Date: 15/09/2020

AI Assignment – 2 – 6064 – TYCS

1. **What’s the trade-off between bias and variance?**

Bias in a machine learning model occurs when the predicted values are further from the actual values. Low bias indicates a model where the prediction values are very close to the actual ones.

Underfitting: High bias can cause an algorithm to miss the relevant relations between features and target outputs.

      Variance refers to the amount the target model will change when trained with different training data. For a good model, the variance should be minimized.

      Overfitting: High variance can cause an algorithm to model the random noise in the training data rather than the intended outputs.

        The bias-variance decomposition essentially decomposes the learning error from any algorithm by adding the bias, variance, and a bit of irreducible error due to noise in the underlying dataset.

      Necessarily, if you make the model more complex and add more variables, you’ll lose bias but gain variance. To get the optimally-reduced amount of error, you’ll have to trade off bias and variance. Neither high bias nor high variance is desired.

High bias and low variance algorithms train models that are consistent, but inaccurate on average.

        High variance and low bias algorithms train models that are accurate but inconsistent**.**

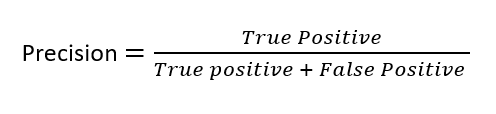
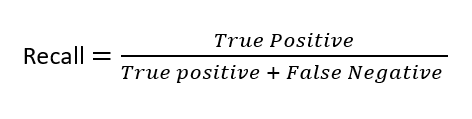
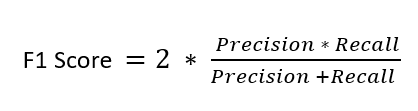
1. **How is KNN different from k-means clustering?**
2. KNN(K-Nearest Neighbours) is a classification (or regression) algorithm that determines the classification of a point, combining the classification of the K nearest points.
3. It is supervised because you are trying to classify a point based on the known classification of other points.
4. K-Means is a clustering algorithm that tries to partition a set of points into K sets (clusters) such that the points in each cluster tend to be near each other.
5. It is unsupervised because the points have no external classification.
6. In KNN, K refers to the number of closest neighbors to look for the output; whereas, in K-means clustering, K refers to the closest number of centroids that are used for predicting the output.

For Example:

KNN: If k = 7, then any new unlabeled data point would look at 7 closest points and classify the point from the majority of the above k classes it is closest to.

K-Means: in this, k refers to the number of centroid i.e. also called clusters or number of groups you want to cluster into.

**3. Define Precision and Recall**

1. Precision and recall are two numbers which together are used to evaluate the performance of classification or information retrieval systems. Precision is defined as the fraction of relevant instances among all retrieved instances.
2. Recall, sometimes referred to as ‘sensitivity, is the fraction of retrieved instances among all relevant instances. A perfect classifier has precision and recall both equal to 1.
3. It is often possible to calibrate the number of results returned by a model and improve precision at the expense of recall, or vice versa.
4. Precision and recall should always be reported together. Precision and recall are sometimes combined together into the F-score, if a single numerical measurement of a system's performance is required.
5. Precision and Recall Formulas : Mathematical definition of precision:
6. 
7. 
8. F1 score of F-measure is defined as the harmonic mean of Precision and Recall:
   1. 
9. For example, we are predicting if a patient has cancer or not p(x) be the probability that our model thinks the patient has cancer. If p(x) ≥ 0.5, then we are predicting the patient has cancer else if p(x) < 0.5 then we are predicting the patient doesn’t have cancer.

