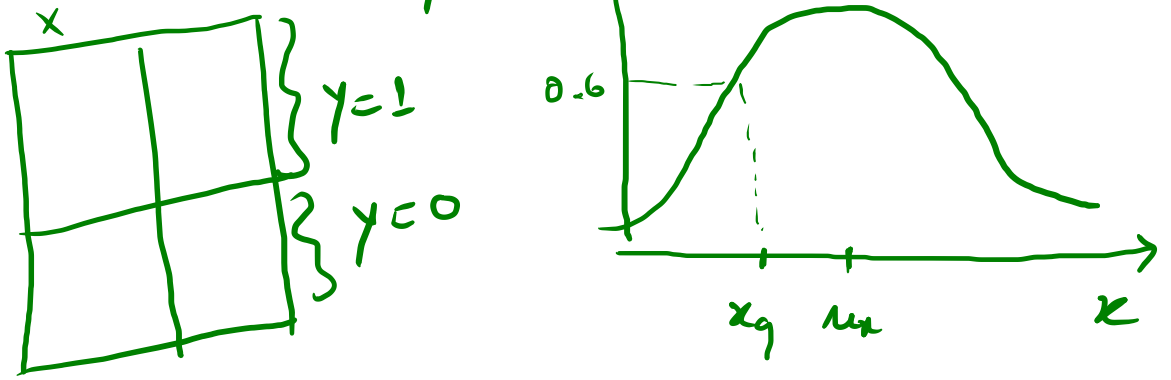
$$w^* = \underset{w}{\operatorname{argmin}} \left(\log(1 + \exp(-z)) \right)$$

where z = signed distance $= y_i w^T x_i$

14. How will we calculate $P(x|y=1)$ for Gaussian NB ?

Ans: in Gaussian NB, our assumption will be that feature follows the Normal distribution so

we can calculate $P(x|y=1)$ from the pdf of feature where $y=1$



So for given point x we can calculate

$$P(x_q|y=1) = 0.6$$

15. Steps for proportional sampling ?

→ proportional sampling is a method for picking an element proportional to its weight

→ higher the weight i.e better chance to get selected

: Steps :

1. normalize the array with sum of array value
2. Calculate cumulative sum
3. pick any uniform random value (0, 1.0)
4. pick the index where thes random value lies.

16. what is hyperparameter for kernel SVM?

→ ① polynomial kernel:

$$K(x_1, x_2) = (x_1^T x_2 + 1)^d$$

Here d is hyperparameter

② RBF kernel:

$$K(x_1, x_2) = \exp\left(\frac{-\|x_1 - x_2\|^2}{2\sigma^2}\right)$$

Here σ will be hyper parameter

③ Normal optimization

$$(w^*, b^*) = \underset{w,b}{\operatorname{argmin}} \left(\frac{\|w\|}{2} + C * \frac{1}{n} \sum \varepsilon_i\right)$$

such that $y_i (w^T x + b) \geq 1 - \varepsilon_i \forall i$

Here C will be hyperparameter

17. hyperparameter in SGD with hinge Loss

→ SGD requires a number of hyperparameter such as

1. regularization hyperparameter
2. No of epoch
3. Learning decay can also be an hyperparameter

18. Is hinge loss differentiable if not then how can we modify that so we can use in SGD?

→ Rewrite hinge loss in terms of w as $f(g(w))$ where $f(z) = \max(0, 1 - y\,z)$ and $g(w) = \mathbf{x} \cdot \mathbf{w}$

Using chain rule we get

$$\frac{\partial}{\partial w_i} f(g(w)) = \frac{\partial f}{\partial z} \frac{\partial g}{\partial w_i}$$

First derivative term is evaluated at $g(w) = x \cdot w$ becoming $-y$ when $\mathbf{x} \cdot w < 1$, and 0 when $\mathbf{x} \cdot w > 1$. Second derivative term becomes x_i. So in the end you get

$$\frac{\partial f(g(w))}{\partial w_i} = \begin{cases} -y\,x_i & \text{if } y\,\mathbf{x} \cdot \mathbf{w} < 1 \\ 0 & \text{if } y\,\mathbf{x} \cdot \mathbf{w} > 1 \end{cases}$$

Since i ranges over the components of x, you can view the above as a vector quantity, and write $\frac{\partial}{\partial w}$ as shorthand for $(\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \ldots)$

19. what is ADADELTA or RMSprop

→ adagrad α can become very large resulting very slow convergence.

→ Adadelta take exponential weighted average of gradient of squares instead of simple sum to avoid large alphas avoiding slow convergence.

$$w_t = w_t - \eta'_t g_t$$

$$\eta'_t = \frac{\eta}{\sqrt{eda_{t-1} + \varepsilon}}$$

This term control the growth of denominator

$$eda_{t-1} = \gamma\, eda_{t-2} + (1-\gamma)\, g^2_{t-1}$$

typically $\gamma = 0.95$

20. what is ADAM?

→ in adadelta, we are storing exponetial weighted average of g_t^2

but the idea in ADAM is, what if we use exponential weighted average of g_t instead of g_t^2.

mean – 1st order momentum

var – 2nd order momentum

Formulation:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1-(\beta_1)^t}$$

$$\hat{v}_t = \frac{v_t}{1-(\beta_2)^t}$$

$$w_t = w_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

$\beta_1 = 0$ then Adagrad

$\beta_1 = \beta_2 = 0$ then Adadelta

21. Difference between ADAM and RMSProp

→ RMSProp : eda (estimated decay average) of g_t^2

ADAM: eda of g_t

22. What is the maximum and minimum value of gradient of sigmoid function?

→ $$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma'(z) = \sigma(z) * (1-\sigma(z))$$

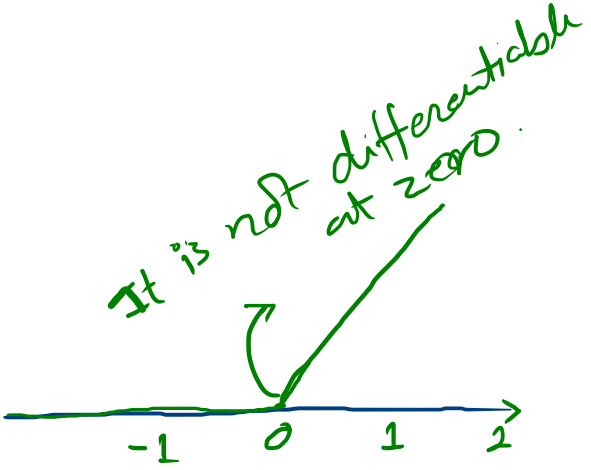
and $\sigma'(z)$ lies between 0 to 0.25

and $\sigma(z)$ lies between 0 to 1.

23. what is reLU ? is it differentiable ?

→ ReLU is an activation function. which is used extensively.

$$R(z) = \max(0, z)$$



It use to overcome vanishing Gradient problem

$f'(z) = \{0 \text{ if } z<0 \text{ or } 1 \text{ if } z>0\}$

Note: ① No vanishing gradient and no exploding gradient but There is dead activation below zero.

