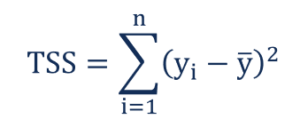
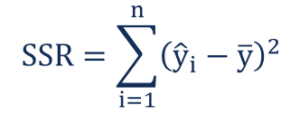
1--

SVM Kernels SVM algorithm is implemented with kernel that transforms an input data space into the required form. SVM uses a technique called the kernel trick in which kernel takes a low dimensional input space and transforms it into a higher dimensional space. In simple words, kernel converts non-separable problems into separable problems by adding more dimensions to it. It makes SVM more powerful, flexible and accurate.

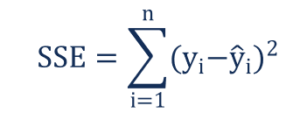
Linear Kernel : It can be used as a dot product between any two observations. The formula of linear kernel is as -- K(x,xi)=sum(x∗xi) From the formula, we can see that the product between two vectors say 𝑥 & 𝑥𝑖 is the sum of the multiplication of each pair of input values. Polynomial Kernel : It is more generalized form of linear kernel and distinguish curved or nonlinear input space.The formula for polynomial kernel k(X,Xi)=1+sum(X∗Xi)^d ,Here d is the degree of polynomial, which we need to specify manually in the learning algorithm. Radial Basis Function (RBF) Kernel : RBF kernel, mostly used in SVM classification, maps input space in indefinite dimensional space.The formula explains it mathematically −K(x,xi)=exp(−gamma∗sum(x−xi^2)) Here, gamma ranges from 0 to 1. We need to manually specify it in the learning algorithm. A good default value of gamma is 0.1.

3—(a) Total sum of squares: The total sum of squares is a variation of the values of a dependent variable from the sample mean of the dependent variable. Essentially, the total sum of squares quantifies the total variation in a sample. It can be determined using the following formula Where yi – the value in a sample and ȳ – the mean value of a sample.

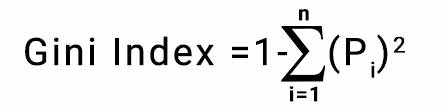
(b) Regression sum of squares (also known as the sum of squares due to regression or explained sum of squares) :-The regression sum of squares describes how well a regression model represents the model data. A higher regression sum of squares indicates that the model does not fit the data well. The formula for calculating the regression sum of squares is:

Where ŷi – the value estimated by the regression line and ȳ – the mean value of a sample

(c) Residual sum of squares (also known as the sum of squared errors of prediction):

The residual sum of squares essentially measures the variation of modelling errors. In other words, it depicts how the variation in the dependent variable in a regression model cannot be explained by the model. Generally, a lower residual sum of squares indicates that the regression model can better explain the data while a higher residual sum of squares indicates that the model poorly explains the data. The residual sum of squares can be found using the formula below Where yi – the observed value and ŷi – the value estimated by the regression line.

-The relationship between the three types of sum of squares can be summarized by the following equation: Relationship Formula.

4-- Gini Index or Gini impurity:- Gini Index also known as Gini impurity, calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly. If all the elements are linked with a single class then it can be called pure. The Gini Index is determined by deducting the sum of squared of probabilities of each class from one, mathematically, Gini Index can be expressed asWhere Pi denotes the probability of an element being classified for a distinct class.

5—Yes unregularized decision-trees prone to overfitting .Over-fitting is the phenomenon in which the learning system tightly fits the given training data so much that it would be inaccurate in predicting the outcomes of the untrained data. In decision trees, over-fitting occurs when the tree is designed so as to perfectly fit all samples in the training data set. Thus it ends up with branches with strict rules of sparse data. Thus this effects the accuracy when predicting samples that are not part of the training set. One of the methods used to address over-fitting in decision tree is called pruning which is done after the initial training is complete. In pruning, you trim off the branches of the tree, i.e., remove the decision nodes starting from the leaf node such that the overall accuracy is not disturbed. This is done by segregating the actual training set into two sets: training data set and validation data set. Prepare the decision tree using the segregated training data set. Then continue trimming the tree accordingly to optimize the accuracy of the validation data set.

6—

An ensemble technique is basically combining a diverse set of learners together to improvise on the stability and predictive power of the model. Some Commonly used Ensemble learning techniques are Bagging, Boosting and Stacking.

7-- **Bagging** is used when the goal is to reduce the variance of a decision tree classifier. Here the objective is to create several subsets of data from training sample chosen randomly with replacement. Each collection of subset data is used to train their decision trees. As a result, we get an ensemble of different models. Average of all the predictions from different trees are used which is more robust than a single decision tree classifier.

Bagging Steps:

-Suppose there are N observations and M features in training data set. A sample from training data set is taken randomly with replacement.

