# Classical ML

Q1- What are the assumptions required for linear regression?

A1- There are four major assumptions:

1. There is a linear relationship between the dependent variables and the regressors, meaning the model you are creating actually fits the data,

2. The errors or residuals of the data are normally distributed and independent from each other,

3. There is minimal multicollinearity between explanatory variables, and

4. Homoscedasticity. This means the variance around the regression line is the same for all values of the predictor variable.

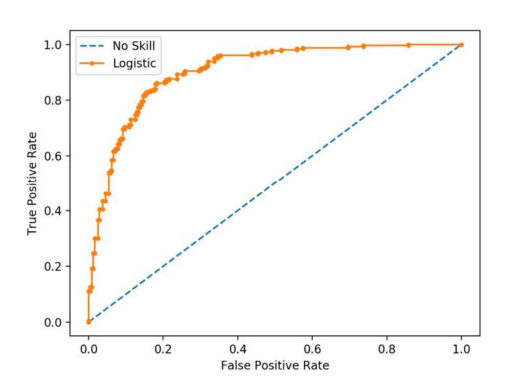
#### Q2- What is the interpretation of a ROC area under the curve?

A2- Receiver operating characteristics (ROC) shows the trade-off between sensitivity and specificity.

* Sensitivity: it is the probability that the model predicts a positive outcome when the actual value is also positive.
* Specificity: it is the probability that the model predicts a negative outcome when the actual value is also negative.

The curve is plotted using the False positive rate (FP/(TN + FP)) and true positive rate (TP/(TP + FN))

The area under the curve (AUC) shows the model performance. If the area under the ROC curve is 0.5, then our model is completely random. The model with AUC close to 1 is the better model.



Q2.1 Why is Area Under ROC Curve (AUROC) better than raw accuracy as an out-of-sample evaluation metric?

A2.1

ROC is robust to class imbalance, unlike raw accuracy. For example, if you want to detect a type of cancer that's prevalent in only 1% of the population, you can build a model that achieves 99% accuracy by simply classifying everyone has cancer-free.

Q3- What are Eigenvectors and Eigenvalues?

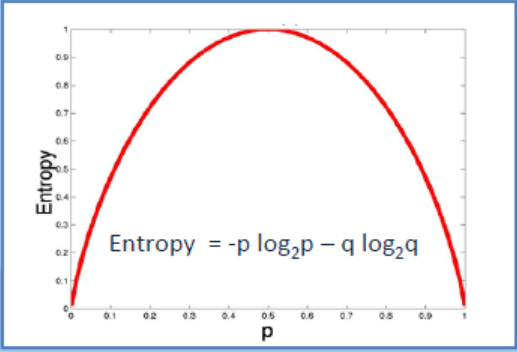
A3-Eigenvectors are used for understanding linear transformations. In data analysis, we usually calculate the eigenvectors for a correlation or covariance matrix. Eigenvectors are the directions along which a particular linear transformation acts by flipping, compressing or stretching. Eigenvalue can be referred to as the strength of the transformation in the direction of eigenvector or the factor by which the compression occurs.

Q4 - What are Entropy and Information gain in Decision tree algorithm?

A4- T

Entropy :

A decision tree is built top-down from a root node and involve partitioning of data into homogenious subsets. ID3 uses enteropy to check the homogeneity of a sample. If the sample is completely homogenious then entropy is zero and if the sample is an equally divided it has entropy of one.



. Q5- What is a Box-Cox Transformation?

A5:-The dependent variable for a regression analysis might not satisfy one or more assumptions of an ordinary least squares regression. The residuals could either curve as the prediction increases or follow the skewed distribution. In such scenarios, it is necessary to transform the response variable so that the data meets the required assumptions. **A Box cox transformation is a statistical technique to transform non-normal dependent variables into a normal shape**. If the given data is not normal then most of the statistical Applying a box cox transformation means that you can run a broader number of tests. A Box-Cox transformation is a way to transform non-normal dependent variables into a normal shape. Normality is an important assumption for many statistical techniques, if your data isn’t normal, applying a Box-Cox means that you are able to run a broader number of tests.

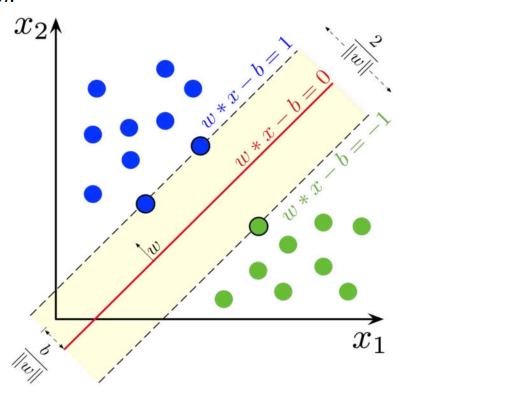
Q6 - What are the support vectors in SVM?