② Relu is non linear, and not differentiable at zero

Q. what is the reason that neuron got dead in ReLU?

Q. How to check neuron become dead?

24. what is F_1 Score?

→ F_1-score is measure of test's accuracy. and It is harmonic mean of precision and recall.

$$F_1\text{-score} = \frac{2 * Pr * Re}{Pr + Re}$$

where $\text{Precision} = \frac{TP}{TP + FP}$ and $\text{recall} = \frac{TP}{TP + FN}$

commonly used for evaluating

1. information retrieval model
2. NLP model

25. what is precision and Recall?

→

precision	Precision is the fraction of true positive examples among the examples that the model classified as positive. In other words, the number of true positives divided by the number of false positives plus true positives.
recall	Recall, also known as sensitivity, is the fraction of examples classified as positive, among the total number of positive examples. In other words, the number of true positives divided by the number of true positives plus false negatives.

$$\text{precision} = \frac{TP}{TP+FP} \quad \text{and} \quad \text{recall} = \frac{TP}{TP+FN}$$

26. Name a few weight initialization technique?

1. He initialization
2. Xavier/Glorot
3. Random uniform initialization

27. Which of these will have more numbers of tunable parameters?

a. 7,7,512) ⇒ flatern ⇒ Dense(512)

b. (7,7,512) ⇒ Conv (512,(7,7))

→ These assertions show how the numbers of parameters of the layers depend on input, output, and each other: `output_size * (input_size + 1) == number_parameters` use for dense Layer

Conv Layer

These assertions show how the numbers of parameters of the layers depend on input, output, and each other: again, `output_size * (input_size + 1) == number_parameters`. For convnets, `output_channels * (input_channels * window_size + 1) == number_parameters`. For the this particular example, `window_size=3*3`

(a) 7,7,512) ⇒ flatten ⇒ dense(512)

= 512 x (7x7x512 +1)

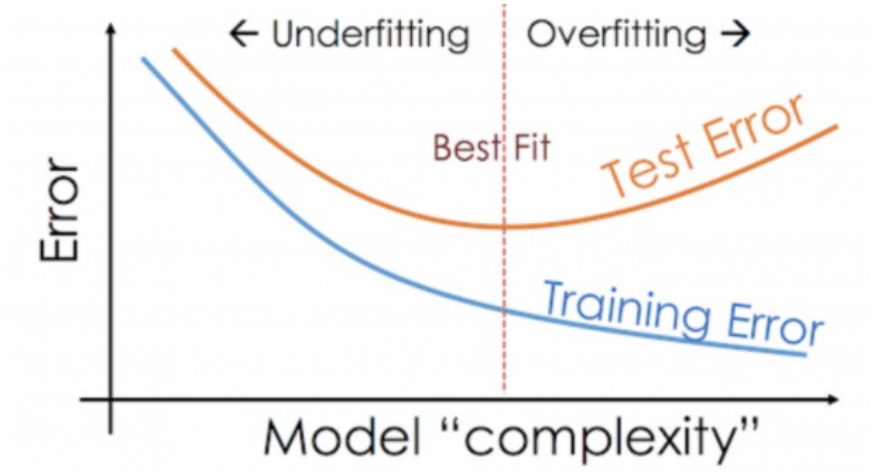
= 12845,568

(b) (7,7,512) ⇒ Conv (512,(7,7))

⇒ 512 x (512 *(7x7) + 1)

⇒ 12845,568

28. what is the overfitting and underfitting ?



overfitting and underfitting are technique to understand How good our model is.

29. what do you do if a deep learning model is overfitting?

→ add regularization, dropout and batchNormalization

30. what is batch Normalization Layers.

→ we generally Normalize our input before feeding the Model. and in deep MLP, we feed data with mini batch. so small changes at input layer into batch can lead to large change at last layer. This problem is known as 'Internal covariance shift".

to overcome from this problem, we use Batch Normalization before every layer, which ensure same distribution for every layer.

→ fast convergence
→ weak regularization
→ ensure normalized data to each layers

31. write keras code to add BN Layers ?

→

```
tf.keras.layers.BatchNormalization(
    axis=-1, momentum=0.99, epsilon=0.001, center=True, scale=True,
    beta_initializer='zeros', gamma_initializer='ones',
    moving_mean_initializer='zeros',
    moving_variance_initializer='ones', beta_regularizer=None,
    gamma_regularizer=None, beta_constraint=None, gamma_constraint=None,
    renorm=False, renorm_clipping=None, renorm_momentum=0.99, fused=None,
    trainable=True, virtual_batch_size=None, adjustment=None, name=None, **kwargs
)
```

32. No of tunable parameter in BN Layer.

These 2048 parameters are in fact `[gamma weights, beta weights, moving_mean(non-trainable), moving_variance(non-trainable)]`, each having 512 elements (the size of the input layer).

As you can read there, in order to make the batch normalization work during training, they need to keep track of the distributions of each normalized dimensions. To do so, since you are in `mode=0` by default, they compute 4 parameters per feature on the previous layer. Those parameters are making sure that you properly propagate and backpropagate the information.

So `4*512 = 2048`, this should answer your question.

33. what is convolution operation ?

→ It is same as dot product of vectors but convolution operation performs on matrix

= multiplication + addition

34. No of parameters in convolution neural Network given in architecture.

→ No of parameter = output Layers * (Input Layers * (window size * wind-size) + 1)

35. what are input required to calculate the average f_1 - score ?

→ input : ① precision and recall
② TP, FP and FN

Note: inverted index used for every search engine

36. what macro average f_1 score for 5-class classification problem.

→ There are two type of f1-score for multi class classification

① micro f_1-score

② macro f_1-score

$$\Rightarrow \text{micro-}f_1\text{-score} = \frac{2 * \text{micro\_precision} * \text{micro recall}}{\text{micro\_precision} + \text{micro recall}}$$

$$\text{micro precision} = \frac{\sum_{i=1}^{c} TP}{\sum_{i}^{c} TP + \sum_{i}^{c} FP} \qquad \forall\, c = \{0, 1 \cdots c\}$$

$$\text{micro recall} = \frac{\sum TP}{\sum TP + \sum FN}$$

49. How to decrease the test time complexity of a Logistic regression model.

→ Logistic regression is mostly used for low latency problem but we want to speed up more to test time. so we can use a trick like we can store weight parameter in cache memory rather than storing into RAM.

50, what is need of sigmoid function in Logistic Reg?

→ sigmoid has tempering behaviour which used for squashing. because if we don't use sigmoi we may be get impacted from outlier easily.

⇒ macro f_1-score

The Macro F1-score is defined as the mean of class-wise/label-wise F1-scores:

$$\text{Macro F1-score} = \frac{1}{N}\sum_{i=0}^{N} \text{F1-score}_i$$

where *i* is the class/label index and *N* the number of classes/labels.

37. How can you get probabilities for RF classifier outputs?

→ we can use callibration

38. is the calibration classifier required to get probability value for Logistic regression?

→ Yes, for actual probability we can use callibration classifier. even Logestic regression return probability value but that are not actual probability.