**5. Why is “Naive” Bayes naive?**

1. Naive bayes algorithm has two parts: 1st is Bayes and 2nd is naive. Bayes is for the simple reason that it uses the Bayes theorem to calculate the conditional probability. Let me first try to explain this part then we will go to the naive part. Suppose we need to classify some email as belonging to spam (+ve class | 1) or ham i.e not spam(-ve class | 0). Let's say our corpus has d unique words and we want to use binary BOW for the text to vector conversion.
2. Then we can assume that our feature space is d dimensional and each component can take 0 or 1 value. To classify our review ( vectorized as Xd which is a d dimensional vector)as spam or ham, we need to find the conditional probability P(y=1|Xd) and P(y=0|Xd) and compare them and classify it spam if former is more and ham if latter is more.
3. Now the difficult part is to find the conditional probability however we calculate this conditional probability using Bayes theorem, so this would be like:
   1. P(y=1 | Xd) = P(Xd | y=1) \* (P(Y=1)/(P(Xd)
4. Now the fun part comes : why is it naive? The answer lies in the assumption which this algorithm makes to solve the above equation.
5. The assumption is that all my features are conditionally independent given Y=1 (or O when we are calculating P(Y=0 | Xd). This means that the value of each component of d dimensional vector Xd is independent of each other given Y=1. This means that occurance of each word is conditional independent of the other words given some class label.
6. To give an example let's assume that we have an spam email then occurance of the word like “lottery” is independent of occurance of the word “money”, however in the real world this is not the case and these words will occur many times in the spam email. Hence this algorithm is called naive because it makes very naive or crude assumptions about the features.
7. By using the aforementioned assumption and chain rule of the conditional probability, we can simplify equation as:
   1. P(y=1 | Xd) = P(x1 | y = 1) \* P(x2 | y = 1) \* P(x3 | y = 1)\*……P(xd | y = 1) \* P(Y=1)/p(x)
8. Now we can calculate all the values of the RHS side using the training data and easily calculate the P(y=1| Xd) and P(y=0 | Xd).
9. One last thing - since denominator is common in both the class' conditional probability given Xd and we need to compare both of these, we can eliminate denominator and calculate the conditional probability to classify the email as spam or ham.
10. Even though this algorithm uses naive assumptions yet it is very effective for text classification tasks and it is used as a baseline algorithm for text classification tasks.
11. It's runtime complexity is O(kd) where k is number of classes and d is feature dimension. Which is small compared to other classifiers like K-NN.

**6. What’s the difference between probability and likelihood?**

1. The word likelihood refers to ‘possibility’. On the other hand, the word probability refers to ‘chance’. This is the main difference between the two words, namely, likelihood and probability.
2. In mathematics, probability is the chance that something can happen out of the total outcomes. This is calculated using the total number of chances and the number of chances the desired outcome can occur.
3. Likelihood is a function weaker than probability. So, likelihood is lower in number.
4. Probability is the measure of the likelihood that an event will occur that is, what is the certainty that a specific event will occur? Where-as a likelihood function is a function of parameters within the parameter space that describes the probability of obtaining the observed data.
5. So the fundamental difference is, Probability attaches to possible results; likelihood attaches to hypotheses.
6. The word probability has an adjective called probable and an adverb called probably.
7. The word likelihood has an adjective called like and an adverb called likely.
8. For example, let us consider a spinner that is divided into 8 equal parts. Four of them are numbered as number 3. Two are numbered as number 1, one as number 4 and one as number 5. The probability of the spinner landing on number 3 is higher. The likelihood of it landing on number 4 or 5 is very low.

**7. How is a decision tree pruned?**

* Tree pruning helps to address overfitting of a decision tree.
* Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations.
* Pruning refers to the process of reducing redundant branches of a decision tree. Decision Trees are prone to overfitting, pruning the tree helps to reduce the size and minimizes the chances of overfitting.
* Pruning involves turning branches of a decision tree into leaf nodes and removing the leaf nodes from the original branch. It serves as a tool to perform the tradeoff.

It can be done by 2 ways:

1. Pre-Pruning.
2. Post-pruning

Pre-Pruning :

* Is the halting of sub-tree construction at some node after checking some measures.
* At each stage of splitting the tree, we check the cross-validation error. If the error does not decrease significantly enough then we stop. Early stopping may underfit by stopping too early. The current split may be of little benefit, but having made it, subsequent splits more significantly reduce the error.
* These measures can be information gain, Gini index, etc.Early stopping- Pre-pruning may stop the growth process prematurely.

Post-Pruning :

* Grow the decision tree to its entirety.
* Trim the nodes of the decision tree in a bottom-up fashion. Post Pruning is done by replacing the node with leaf.
* If error improves after trimming, replace the sub tree by a leaf node.
* Post pruning decision trees is more mathematically rigorous, finding a tree at least as good as early stopping. Early stopping is a quick fix heuristic.
* The procedures are differentiated on the basis of their approach in the tree (top-down or bottom-up).
* Top-down fashion. It will traverse nodes and trim subtrees starting at the root
* Bottom-up fashion. It will begin at the leaf nodes
* There is a popular pruning algorithm called reduced error pruning, in which:
* Starting at the leaves, each node is replaced with its most popular class
* If the prediction accuracy is not affected, the change is kept.
* There is an advantage of simplicity and speed.
* Criteria:

If the error of the parent is lesser than its child then prune the tree else not. i.e