A6:-

In the diagram, we see that the sketched lines mark the distance from the classifier (the hyper plane) to the closest data points called the support vectors (darkened data points). The distance between the two thin lines is called the margin. To extend SVM to cases in which the data are not linearly separable, we introduce the hinge loss function,



This function is zero if x lies on the correct side of the margin. For data on the wrong side of the margin, the function's value is proportional to the distance from the margin.



Q7- What is a kernel? Explain the kernel trick

A7-

A kernel is a way of computing the dot product of two vectors  x and ᫣y in some (possibly very high dimensional) feature space, which is why kernel functions are sometimes called “generalized dot product” The kernel trick is a method of using a linear classifier to solve a non-linear problem by transforming linearly inseparable data to linearly separable ones in a higher dimension.

Q7.1 What are the different kernels in SVM?

A7.1-

There are four types of kernels in SVM. 1. LinearKernel 2. Polynomial kernel 3. Radial basis kernel 4. Sigmoid kernel

### Q8- What are Loss Function and Cost Functions? Explain the key Difference Between them?

A8 - When calculating loss we consider only a single data point, then we use the term loss function.

Whereas, when calculating the sum of error for multiple data then we use the cost function. There is no major difference.

In other words, the loss function is to capture the difference between the actual and predicted values for a single record whereas cost functions aggregate the difference for the entire training dataset.

The Most commonly used loss functions are Mean-squared error and Hinge loss.

Mean-Squared Error(MSE): In simple words, we can say how our model predicted values against the actual values.

MSE = √(predicted value - actual value)2

Hinge loss: It is used to train the machine learning classifier, which is

L(y) = max(0,1- yy)

Where y = -1 or 1 indicating two classes and y represents the output form of t

### Q9- 9. What is a Random Forest? How does it work?

A9-

Random forest is a versatile machine learning method capable of performing both regression and classification tasks.

Like bagging and boosting, random forest works by combining a set of other tree models. Random forest builds a tree from a random sample of the columns in the test data.

Here’s are the steps how a random forest creates the trees:

* Take a sample size from the training data.
* Begin with a single node.
* Run the following algorithm, from the start node:
  + If the number of observations is less than node size then stop.
  + Select random variables.
  + Find the variable that does the “best” job of splitting the observations.
  + Split the observations into two nodes.
  + Call step `a` on each of these nodes.

### Q10-How do check the Normality of a dataset?

Visually, we can use plots. A few of the normality checks are as follows:

* Shapiro-Wilk Test
* Anderson-Darling Test
* Martinez-Iglewicz Test
* Kolmogorov-Smirnov Test
* D’Agostino Skewness Test

Q11-What are the feature selection methods used to select the right variables?

A11-

There are two main methods for feature selection.

Filter Methods This involves: • Linear discrimination analysis • ANOVA • Chi-Square

Wrapper Methods This involves: • Forward Selection: We test one feature at a time and keep adding them until we get a good fit • Backward Selection: We test all the features and start removing them to see what works better • Recursive Feature Elimination: Recursively looks through all the different features and how they pair together Wrapper methods are very labor-intensive, and high-end computers are needed if a lot of data analysis is performed with the wrapper method

Q12- What is the difference between stochastic gradient descent (SGD) and gradient descent (GD)?

A12-

Both algorithms are methods for finding a set of parameters that minimize a loss function by evaluating parameters against data and then making adjustments. In standard gradient descent, you'll evaluate all training samples for each set of parameters. This is akin to taking big, slow steps toward the solution. In stochastic gradient descent, you'll evaluate only 1 training sample for the set of parameters before updating them. This is akin to taking small, quick steps toward the solution.

Q13-When is Ridge regression favorable over Lasso regression?

A13-

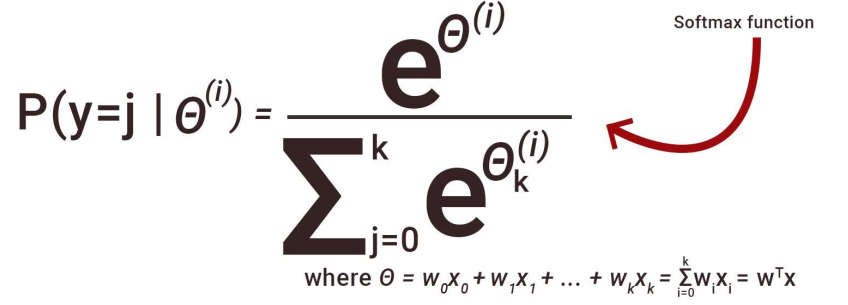
In the presence of few variables with medium / large sized effect, use lasso regression. In presence of many variables with small/medium-sized effects, use ridge regression. Conceptually, we can say, lasso regression (L1) does both variable selection and parameter shrinkage, whereas Ridge regression only does parameter shrinkage and end up including all the coefficients in the model. In the presence of correlated variables, ridge regression might be the preferred choice. Also, ridge regression works best in situations where the least square estimates have higher variance. Therefore, it depends on our model objective.