39. How does kernel svm work in test time?

→ we use

$$\sum_{i=1}^{n} \alpha_i y_i x_i^T x_q$$

where α_i is support vector and it's value will be 1 for support vector and zero for other vector

40. what kind of base Learner are foreferable in random forest classifier?

→ base Learner should be with high variance and Low bias.

41. How does bootstraping works in RF classifiers?

→

Random Forest = row wise bootstrap + col wise bootstrap
+ base Learner model

42. difference between one vs rest and one vs one?

- Binary classification models like logistic regression and SVM do not support multi-class classification natively and require meta-strategies.
- The One-vs-Rest strategy splits a multi-class classification into one binary classification problem per class.
- The One-vs-One strategy splits a multi-class classification into one binary classification problem per each pair of classes.

Example for one vs rest:

For example, given a multi-class classification problem with examples for each class '*red*,' '*blue*,' and '*green*'. This could be divided into three binary classification datasets as follows:

- **Binary Classification Problem 1**: red vs [blue, green]
- **Binary Classification Problem 2**: blue vs [red, green]
- **Binary Classification Problem 3**: green vs [red, blue]

Example of one vs one?

For example, consider a multi-class classification problem with four classes: '*red*,' '*blue*,' and '*green*,' '*yellow*.' This could be divided into six binary classification datasets as follows:

- **Binary Classification Problem 1**: red vs. blue
- **Binary Classification Problem 2**: red vs. green
- **Binary Classification Problem 3**: red vs. yellow
- **Binary Classification Problem 4**: blue vs. green
- **Binary Classification Problem 5**: blue vs. yellow
- **Binary Classification Problem 6**: green vs. yellow

The formula for calculating the number of binary datasets, and in turn, models, is as follows:

- (NumClasses * (NumClasses – 1)) / 2

42. Which one is better one vs rest or one vs one?

→ I think one vs rest will be good because for this we have to train c model where c is no of class.
but in case of one vs one we have to train
$c + (c-1)/2$ model.

44. what will happen if gamma(c) increases in RBF Kernel SVM?

`gamma` is a parameter of the RBF kernel and can be thought of as the 'spread' of the kernel and therefore the decision region. When `gamma` is low, the 'curve' of the decision boundary is very low and thus the decision region is very broad. When `gamma` is high, the 'curve' of the decision boundary is high, which creates islands of decision-boundaries around data points. We will see this very clearly below.

`C` is a parameter of the SVC learner and is the penalty for misclassifying a data point. When `C` is small, the classifier is okay with misclassified data points (high bias, low variance). When `C` is large, the classifier is heavily penalized for misclassified data and therefore bends over backwards avoid any misclassified data points (low bias, high variance).

45. Explain linear regression.

→ Linear regression is the type of supervised learning. In this, our model will predict y which will be continuous random variable.

suppose $X \in \mathbb{R}^d$ and $Y \in \mathbb{R}$

so we try to best fit our model with parameter w. so that error between predicted and actual will be less.

$$L(w^*) = \underset{w}{\operatorname{argmin}} \sum_i (y_i - \hat{y}_i)^2$$

where

$$\hat{y}_i = w^T x_i + w_0$$

46. what is difference between one hot encoding and a binary BOW?

→ one hot encoding is generally used to convert categorical variable to vector form.
but binary BOW can be used for converting text document as well as categorical variable into vector form.

47. Kernal svm and linear svm (SGD classifier with hinge loss). Which has low latency and why.

→ SGD with hinge loss < linear SVM < Kernel SVM

in this we can store all weight value and can directly multiply with query x_q.

Need to figure out

in this we have to do multiplication for every support vector

48. Explain Bayes theorem.

→ Formula For Bayes' Theorem

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

This is conditional probability

where:

$P(A)$ = The probability of A occurring

$P(B)$ = The probability of B occurring

$P(A|B)$ = The probability of A given B

$P(B|A)$ = The probability of B given A

$P(A \cap B))$ = The probability of both A and B occurring

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

→ Bayes theorem

[In probability theory, It is related to conditional probability of two random variable.]

51. why we need calibration ?

We calibrate our model when the probability estimate of a data point belonging to a class is very important.

Calibration is comparison of the actual output and the expected output given by a system. Now let me put this in the perspective of machine learning.

In calibration we try to improve our model such that the distribution and behavior of the probability predicted is similar to the distribution and behavior of probability observed in training data.

52. what is mean average precision ?

> *AP is averaged over all categories. Traditionally, this is called "mean average precision" (mAP). We make no distinction between AP and mAP (and likewise AR and mAR) and assume the difference is clear from context.* COCO Evaluation

Note: It is generally used for object detection problem.

The mean Average Precision or mAP score is calculated by taking the mean AP over all classes and/or overall IoU thresholds, depending on different detection challenges that exist.

53. Why do we need gated mechanism in LSTM?

Long Short-Term Memory (LSTM) is one of the most widely used recurrent structures in sequence modeling. It uses gates to control information flow in the recurrent computations.

Preserving the long term dependencies in the network is done by its Gating mechanisms. The network can store or release memory on the go through the gating mechanism.

1. forget gate: Control what information to throw away from memory.
2. Input gate: Control what new info is added to cell state from current input
3. output gate: Conditionally decide what to come out from memory.

54. what is stratified sampling?

In a **stratified sample**, researchers divide a population into homogeneous subpopulations called *strata* (the plural of *stratum)* based on specific characteristics (e.g., race, gender, location, etc.). Every member of the population should be in exactly one stratum.

Each stratum is then sampled using another probability sampling method, such as cluster or simple random sampling, allowing researchers to estimate statistical measures for each sub-population.

55. How can we compare two distribution

→ we have many technique through which we can compare two distribution

① Q-Q plot

② KS test

56. what will be happened to train time of k-means of data has very high dimension.

→ as we know that time complexity of k-mean

$$= o(n.k.d.i) \Rightarrow o(nd)$$

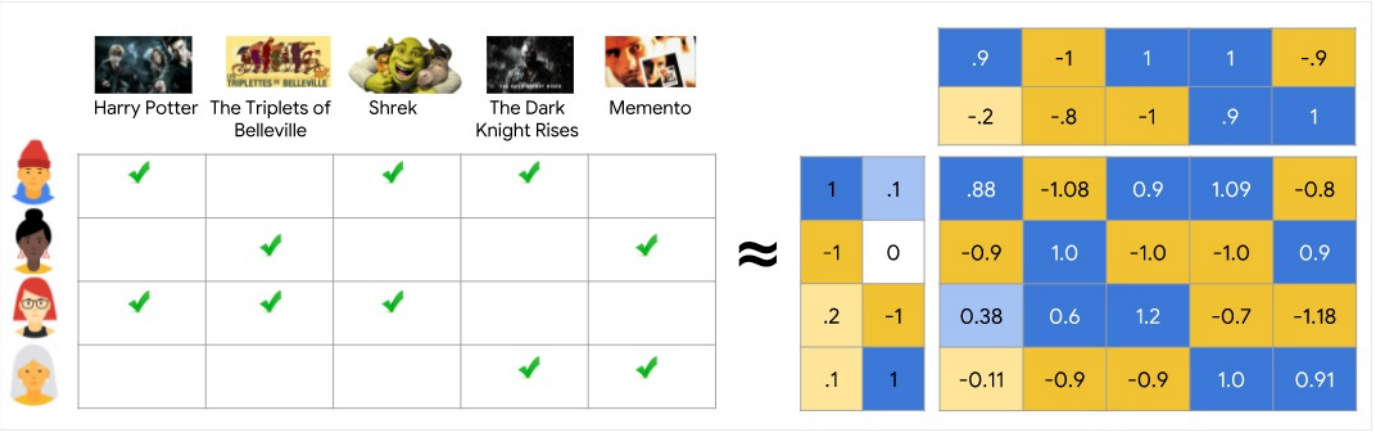
So d increase then time complexity will increase Even we know we use euclidian distance which will be got impacted.