**if Parent (error)< Child(error) then "Prune"**

**else don't Prune.**

**8. When should you use classification over regression?**

* 1. Regression: If your outcome of interest is a real or continuous value. For example, you would like to predict the amount of money a person will spend when they visit your website.
  2. Classification: If your outcome of interest is discrete or categorical, and you would like to predict class membership. For example, you would like to predict whether or not a person will elect to purchase something when they visit your website. Classification methods extend to more than two classes as well.
  3. In short, Classification is the task of predicting a discrete class label. Whereas Regression is the task of predicting a continuous quantity.
  4. Some algorithms can be used for both classification and regression with small modifications, such as decision trees and artificial neural networks. Some algorithms cannot, such as linear regression for regression predictive modeling and logistic regression for classification predictive modeling.
  5. The way we evaluate classification and regression predictions varies and does not overlap, for example:
     1. Classification predictions can be evaluated using accuracy, whereas regression predictions cannot.
     2. Regression predictions can be evaluated using root mean squared error, whereas classification predictions cannot.
  6. Examples of classification problems include:
     1. Predicting yes or no
     2. Estimating gender
     3. Breed of an animal
     4. Type of color
  7. Examples of regression problems include:
     1. Estimating sales and price of a product
     2. Predicting the score of a team
     3. Predicting the amount of rainfall

**9. Name an example where ensemble techniques might be useful.**

·        Deep learning neural networks are **nonlinear methods**.

·        They offer **increased flexibility** and can scale in proportion to the amount of training data available.

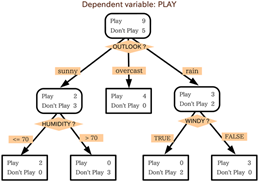
·        A downside of this flexibility is that they learn via a **stochastic training algorithm** which means that they are sensitive to the specifics of the training data and may find a **different set of weights** each time they are trained, which in turn produce different predictions.

·        Generally, this is referred to as neural networks having a **high variance** and it can affect the outcome when trying to develop a final model to use for making predictions.

·        A successful approach to reducing the variance of neural network models is to **train multiple models** instead of a single model and to **combine** the predictions from these models. This is called **ensemble learning** and not only **reduces the variance** of predictions but also can result in **predictions that are better** than any single model.

·        Ensemble methods is a machine learning technique that combines several base models to produce one optimal predictive model.

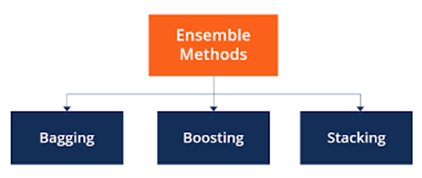
·        For example, consider the following decision tree.



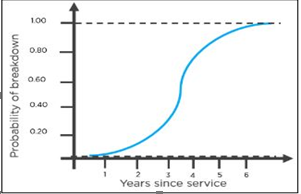
·        When making Decision Trees, there are several factors we must take into consideration: On what features do we make our decisions on? What is the threshold for classifying each question into a yes or no answer? In the above Decision Tree, what if we wanted to ask ourselves if we had friends to play with or not. If we have friends, we will play every time. If not, we might continue to ask ourselves questions about the weather. By adding a question, we hope to better define the Yes and No classes.

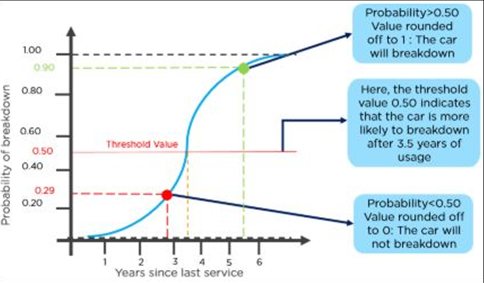
·        This is where Ensemble Methods are useful. Rather than just relying on one Decision Tree and hoping we made the right decision at each split, Ensemble Methods allow us to take a sample of Decision Trees into account, calculate which features to use or questions to ask at each split and make a final predictor based on the aggregated results of the sampled Decision Trees.

·        Ensemble methods are of three types:



**11 .How would you evaluate a logistic regression model?**

1. Logistic regression is a classification algorithm used to predict a binary outcome for a given set of independent variables.
2. The output of logistic regression is either a 0 or 1 with a threshold value of generally 0.5. Any value above 0.5 is considered as 1, and any point below 0.5 is considered as 0.
3. The following is an example of a logistic function we can use to find the probability of a vehicle breaking down, depending on how many years it has been since it was serviced last.
4. 
5. Here is how you can interpret the results from the graph to decide whether the vehicle will break down or not.