# DEEP-LEARNING

Q1- Why we generally use Soft-max (or sigmoid) non-linearity function as last operation in-network? Why RELU in an inner layer?

A1-:

It is because it takes in a vector of real numbers and returns a probability distribution. Its definition is as follows. Let x be a vector of real numbers (positive, negative, whatever, there are no constraints). Then the i-eth component of soft-max(x) is:



It should be clear that the output is a probability distribution: each element is non-negative and the sum over all components is 1. RELU because it avoids the vanishing gradient descent issue.

Q2- How Are Weights Initialized in a Network?

A2-

There are two methods here:

1. we can either initialize the weights to zero or assign them randomly. Initializing all weights to 0: This makes your model similar to a linear model. All the neurons and every layer perform the same operation, giving the same output and making the deep net useless.
2. Initializing all weights randomly: Here, the weights are assigned randomly by initializing them very close to 0. It gives better accuracy to the model since every neuron performs different computations. This is the most commonly used method.

Q3- What Is The Difference Between Epoch, Batch, and Iteration in Deep Learning?

A3-

Epoch – Represents one iteration over the entire dataset (everything put into the training model).

• Batch – Refers to when we cannot pass the entire dataset into the neural network at once, so we divide the dataset into several batches.

• Iteration – if we have 10,000 images as data and a batch size of 200. then an epoch should run 50 iterations (10,000 divided by 50).

Q4- What is the role of the Activation Function?

A4- The Activation function is used to introduce non-linearity into the neural network helping it to learn more complex function. Without which the neural network would be only able to learn linear function which is a linear combination of its input data. An activation function is a function in an artificial neuron that delivers an output based on inputs.

The net input value can be anything between -inf to +inf, and the neuron doesn't know how to bound the values, thus unable to decide the firing pattern. The activation function decides whether a neuron should be activated or not to bound the net input values.

Most common types of Activation Functions:

* Step Function
* Sigmoid Function
* ReLU
* Leaky ReLU

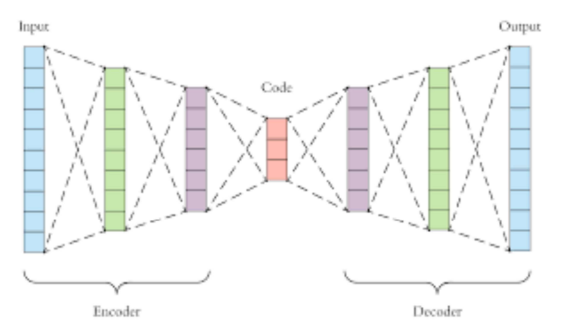
Q5- Difference Between Sigmoid and Softmax functions?

A5-

The sigmoid function is used for binary classification. The probabilities sum needs to be 1. Whereas, Softmax function is used for multi-classification. The probabilities sum will be 1.

Q6- What is an Auto-Encoder?

Auto-encoders are simple learning networks that aim to transform inputs into outputs with the minimum possible error. This means that we want the output to be as close to input as possible. ***We add a couple of layers between the input and the output, and the sizes of these layers are smaller than the input layer***. The auto-encoder receives unlabeled input which is then encoded to reconstruct the input. An autoencoder is a type of artificial neural network used to learn efficient data coding in an unsupervised manner. ***The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal “noise”. Along with the reduction side, a reconstructing side is learnt, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input, hence its name***. Several variants exist to the basic model, with the aim of forcing the learned representations of the input to assume useful properties. Autoencoders are effectively used for solving many applied problems, from face recognition to acquiring the semantic meaning of words.



Q7. What Is Dropout and Batch Normalization?

A7-

Dropout is a technique of dropping out hidden and visible nodes of a network randomly to prevent overfitting of data (typically dropping 20 per cent of the nodes). It doubles the number of iterations needed to converge the network. It used to avoid overfitting, as it increases the capacity of generalization.

Batch normalization is the technique to improve the performance and stability of neural networks by normalizing the inputs in every layer so that they have mean output activation of zero and standard deviation of one.