57. If you have 10mill records with 100dimension each for a clustering task. Which algorithm will you try first and why ?

58. What is matrix factorization? Explain with example.

Matrix factorization is a simple embedding model. Given the feedback matrix A $\in R^{m \times n}$, where m is the number of users (or queries) and n is the number of items, the model learns:

- A user embedding matrix $U \in \mathbb{R}^{m \times d}$, where row i is the embedding for user i.
- An item embedding matrix $V \in \mathbb{R}^{n \times d}$, where row j is the embedding for item j.



The embeddings are learned such that the product UV^T is a good approximation of the feedback matrix A. Observe that the (i, j) entry of $U.V^T$ is simply the dot product $\langle U_i, V_j \rangle$ of the embeddings of user i and item j, which you want to be close to $A_{i,j}$.

Matrix factorization is a class of collaborative filtering algorithms used in recommender systems. Matrix factorization algorithms work by decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices.

59. which algorithm will give high time complexity if you have 10 million records for a clustering task?

→ K-mean

60. Difference between GD and SGD.

→ Gradient descent:

So, Gradient descent requires calculating the gradient using the entire dataset to perform updates to the model's parameters. In practice, the computational cost of Gradient descent can be very high and the time taken to reach the optimal weight vector w can be very long.

→ Stochastic GD:

Stochastic Gradient Descent (SGD) is a variation of Gradient descent that randomly samples **one** training sample from the dataset to be used to compute the gradient per iteration. Stochastic Gradient Descent works well because we are using just one data point to calculate the gradient, update the weight vector w, and compute the loss function value.

61. which one will you choose GD or SGD? why?

→ It depends on our requirment

① if dataset is small and we have enough computational power then I would like to choose "GD". because it will reach to optimal solution very fast (less iteration)

② If dataset is large and haven't enough computational power then we will use SGD.

optimal value of GD ≃ optimal value by SGD

62. Why do we need reptitive training of model?

→ to avoid degradation of predicted due to outdated data. so it's better to practice repetitive training of model.

63. How do you evaluate the model after production-ization?

→ I think we will check distribution of training data over time and we should evalute the model with new data.

64. Explain Gini impurity.

Gini Impurity is the probability of *incorrectly* classifying a randomly chosen element in the dataset if it were randomly labeled *according to the class distribution* in the dataset. It's calculated as

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

where C is the number of classes and $p(i)$ is the probability of randomly picking an element of class i.

66. Explain entrophy.

The entropy measures the "amount of information" present in a variable. Now, this amount is estimated not only based on the number of different values that are present in the variable but also by the amount of *surprise* that this value of the variable holds.

$$\mathbf{H(X) = -\Sigma\,(p_i * \log_2 p_i)}$$

$$Entropy\,(p) = -\sum_{i=1}^{N} p_i \log_2 p_i$$

67. How to do multi class classification with RF?

→ one vs rest is not required as entrophy takes all category while calculating them.

68. What is k-fold cross validation?

→ **K-Fold Cross Validation** is a common type of cross validation that is widely used in machine learning.

K-fold cross validation is performed as per the following steps:

1. Partition the original training data set into k equal subsets. Each subset is called a **fold**. Let the folds be named as $f_1, f_2, ..., f_k$.
2. For i = 1 to i = k
   a. Keep the fold f_i as Validation set and keep all the remaining *k-1* folds in the Cross validation training set.
   b. Train your machine learning model using the cross validation training set and calculate the accuracy of your model by validating the predicted results against the validation set.
3. Estimate the accuracy of your machine learning model by averaging the accuracies derived in all the *k* cases of cross validation.

In the k-fold cross validation method, all the entries in the original training data set are used for both training as well as validation. Also, each entry is used for validation just once.

Generally, the value of *k* is taken to be 10, but it is not a strict rule, and *k* can take any value.

69. what is need for CV?

→ **Cross Validation** is a very useful technique for assessing the performance of machine learning models. It helps in knowing how the machine learning model would generalize to an independent data set. You want to use this technique to estimate how accurate the predictions your model will give in practice.

70. How do you to CV for text classification problem using random search.

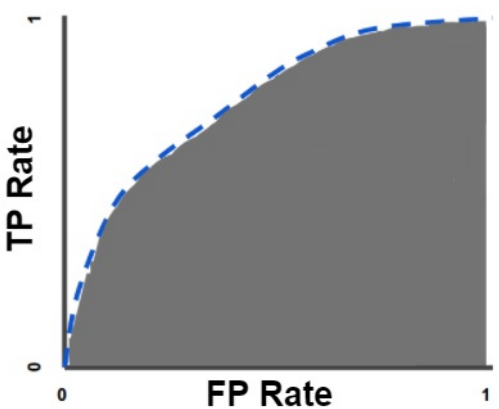
71. Assume We have very high dimension data. Which model will you try and which model will be better in a classification problem.

→ ① Logistic regression - because it is useful for Low Latency problem. = old)

② LDA + SVM -
↳ dimension reduction technique

72. what is AUC ?

→ **AUC** stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).



It's used for binary classification problem only.

73. Tell me one business case where recall is more important than precision?

→

Recall is more important where Overlooked Cases (False Negatives) are more costly than False Alarms (False Positive). The focus in these problems is finding the positive cases.

FN means predicted negative but actual is posetive

Example :- fire brigade alarming system.

→ our model should not predict 0 when actual fire is happened. and It's okay if our model alarms wrongly

74. Tell me one business case where precision is more important than recall

→

Precision is more important where False Alarms (False Positives) are more costly than Overlooked Cases (False Negatives). The focus in these problems is in weeding out the negative cases.

example: cancer detection problem → 1 - Not cancer
0 - cancer

75. Can we use accuracy for very much imbalanced data ? if yes/no, why ?

→ No we can't use accuracy measure for Imbalanced dataset. because It will get impacted by majority class.

Dataset (n) → +v 900
Dataset (n) → -v 100

Suppose we have random model which predict everything +v. so accuracy of model will be 90%. which is not correct.