**12. What is ‘training Set’ and ‘test Set’ in a Machine Learning Model? How Much Data Will You Allocate for Your Training, Validation, and Test Sets?**

1) Training sets and test sets are used in Supervised Learning algorithms as a source of data.

2) A training dataset of labeled examples is used during the learning process and is used to estimate the parameters.

3) A training set is used to feed the neural network for training.

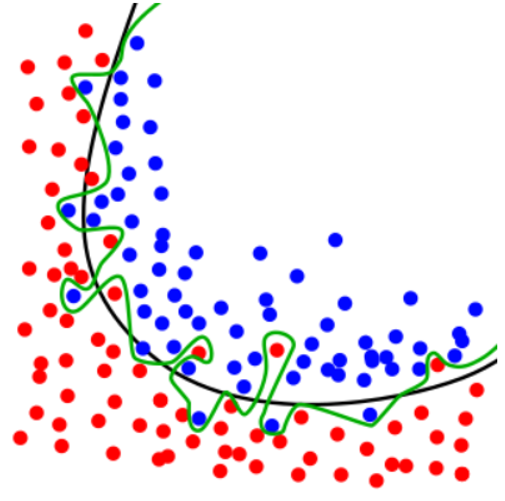
4) A test dataset is a dataset that is independent of the training dataset, but that follows the same probability distribution as the training dataset.

5) A better fitting of the training dataset as opposed to the test dataset usually leads to overfitting.

6)  The only reason why the test dataset is independent of the training dataset is for preventing the model from overfitting.

7) It is common to **allocate** 50 percent or more of the **data** to the **training set**, 25 percent to the **test set**, and the remainder to the **validation set**.

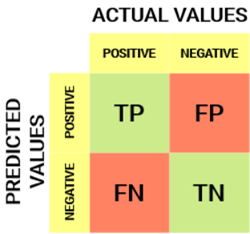
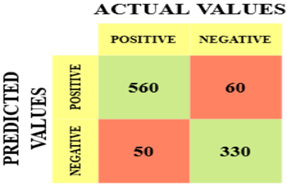
Example of Overfitting Data by using the same data for training and testing:



In the above-given example,the same dataset is used for training and testing which leads the green curve to overfit the data.

The black curve is the optimal curve expected.

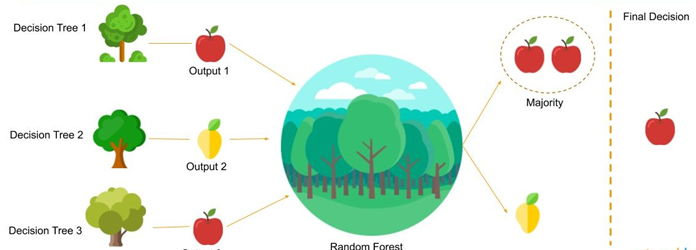
**13. Explain the confusion matrix with Respect to Machine Learning Algorithms**.

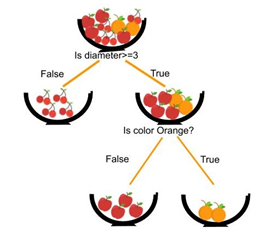
1. A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.
2. We would have a 2 x 2 matrix as shown below with 4 values:
3. 
4. The target variable has two values: Positive or Negative
5. The columns represent the actual values of the target variable
6. The rows represent the predicted values of the target variable
7. Understanding True Positive, True Negative, False Positive and False Negative in a Confusion Matrix
8. True Positive (TP): The predicted value matches the actual value. The actual value was positive and the model predicted a positive value.
9. True Negative (TN): The predicted value matches the actual value. The actual value was negative and the model predicted a negative value.
10. False Positive (FP) – Type 1 error: The predicted value was falsely predicted. The actual value was negative but the model predicted a positive value. Also known as the Type 1 error.
11. False Negative (FN) – Type 2 error: The predicted value was falsely predicted. The actual value was positive but the model predicted a negative value. Also known as the Type 2 error
12. Example: Suppose we had a classification dataset with 1000 data points.
13. 
14. The different values of the Confusion matrix would be as follows:
15. True Positive (TP) = 560; meaning 560 positive class data points were correctly classified by the model
16. True Negative (TN) = 330; meaning 330 negative class data points were correctly classified by the model
17. False Positive (FP) = 60; meaning 60 negative class data points were incorrectly classified as belonging to the positive class by the model.
18. False Negative (FN) = 50; meaning 50 positive class data points were incorrectly classified as belonging to the negative class by the model.
19. This turned out to be a pretty decent classifier for our dataset considering the relatively larger number of true positive and true negative values.

