**1. Give short description each of Linear, RBF, Polynomial kernels used in SVM.**

The basic difference between three kernels of SVM is how they make decisions in choosing the hyper planes. Kernels basically project data into higher dimensional space so that it can be made linearly separable.

RBF stands for radial bias function. On two samples x and x’ represented as feature vectors in some space, it can be defined as

K(x,x’) = exp(- || x-x’ ||2 / 2σ2)

On the other hand, polynomial kernel represents the similarity of two vectors in a feature space over the polynomials of original variables thus allowing the learning of non-linear models. It not only does look on the given features to calculate similarity, but also look at the combination of these. It is defined as

K(x,y) = (xTy + c)d

for a degree-d polynomial.

Linear kernel is the fastest and is used when the data is already linearly seperable.

**2. R-squared or Residual Sum of Squares (RSS) which one of these two is a better measure of goodness of fit of model in regression and why?**

R-squared is a better measure of goodness of fit for a model as compared to residual sum of squares. R-squared tell us about the total variance in our target variable that could be explained by the independent variables while RSS is just sum of squares of residuals.

The value of R-squared lies between 0 and 1 and therefore it is easily interpretable, while the interpretability of RSS is not so good.

We can not comment anything on the goodness of fit of a model by looking at RSS score of 2000 because it is scale-dependent and is also affected by outliers but we can surely comment on goodness by R-squared score of 0.8.

**3. What are TSS (Total Sum of Squares), ESS (Explained Sum of Squares) and RSS (Residual Sum of Squares) in regression. Also mention the equation relating these three metrics with each other.**

The explained sum of squares (ESS) is the sum of the squares of the deviations of the predicted values from the mean value of target variable.

The total sum of squares is the sum of squares of deviation of the actual values from the mean value of target variable.

The total sum of squares is the sum of squares of deviation of the actual values from the predicted values of target variable.

Y = actual values

Y’ = predicted values and

μ = mean value of target variable

ESS = Σ(Y’ – μ)2

TSS = Σ(Y – μ)2

RSS = Σ(Y’ – Y)2

**4. What is Gini –impurity index?**

The Gini impurity measure is one of the methods used in decision tree algorithms to decide the optimal split from a root node, and subsequent splits. Gini Impurity tells us what is the probability of misclassifying an observation. Note that the lower the Gini, the better the split.

**5. Are un-regularized decision-trees prone to over-fitting? If yes, why?**

Yes, decision trees are more prone to over-fitting if they are not regularized using the hyper-parameters. The reason is, un-regularized tress grow to a depth until they don’t get pure leaf nodes. Sometimes, it may happen that there is only one point in each leaf nodes. Now, changing a single point in dataset will cause a change in the entire model, which is nothing but over-fitting.

**6. What is an ensemble technique in machine learning?**

Ensemble technique refers to the process of combining various weak learners to result into a strong learner. Bagging, boosting and stacking are examples of ensmebling techniques used widely.

**7. What is the difference between Bagging and Boosting techniques?**

**Bagging :** Bagging is also known as bootstrap aggregating sits on top of the majority voting principle. The samples are bootstrapped each time when the model is trained. When the samples are chosen, they are used to train and validate the predictions. The samples are then replaced back into the training set. The samples are selected at random. This technique is known as bagging. To sum up, base classifiers such as decision trees are fitted on random subsets of the original training set. Subsequently, the individual predictions are aggregated (voting or averaging etc.). The final results are then used as predictions. It reduces the variance of a black box estimator. Due to this the chances of overfitting is ruled out.

**Boosting:** The concept of Adaptive Boost revolves around correcting previous classifier mistakes. Each classifier gets trained on the sample set and learns to predict. The misclassification errors are then fed into the next classifier in the chain and are used to correct the mistakes until the final model predicts accurate results. When a weak-classifier misclassifies a training sample, the algorithm then uses these very samples to improve the performance of the ensemble.

**8. what is out-of-bag error in random forests?**

**Out-of-bag error**, also called **out-of-bag estimate**, is a method of measuring the prediction error of random forest, boosted decision trees, and other models utilizing bagging to sub-sample data samples used for training. OOB is the mean prediction error on each training sample xi, using only the trees that did not have xiin their bootstrap sample.

Sub-sampling allows us to define an out-of-bag estimate of the prediction performance improvement by evaluating predictions on those observations which were not used in the building of the next base learner.

**9. What is K-fold cross-validation?**

K-fold Cross validation is a technique to fit a model on data set. In cross validation the data set is divided into ‘k’ number of sets where ‘k-1’ sets are used for training and 1 set is used as validation set. And this is done for all the set one by one and the final score of model is taken as average score of all the ‘k’ number of fits.

Advantage of using Cross validation is that, there is no need of separate validation data, cross validation reduces chances of over-fitting and gives a more generic model. Cross validation has a disadvantage that it takes more time to fit the model over a large dataset and the model built is more complex than the basic model.

**10. What is hyper parameter tuning in machine learning and why it is done?**

Hyper-parameter tuning basically refers to the process of tuning the parameters of a given algorithm (for example- max\_depth in decision trees) such as to regularize our model and prevent it from over-fitting and under-fitting.

**11. What issues can occur if we have a large learning rate in Gradient Descent?**

If the learning rate in gradient descent is very large, instead of converging to the minima, loss might diverge and over shoot. In such cases, it will never attain minima.

**12. What is bias-variance trade off in machine learning?**

In machine learning, the error term comprises of three terms

Error = Bias2 + Variance + Irreducible Err

Now, in order to minimize the overall we’ll have to minimize bias as well variance. But, it is a known fact that as bias reduces, variance increases and vice-versa. Therefore, we’ll have to set a trade-off between bias and variance such that overall error is the lowest. This trade-off between bias and variance is known as bias-variance trade-off.

**13. What is the need of regularization in machine learning?**

We need regularization in order to prevent our model from under-fitting or over-fitting.

**14. Differentiate between Ada-boost and Gradient Boosting**

Both methods use a set of weak learners. They try to boost these weak learners into a strong learner.

Gradient boosting generates learners during the learning process. It build first learner to predict **the values/labels of samples**, and calculate the loss (the difference between the outcome of the first learner and the real value). It will build a second learner to predict the **loss after the first step.** The step continues to learn until certain threshold.

Ada-Boost is designed in such a way that at every step the sample distribution is adapted to put more weight on misclassified samples and less weight on correctly classified samples. The final prediction is a weighted average of all the weak learners, where more weight is placed on stronger learners.

**15. Can we use Logistic Regression for classification of Non-Linear Data? If not, why?**

No, we should not use logistic regression for classification of non-linear data because logistic regression performs poorly on non-linear data. Though, we can explicitly apply some feature transformations (such as polynomial features) to make the data linearly separable and then use logistic regression on top of it.

The reason behind this is that the decision boundary of logistic regression is linear in X.