76. difference between macro and micro average F_1-score

→ macro average F_1-score $= \frac{F_a + F_b + F_c}{3}$

where F_a = F_1-score for class a

F_b, F_c = F_1-score for b, c respectively.

micro average F_1 score $= \frac{2 \times \sum_{i}^{3} P_i * \sum_{i}^{3} Re}{\sum_{i}^{3} P_i + \sum_{i}^{3} Re}$

77. difference between AUC and accuracy.

→ The first big difference is that you **calculate accuracy on the predicted classes** while you **calculate ROC AUC on predicted scores**. That means you will have to find the optimal threshold for your problem.

Moreover, accuracy looks at fractions of correctly assigned positive and negative classes. That means if our **problem is highly imbalanced** we get a really **high accuracy score** by simply predicting that **all observations belong to the majority class.**

On the flip side, if your problem is **balanced** and you **care about both positive and negative predictions**, **accuracy is a good choice** because it is really simple and easy to interpret.

Another thing to remember is that **ROC AUC is especially good at ranking** predictions. Because of that, if you have a problem where sorting your observations is what you care about ROC AUC is likely what you are looking for.

77. How do we calculate AUC for a multiclass classification

→ generally AUC is used for binary classification problem. but we can use AUC for multiclass problem using OVR and OVO technique.

79. Test the complexity of kernel SVM?

→

1. Linear SVM has prediction complexity $O(d)$ with d the number of input dimensions since it is just a single inner product.
2. NN complexity is related to the architecture, but will surely be above that of linear SVM.
3. Prediction complexity of kernel SVM depends on the choice of kernel and is typically proportional to the number of support vectors. For most kernels, including polynomial and RBF, this is $O(n_{SV}d)$ where n_{SV} is the number of support vectors. An approximation exists for SVMs with an RBF kernel that reduces the complexity to $O(d^2)$. For computer vision applications, additive kernels are often used because they yield very fast prediction speed (independent of the number of SVs).

80. Can we use tSNE for dimensionality reduction?

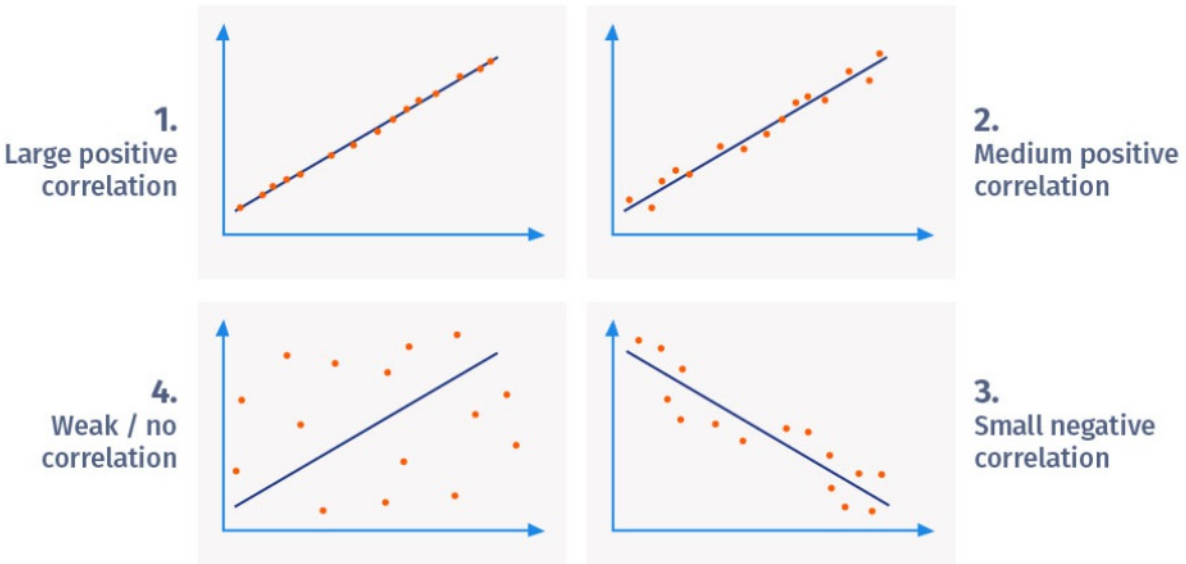
→ Yes, we can use tSNE for dimensionality reduction, but we must use dimension reduction using t-SNE for visualization.

81. what is pearson correlation coeffcient ?

→ problem with covariance : does not take affect what is variablity with X and Y.

PCC →

$$\rho_{xy} = \frac{Cov(x,y)}{\sigma_x * \sigma_y}$$



Pearson correlation coefficient or Pearson's correlation coefficient or Pearson's r is defined in statistics as the measurement of the strength of the relationship between two variables and their association with each other.

82. Training time complexity of Naive Bayes

→ Training:

time complexity: $O(ndc)$

space complexity: $O(dc)$

Runtime:

time complexity: $O(dc)$

83. No of tunable parameter in MaxPooling Layer?

→ No of tunable parameter in MaxPool1D

$= 0$

84. 100, 50) → embeddy Layer (36) → outputLayers ?

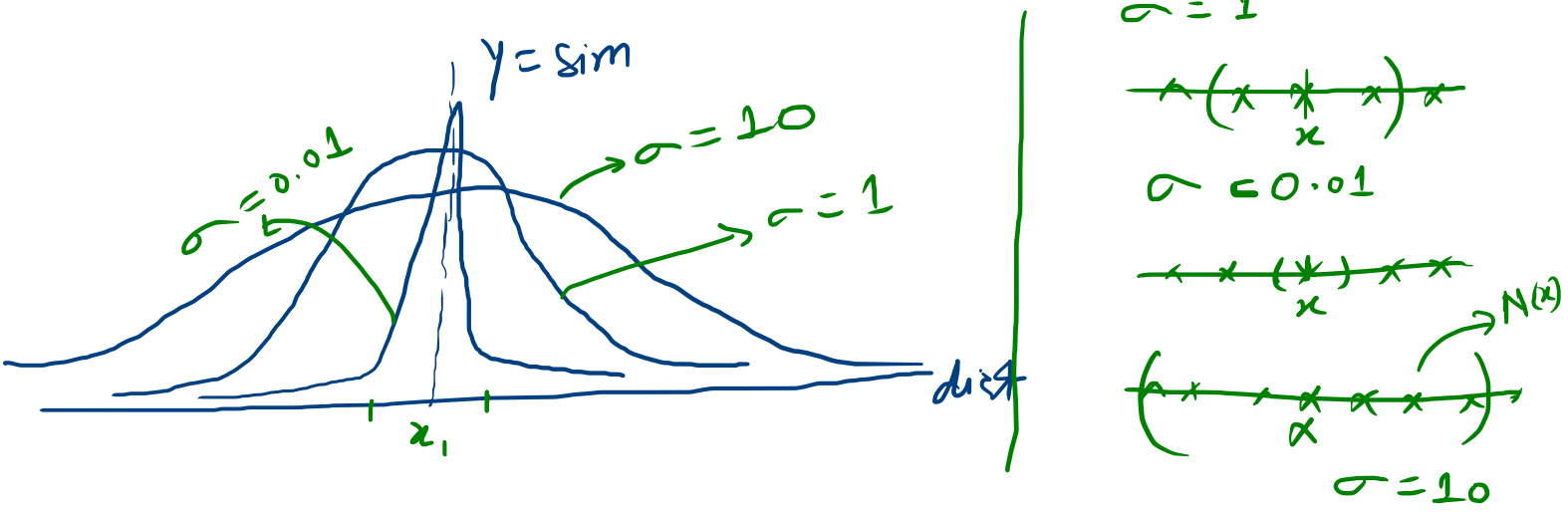
85. No of tunable parameters in embeedding Layers (36, vocal size = 75)

86. relationship between KNN and kernel SVM (RBF)?