**14. What Is a False Positive and False Negative and How Are They Significant?**

1. A test condition that returns a result as a present when it does not is known as a False Positive.
2. A test condition that returns a result as false even when the case is true is known as False Negative.
3. False positives and false negatives are of significance when dealing with hypothesis testing.
4. Type I error is the rejection of a true null hypothesis, which is the same as a False Positive.
5. Type II error is the non-rejection / acceptance of a false null hypothesis, which is the same as a False Negative.
6. Most Machine Learning stems from models being held across some distribution with some error rate, thereby the algorithms being statistical.
7. The significance is then further amplified by the above application of hypothesis testing.
8. For example, a pregnancy test which indicates a woman is pregnant when she is not is an example of a False positive, whereas when a pregnancy test indicates a woman is not pregnant, but she is, is an example of a False Negative.

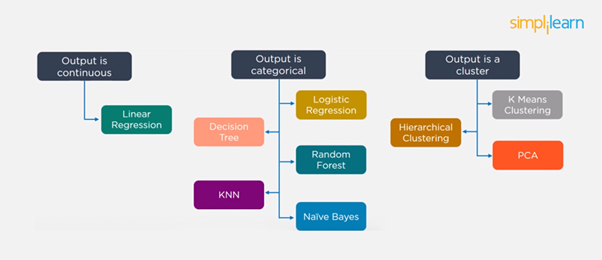
**15. What is a Random Forest?**

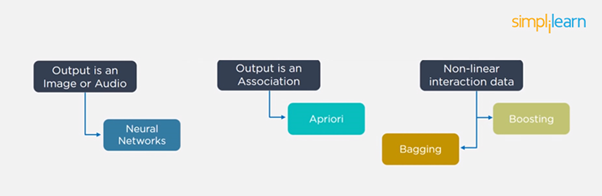
1. Random Forest is a learning method that operates by constructing multiple decision trees. The final decision is made based on the majority of the trees and is chosen by the random forest.
2. 
3. A decision tree is a tree-shaped diagram used to determine a course of action. Each branch of the tree represents a possible decision, occurrence, or reaction.



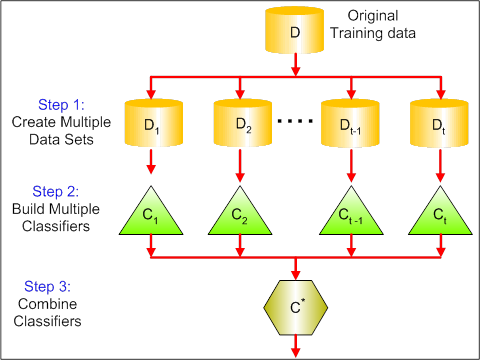
**16. Considering a Long List of Machine Learning Algorithms, given a Data Set, How Do You Decide Which One to Use?**

1. There is no master algorithm for all situations. Choosing an algorithm depends on the following questions:
2. How much data do you have, and is it continuous or categorical?
3. Is the problem related to classification, association, clustering, or regression?
4. Predefined variables (labeled), unlabeled, or mix?
5. What is the goal?
6. Based on the above questions, the following algorithms can be used:





**17. Explain bagging.**

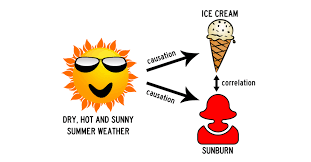
1. Bagging is an ensemble technique mainly used to reduce the variance of our predictions by combining the result of *multiple classifiers* modelled on different *subsamples* of the same data set.
2. 
3. Intuitively, let’s say, you want to carry 1000 kgs of potatoes from Delhi to Mumbai. Obviously, you can’t stock them in a single place. You divide them in bags of 100 kgs, transport them, and then aggregate after reaching. Same with data.
   1. Main Steps:
4. Creating multiple datasets: Sampling is done withreplacement on the original data and new datasets are formed.
5. Building multiple classifiers: On each of these smaller datasets, a classifier is built, usually the same classifier on all the datasets.
6. Combining Classifiers: The predictions of all the individual classifiers are now combined to give a better classifier, usually with very less variance compared to before.
7. In words :
8. For datasets with high variance, we could use the bagging algorithm to handle it. Bagging algorithm splits the data into subgroups with sampling replicated from random data. After the data is split, random data is used to create rules using a training algorithm. Then we use a polling technique to combine all the predicted outcomes of the model.
9. Bagging is utilised where multiple decision trees are made which are trained on samples of the original data and the final result is the average of all these individual models.
10. Bagging is the technique used by Random Forests. Random forests are a collection of trees which work on sampled data from the original dataset with the final prediction being a voted average of all trees.