Q8 What is the exploding gradient problem while using the backpropagation technique?

A8-

When large error gradients accumulate and result in large changes in the neural network weights during training, it is called the exploding gradient problem. The values of weights can become so large as to overflow and result in NaN values. This makes the model unstable and the learning of the model to stall just like the vanishing gradient problem.

Q9- Difference between BatchNorm and LayerNorm?

A9-

BatchNorm — Compute the mean and var at each layer for every minibatch LayerNorm — Compute the mean and var for every single sample for each layer independently

Q-10- How would you reduce the inference time of a trained NN model?

A-10:

● Serve on GPU/TPU/FPGA

● 16 bit quantisation and served on GPU with fp16 support

● Pruning to reduce parameters

● Knowledge distillation (To a smaller transformer model or simple neural network)

● Hierarchical softmax/Adaptive softmax

● You can also cache results as explained here.

Q-11 How is backpropagation different in RNN compared to ANN?

A11-

In Recurrent Neural Networks, we have an additional loop at each node: This loop essentially includes a time component into the network as well. This helps in capturing sequential information from the data, which could not be possible in a generic artificial neural network. This is why the backpropagation in RNN is called Backpropagation through Time, as in backpropagation at each time step

Q12- How Does an LSTM Network Work?

A12-

Long-Short-Term Memory (LSTM) is a special kind of recurrent neural network capable of learning long-term dependencies, remembering information for long periods as its default behavior. There are three steps in an LSTM network:

Step 1: The network decides what to forget and what to remember.

Step 2: It selectively updates cell state values.

Step 3: The network decides what part of the current state makes it to the output.

Q13- Why is GRU faster as compared to LSTM?

A13-

As you can see, the LSTM model can become quite complex. In order to still retain the functionality of retaining information across time and yet not make a too complex model, we need GRUs. Basically, in GRUs, instead of having an additional Forget gate, we combine the input and Forget gates into a single Update Gate:

Q14-

# NLP- GENAI-

Q1-What TF-IDF helps you to establish?

A1-

TF-IDF helps to establish how important a particular word is in the context of the document corpus. TF-IDF takes into account the number of times the word appears in the document and offset by the number of documents that appear in the corpus.

● TF is the frequency of term divided by a total number of terms in the document.

● IDF is obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm of that quotient.

Q2- What is ngram in NLP?

A2-

N-gram in NLP is simply a sequence of n words, and we also conclude the sentences which appeared more frequently, for example, let us consider the progression of these three words: ● New York (2 gram) ● The Golden Compass (3 gram) ● She was there in the hotel (4 gram) Now from the above sequence, we can easily conclude that sentence (a) appeared more frequently than the other two sentences, and the last sentence(c) is not seen that often. Now if we assign probability in the occurrence of an n-gram, then it will be advantageous. It would help in making next-word predictions and in spelling error corrections.

Q3- How does the encoder-decoder structure work for language modelling?

A3-

The encoder-decoder structure is a deep learning model architecture responsible for several state of the art solutions, including Machine Translation. The input sequence is passed to the encoder where it is transformed to a fixed-dimensional vector representation using a neural network. The transformed input is then decoded using another neural network. Then, these outputs undergo another transformation and a softmax layer. The final output is a vector of probabilities over the vocabularies. Meaningful information is extracted based on these probabilities.

Q4- What are attention mechanisms and why do we use them?

A4-

Attention mechanisms are a function of the hidden weights at each time step. When we use attention in encoder-decoder networks, the fixed-dimensional vector passed to the decoder becomes a function of all vectors outputted in the intermediary steps. Two commonly used attention mechanisms are additive attention and multiplicative attention. As the names suggest, additive attention is a weighted sum while multiplicative attention is a weighted multiplier of the hidden weights. During the training process, the model also learns weights for the attention mechanisms to recognize the relative importance of each time step.

Q5- How is RAG implemented on Custom data

Q6- When Should will you think of finetuning the LLM model

Q7- What are different techniques you will employ the finetune LLM model

Q8- How would you implement an NLP system as a service, and what are some pitfalls you might face in production?

A-

This is less of a NLP question than a question for productionizing machine learning models. There are however certain intricacies to NLP models. Steve Nouri https://www.linkedin.com/in/stevenouri/ Without diving too much into the productionization aspect, an ideal Machine Learning service will have: ● endpoint(s) that other business systems can use to make inference

● a feedback mechanism for validating model predictions

● a database to store predictions and ground truths from the feedback

● a workflow orchestrator which will (upon some signal) re-train and load the new model for serving based on the records from the database + any prior training data

● some form of model version control to facilitate rollbacks in case of bad deployments

● post-production accuracy and error monitoring