→ RBF kernel is related to KNN because of sigma(σ)

$$RBF(x_1, x_2) = \exp\left(\frac{-\|x_1 - x_2\|^2}{2\sigma^2}\right)$$

and as σ increases, variance increases, so sigma increment is similar to increment of k in KNN.



87. which is faster

(a) SVC(C=1).fit(x,y)

(b) SGD(Loss='hinge).fit(x,y)

→ SGD(Loss='hinge').fit(x,y) will be faster

88. Explain about KS-Test.

→ we generally use ks-Test for matching two distribution are following same distribution or not

assume $x_1 : [x_1', x_2' \cdots x_n']$

$x_2 : [x_1, x_2 \cdots x_m]$

want to check that both have same distribution or not

for checking we have two follow below steps:

① plot CDF of x_1 and x_2 random variable

step ② define hypothesis

ⓐ H_0 : x_1 and x_2 have same distribution

ⓑ H_1 : Not from same distribution

Test statistics of KS Test :

$$D_{n,m} = \sup_x | CDF_{1,n}(x) - CDF_{2m}(x) |$$

where $CDF_{2,m}(x)$ → cdf of R.V x_2 with m sample

Sup → maximum value

step ③ Null hypothesis will be rejected at level α if

$$D_{n,m} > C(\alpha) \sqrt{\frac{n+m}{n \times m}}$$

and $c(\alpha)$ value can be get from table for common α. and even we can calculate

$$c(\alpha) = \sqrt{-\frac{1}{2} \ln\left(\frac{\alpha}{2}\right)}$$

where α is p-value or significant level.

Example: Suppose $n = 1000$ and $m = 5000$

want outer p-value $= 0.05$ so $c(\alpha) = 1.36$

then $$D_{n,m} > 1.36 \sqrt{\frac{1k + 5k}{1k * 5k}}$$

$$D_{n,m} > 0.047$$

so if $D_{n,m} > 0.047$ then we reject H_0 hypothesis at 5% of significant level and will accept alternative hypothesis.

89. what is KL - divergence ?

→ It is generally used to calculate distance between two distribution.

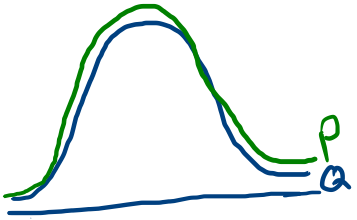
We also use KS-Test to calculate distance of two pdf by converting into CDf. but It is not differentiable So we use KL-divergence which is differentiable

$$dist(P,Q) = \sum_x P(x) \log\left(\frac{P(x)}{Q(x)}\right) \rightarrow \text{used for discrete R.V.}$$

or

$$\int_x P(x) \log\left(\frac{P(x)}{Q(x)}\right) \rightarrow \text{used for continuous R.V.}$$

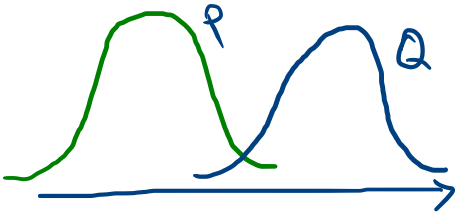
case 1:



$$\Rightarrow \frac{P(x)}{Q(x)} = 1 \rightarrow \log\left(\frac{P(x)}{Q(x)}\right) = 0 \text{ so}$$

$$D(P||Q) = 0$$

Case 2:



$P(x)$ = Some value

$Q(x) = 0.01$

then $D(P||Q) > 0$

And when $Q(x) = 0$ then $D(P||Q)$ = undefined

Note: ① KL value will be increase where there is dissimilarity in P and Q increasing.

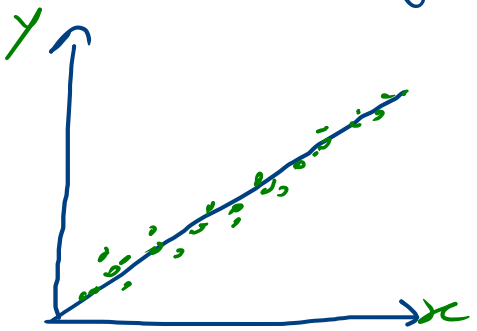
② KL statistic is differentiable and used for Similarity distribution

90. How QQ plot works?

→ It is generally used to verify that two distribution have same distribution or not.

the idea How it works is, we calculate 1 to 100 quantile for both Random varible and store $x_{1\text{to}100}$ and $y_{1\text{to}100}$ respectively.

After this we arrang as data points : $(x_1 y_1)$ $(x_2 y_2)$ ----- (x_{100}, y_{100}) and will plot the same



if all points lie on line then both R.V are following same distribution. some time this plot is used to check a RV follow $N(\mu, \sigma)$ distri. or not

91. what is the needed of confidence interval?

→ Sometime we provide interval with confidence probability.

Like population has height between 160 to 180 with 95% confidence.

This type of statement are called confidence Interval. which is very rich statement than point estimation.

92. How do you find out the outliers in the given dataset

→ we can detect outlier using domain knowledge, LOF and Logistic regression too.

93. Can you name a few sorting algorithm and their complexity?

→

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	Θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	Θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	Θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	Θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)

94. What is the time complexity of "a in list()"?

→ O(n)

95. What is the time complexity of "a in set()"?

→ O(1)

201. what does trainable = True/false mean in embedding Layer?

→ we set trainable = False to prevent weight update during training.

202. what happen when we set return sequence = True in LSTM?

→ return_sequence = True mean it will return Last output of LSTM. And Output can be in output sequence or the full sequence.

203. why are RNN's and CNN's called weight sharable layers?

→ Because in RNN, weight shared across the time stamp thus helps in understanding the sequence as well as

in applied point of view reduce the traing time.

→ in CNN

To reiterate parameter **sharing** occurs when a feature map is generated from the result of the convolution between a filter and input data from a unit within a plane in the conv layer. All units within this layer plane share the same **weights**; hence it is called **weight**/parameter **sharing**.

204. what happens during the fit and transform of following module?

① Standard scaler ② Count vectorizer

③ PCA

→ ① standard scaler:

"**fit**" computes the mean and std to be used for later **scaling**. (jsut a computation), nothing is given to you. "**transform**" uses a previously computed mean and std to autoscale the data (subtract mean from all values and then divide it by std). "fit_transform" does both **at** the same time.