**18. How are covariance and correlation different from one another?**

1. “Covariance” indicates the direction of the linear relationship between variables. “Correlation” on the other hand measures both the strength and direction of the linear relationship between two variables.
2. With some pseudo-mathematical notation, **f(***correlation***)** = *covariance*.
3. The formula of Covariance is below.
   1. 
   2. Or in terms of Expected value, it is written as
   3. Cov(x, y)=E[(x- x) (y - y)]
4. Where x, y are population means of x and y.
5. For Correlation, you can employ the same with minor change since it is the function of covariance. The formula is below,
   1. Corr(x, y)=Cov(x,y)xy
6. Where x, y are the standard deviations of x, y respectively.

**19. State the Difference between Causality and Correlation.**

1. Correlation doesn't imply causation. Causation explicitly applies to cases where action A Causation explicitly applies to cases where action A causes outcome B. On the other hand, correlation is simply a relationship. Action A relates to Action B—but one event doesn’t necessarily cause the other event to happen.
2. Correlation and causation are often confused because the human mind likes to find patterns even when they do not exist. We often fabricate these patterns when two variables appear to be so closely associated that one is dependent on the other. That would imply a cause and effect relationship where the dependent event is the result of an independent event.
3. However, we cannot simply assume causation even if we see two events We cannot simply assume causation even if we see two events happening, seemingly together, before our eyes.happening, seemingly together, before our eyes. One, our observations are purely anecdotal. Two, there are so many other possibilities for an association, including. The opposite is true: B actually causes A.
4. The two are correlated, but there’s more to it: A and B are correlated, but they’re actually caused by C.
5. There’s another variable involved: A does cause B—as long as D happens.
6. There is a chain reaction: A causes E, which leads E to cause B



**20. What is the difference between stochastic gradient descent (SGD) and gradient descent (GD)?**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **BATCH GRADIENT DESCENT** | **STOCHASTIC GRADIENT DESCENT** |
| 1. | Computes gradient using the whole Training sample | Computes gradient using a single Training sample |
| 2. | Slow and computationally expensive algorithm | Faster and less computationally expensive than Batch GD |
| 3. | Not suggested for huge training samples. | Can be used for large training samples. |
| 4. | Deterministic in nature. | Stochastic in nature. |
| 5. | Gives optimal solution given sufficient time to converge. | Gives a good solution but not optimal. |
| 6. | No random shuffling of points is required. | The data sample should be in a random order, and this is why we want to shuffle the training set for every epoch. |
| 7. | Can’t escape shallow local minima easily. | SGD can escape shallow local minima more easily. |
| 8. | Convergence is slow. | Reaches convergence much faster. |

**21. When does regularization come into play in Machine Learning?**

1. Regularisation is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting.
2. The commonly used regularisation techniques are :
3. L1 regularisation
4. L2 regularisation
5. Dropout regularisation
6. A regression model which uses L1 Regularisation technique is called LASSO(Least Absolute Shrinkage and Selection Operator) regression.
7. A regression model that uses L2 regularisation technique is called Ridge regression.
8. Lasso Regression adds “absolute value of magnitude” of coefficient as penalty term to the loss function(L).
9. Ridge regression adds “squared magnitude” of coefficient as penalty term to the loss function(L).
10. We define Loss function in Logistic Regression as :
    * 1. L(y\_hat,y) = y log y\_hat + (1 - y)log(1 - y\_hat)
    * Loss function with no regularisation :
      1. L = y log (wx + b) + (1 - y)log(1 - (wx + b))
    * Let's say the data overfits the above function.
11. Loss function with L1 regularisation :
12. L = y log (wx + b) + (1 - y)log(1 - (wx + b)) + lambda\*||w||1
13. Loss function with L2 regularisation :
14. L = y log (wx + b) + (1 - y)log(1 - (wx + b)) + lambda\*||w||22
    * lambda is a Hyperparameter Known as regularisation constant and it is greater than zero.