(2) CountVectorizer:

Word Counts with **CountVectorizer**

Call the **fit**() function **in** order to learn a vocabulary from one or more documents. Call the **transform**() function on one or more documents as needed to encode each as a vector.

(3) PCA:

205. Can we use t-SNE for transforming test data? if not why?

→ Judging by the documentation of sklearn, **TSNE** simply does not have any **transform** method. Also, **TSNE** is an unsupervised method for dimesionality reduction/visualization, so it does not really work with a TRAIN and **TEST**. **You** simply take all of your **data** and **use** fit_transform to have the **transformation** and plot it. 06-Dec-2019

206. find the sum of diagonal in numpy array?

→ ① np.trace(Array) — return sum of diagonal element

② np.diagonal(Array) — return diagonal element of array or list. we can sum the list later.

207. write to code to get the count of row for each category in the dataframe?

→ df.groupby(key = "category")["category"].count()

208. different between categorical cross entrophy and binary cross entrophy.

→ With binary cross entropy, you can only classify two classes. With categorical cross entropy, you're not limited to how many classes your model can classify.

Binary cross entropy is just a special case of categorical cross entropy. The equation for binary cross entropy loss is the exact equation for categorical cross entropy loss with one output node.

For example, binary cross entropy with one output node is the equivalent of categorical cross entropy with two output nodes.

209. When you use w2v for test factorization, and we each sentence is having different words how can you forward data into models ?

210. what is tfidf w2v ?

→ It is a technique to transform text data into vector.

Suppose we want to convert a sentence into vector with tfidf w2v and that sentence has c words.

sent : w_1 w_2 w_3 ... w_c

So vector for sentence :

$$\text{vector} : \frac{tfidf(w_1) \times w_2v(w_1) + \ldots\ldots + tfidf(w_c) \times w_2v(c)}{\sum_{w_i \in sent}^{c} tfidf(w_i)}$$

211. How to use weighted distance in content based recommendation?

212. what is time complexity of SVD decomposition?

213. what is difference between content based and collaborative recommendation ?

→

214. why do you think inertia actually works in choosing elbow point in clustering?

215. what is gradient clipping?

→ Gradient clipping is a technique to prevent from exploding gradient in very deep Network

example: we can have L_2 Norm clipping

Suppose W:  → single vector for whole W

G:  Gradient vector

L_2 Norm Gradient $G_{new} = \frac{G}{\|G\|_2} \times c$ → threshold

So $G_{new} \leq 1 + e$

216. Which of these layers will be a better option as a last layer in multilabel classificatior

a. Sigmoid
b. Softmax

→ for multi Label classification "Sigmoid" activation and multi class classification – softmax activation

217. Is there a relation or similarity between LSTM and RESNET?

→ in LSTM, we have forget gate. through which it decide that previous information are important or not and based on it, we keep it or forget it. Same like RESNET, we can skip some portion of Layer if that Layer is not adding much information to model weight update.

218. what are value return by np.histogram()?

→ Returns:

hist: array

The density function returns the values of the histogram.

edge_bin: an array of float dtype

This function returns the bin edges **(length(hist+1))**.

219. what is PDF? can we calculate PDF for discrete destribution?

→ In probability theory, a probability density function (PDF), or density of a continuous random variable, is a function that describes the relative likelihood for this random variable to take on a given value.

Probability density function is defined by following formula:

$$P(a \leq X \leq b) = \int_a^b f(x)d_x$$

Note: for discrete R.V, we calculate PMF(Probability Mass function) which is same as PDF for Cont. R.V.

Where –

- $[a, b]$ = Interval in which x lies.
- $P(a \leq X \leq b)$ = probability that some value x lies within this interval.
- d_x = b-a

220. can the range of CDF be (0.5-1.5) ?

→ The **Cumulative Distribution Function (CDF)**, of a real-valued random variable X, evaluated at x, is the probability function that X will take a value less than or equal to x.

so probability term always lies between 0 - 1

221. Number of parameters in the following network :

a. Number of neurons = 4
b. Problem = binary classification
c. no: of FC = 2
d. Neurons in 1st FC = 5
e. Neurons in 2nd FC = 3

→ Input Layers : 4 (features)

No of parameter at first Layers : $5 \times (4+1) = 25$

No of parameter at 2nd Layers : $3 \times (5+1) = 18$

No of param at output layer : $1 \times (3+1) = 4$

Total = 47

222. How do we interpret alpha in dual form of SVM? what is relation between C and Alpha?

→ For now, let's just work with linear kernels. The primal representation w is related to the dual representation α in the following manner:

$$w = \sum_i \alpha_i x_i \quad (1)$$

You can interpret α as the contribution of the i-th training example to the final solution w.

There are three case though which we can understand How α is related to C.

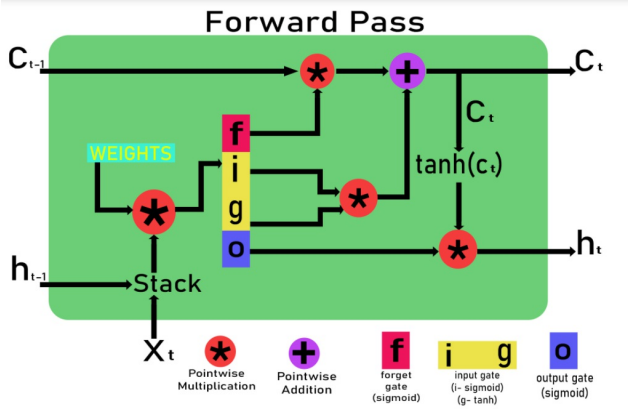
1. positive point has been miclassified by optimal w then α will be same as C.
2. positive point has been classified correctly above the margin then corresponding α will be zero.
3. positive point is exactly classified at margin then corresponding α will be lie b/w 0 and C.

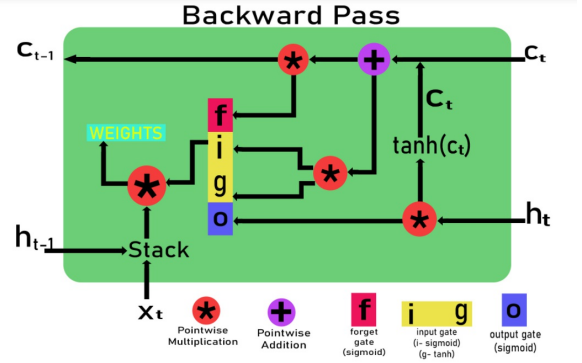
(8) What does the alpha mean in the dual form of the SVM optimization problem? - Quora

223. How does back propogation in case of LSTM?

→ In LSTM, back propogation is calculate through time.

As the name suggests backpropagation through time is similar to backpropagation in DNN(deep neural network) but due to the dependency of time in RNN and LSTM, we will have to apply the chain rule with time dependency.





C_t - cell state at time t

h_t - output at time t

224. Difference between supervised and unsupervised models ?

→

Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.
Supervised learning can be categorized in **Classification** and **Regression** problems.	Unsupervised Learning can be classified in **Clustering** and **Associations** problems.
Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.
Supervised learning model produces an accurate result.	Unsupervised learning model may give less accurate result as compared to supervised learning.

225. what is derivative of this fraction $1/(1+e^{-\sin x})$?

$$\rightarrow \frac{1}{(1+e^{-\sin x})} = (1+e^{-\sin x})^{-1} = M$$

by chain rule

$$\frac{dM}{dx} = \frac{dM}{da} \times \frac{da}{db} \times \frac{db}{dc} \times \frac{dc}{dx}$$

where we are assuming:

$$a = 1+e^{-\sin x} \qquad b = e^{-\sin x}$$

$$c = \sin x$$

$$\Rightarrow \frac{dM}{dx} = \frac{d(1+e^{-\sin x})^{-1}}{d(1+e^{-\sin x})} \times \frac{d(1+e^{-\sin x})}{d(e^{-\sin x})} \times \frac{d(e^{-\sin x})}{d(\sin x)} \times \frac{d(\sin x)}{dx}$$

$$\Rightarrow \frac{dM}{dx} = -1 \times (1+e^{-\sin x})^{-2} \times 1 \times -1 \times e^{-\sin x} \times \cos x$$

$$\Rightarrow \frac{1}{(1+e^{-\sin x})^2} \times e^{-\sin x} * \cos x$$

$$\Rightarrow \frac{e^{-\sin x} \times \cos x}{(1+e^{-\sin x})^2}$$

226: what will be output of a = [[1,2,3,10], [5,4,6,11], [8,7,9,12]]

then a[:, :-1]

→ a[:, :-1] = [[1,2,3]
[5,4,6]
[8,7,9]] } output

227. what is the output of this

```
a = [[1 5 9]
     [2 6 10]
     [3 7 11]
     [4 8 12]]

a[:-2,:]
```

→ output of a[:-2,:] = [[1, 2, 3, 4]]

228. what will be output of

```
a = dict()
a[('a','b')] = 0
a[(a,b)] = 1
print(a)
```

→ error will be occurred because b is not defined in third line.

229. what will be the output of

```
a = [[1,2,3],[4,5,6],[7,8,9]]
np.mean(a, axis=1)
```

→ output will be = [2, 5, 8]

Note: axis = 1 mean horizontal

axis = 0 mean vertical

230. what will be the output of np.stack((a,b), axis=0)

```
a = [3,4,5],[6,7,8],[9,10,11]
b = [1,2,3],[4,5,6],[7,8,9]
```

→ output: array = ([[3,4,5],(6,7,8),[9,10,11]],[[1 2 3]
[4 5 6]
[7 8 9]]))

231. what is "Local outlier factor" ?

→ Local outlier factor (LOF) is an algorithm used for Unsupervised outlier detection. It produces an anomaly score that represents data points which are outliers in the data set. It does this by measuring the local density deviation of a given data point with respect to the data points near it.

232. How RANSAC works ?

233. What is Jaccard & Cosine similarity ?

→ Jaccard similarity index

The **Jaccard Similarity Index** is a measure of the similarity between two sets of data.

It is calculated for A and B as:

$$\text{Jaccard similarity} = \frac{A \cap B}{A \cup B} \; ; \; 0 \leq J(A,B) \leq 1$$

Cosine Similarity:

Cosine similarity is a metric, helpful in determining, how similar the data objects are irrespective of their size. We can measure the similarity between two sentences in Python using Cosine Similarity. In cosine similarity, data objects in a dataset are treated as a vector. The formula to find the cosine similarity between two vectors is –

```
Cos(x, y) = x . y / ||x|| * ||y||
```

- **x . y** = product (dot) of the vectors 'x' and 'y'.
- **||x||** and **||y||** = length of the two vectors 'x' and 'y'.
- **||x|| * ||y||** = cross product of the two vectors 'x' and 'y'.

difference between jaccard similarity and cosine sim.

Jaccard similarity is used for two types of binary cases:

1. Symmetric, where 1 and 0 has equal importance (gender, marital status,etc)
2. Asymmetric, where 1 and 0 have different levels of importance (testing positive for a disease)

Cosine similarity is usually used in the context of text mining for comparing documents or emails. If the cosine similarity between two document term vectors is higher, then both the documents have more number of words in common

Another difference is 1 - Jaccard Coefficient can be used as a dissimilarity or distance measure, whereas the cosine similarity has no such constructs. A similar thing is the Tonimoto distance, which is used in taxonomy.

234. what are assumption of pearson correlation?

→ The assumptions for Pearson correlation coefficient are as follows: level of measurement, related pairs, absence of outliers, normality of variables, linearity, and homoscedasticity.

235. difference between pearson and spearman correlation?

→ Pearson's Correlation measures the *linear* correlation between two variables.

It would be most appropriate for finding the correlation between X and Y where

$$Y = aX + b + \epsilon,$$

where $\epsilon \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$,

and $a, b \in \mathbb{R}$

Spearman's Correlation measures linear as well as nonlinear *monotonic* relationships, such as

$$Y = aX^3 + bX + c + \epsilon,$$

or $Y = Ae^{mX} + \epsilon$

Spearman's Correlation is essentially a ranked version of Pearson's. If you have a nonlinear monotonic data set, the elements can be ranked such that Pearson's Correlation thinks it's linear, and the correlation will be, I believe, what Spearman's is.

236. what is train time complexity of DBSCAN?

→

237. Explain the proccdure of "predicting in hirarchical" clustering?

→ **Steps to Perform Hierarchical Clustering**

1. Step 1: First, we assign all the points to an individual **cluster**:
2. Step 2: Next, we will look at the smallest distance in the proximity matrix and merge the points with the smallest distance. ...
3. Step 3: We will repeat step 2 until only a single **cluster** is left.

Source: https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/

238. Relation between KNN and Kernel SVM ?

→ please go to question No: 86 for answer

239. proof of convergence of 'kmeans' ?

→ First, there are at most k^N ways to partition N data points into k clusters; each such partition can be called a "clustering". This is a large but finite number. For each iteration of the algorithm, we produce a new clustering based *only* on the old clustering. Notice that

1. if the old clustering is the same as the new, then the next clustering will again be the same.
2. If the new clustering is different from the old then the newer one has a lower cost

Since the algorithm iterates a function whose domain is a finite set, the iteration must eventually enter a cycle. The cycle can not have length greater than 1 because otherwise by (2) you would have some clustering which has a lower cost than itself which is impossible. Hence the cycle must have length exactly 1. Hence k-means converges in a finite number of iterations.

240. what is optimal value of minpoints for the data(1000,50)

Q. possible termination condition for clustering

→ These can be used for termination

1. This condition limits the runtime of the clustering algorithm, but in some cases the quality of the clustering will be poor because of an insufficient number of iterations.
2. Except for cases with a bad local minimum, this produces a good clustering, but runtimes may be unacceptably long.
3. This also ensures that the algorithm has converged at the minima.
4. Terminate when RSS falls below a threshold. This criterion ensures that the clustering is of a desired quality after termination. Practically, it's a good practice to combine it with a bound on the number of iterations to guarantee termination.