-A subset of M features are selected randomly and whichever feature gives the best split is used to split the node iteratively. The tree is grown to the largest.

-Above steps are repeated n times and prediction is given based on the aggregation of predictions from n number of trees.

Advantages:

Reduces over-fitting of the model.

Handles higher dimensionality data very well.

-Maintains accuracy for missing data.

Disadvantages:

-Since final prediction is based on the mean predictions from subset trees, it won’t give precise values for the classification and regression model.

**Boosting** is used to create a collection of predictors. In this technique, learners are learned sequentially with early learners fitting simple models to the data and then analysing data for errors. Consecutive trees (random sample) are fit and at every step, the goal is to improve the accuracy from the prior tree. When an input is misclassified by a hypothesis, its weight is increased so that next hypothesis is more likely to classify it correctly. This process converts weak learners into better performing model.

Boosting Steps:

-Draw a random subset of training samples d1 without replacement from the training set D to train a weak learner C1.

-Draw second random training subset d2 without replacement from the training set and add 50 percent of the samples that were previously falsely classified/misclassified to train a weak learner C2.

-Find the training samples d3 in the training set D on which C1 and C2 disagree to train a third weak learner C3

-Combine all the weak learners via majority voting.

Advantages:

-Supports different loss function (we have used ‘binary: logistic’ for this example).

-Works well with interactions.

Disadvantages:

-Prone to over-fitting.

Requires careful tuning of different hyper-parameters.

8—

The Random Forest Classifier is trained using bootstrap aggregation, where each new tree is fit from a bootstrap sample of the training observations zi=(xi,yi). The out-of-bag (OOB) error is the average error for each zi calculated using predictions from the trees that do not contain zi in their respective bootstrap sample. This allows the Random Forest Classifier to be fit and validated whilst being trained.

9-- **Cross-validation** is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation. Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

**Configuration of k:**The k value must be chosen carefully for your data sample. A poorly chosen value for k may result in a mis-representative idea of the skill of the model, such as a score with a high variance (that may change a lot based on the data used to fit the model), or a high bias, (such as an overestimate of the skill of the model).Three common tactics for choosing a value for k are as follows:-

Representative: The value for k is chosen such that each train/test group of data samples is large enough to be statistically representative of the broader dataset.

k=10: The value for k is fixed to 10, a value that has been found through experimentation to generally result in a model skill estimate with low bias a modest variance.

k=n: The value for k is fixed to n, where n is the size of the dataset to give each test sample an opportunity to be used in the hold out dataset. This approach is called leave-one-out cross-validation.

10—

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters. However, there is another kind of parameters, known as Hyperparameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

Some examples of model hyperparameters include:

1. The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization
2. The learning rate for training a neural network.
3. The C and sigma hyperparameters for support vector machines.
4. The k in k-nearest neighbors.

Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are:GridSearchCV and RandomizedSearchCV.

12—

In [statistics](https://en.wikipedia.org/wiki/Statistics) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning), the bias–variance tradeoff is the property of a set of predictive models whereby models with a lower [bias](https://en.wikipedia.org/wiki/Bias_of_an_estimator) in [parameter](https://en.wikipedia.org/wiki/Statistical_parameter) [estimation](https://en.wikipedia.org/wiki/Estimation_theory) have a higher [variance](https://en.wikipedia.org/wiki/Variance) of the parameter estimates across [samples](https://en.wikipedia.org/wiki/Sample_(statistics)), and vice versa. The bias–variance dilemma or bias–variance problem is the conflict in trying to simultaneously minimize these two sources of [error](https://en.wikipedia.org/wiki/Errors_and_residuals_in_statistics) that prevent [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) algorithms from generalizing beyond their [training set](https://en.wikipedia.org/wiki/Training_set):

-The [*bias error*](https://en.wikipedia.org/wiki/Bias_of_an_estimator) is an error from erroneous assumptions in the learning [algorithm](https://en.wikipedia.org/wiki/Algorithm). High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

-The [*variance*](https://en.wikipedia.org/wiki/Variance) is an error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random [noise](https://en.wikipedia.org/wiki/Noise_(signal_processing)) in the training data, rather than the intended outputs ([overfitting](https://en.wikipedia.org/wiki/Overfitting)).

13-- Regularisation is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting.  
The commonly used regularisation techniques are :

1. L1 regularisation
2. L2 regularisation
3. Dropout regularisation

14—

**Adaboost** is more about ‘voting weights’ and Gradient boosting is more about ‘adding gradient optimization’. Adaboost increases the accuracy by giving more weightage to the target which is misclassified by the model. At each iteration, Adaptive boosting algorithm changes the sample distribution by modifying the weights attached to each of the instances. It increases the weights of the wrongly predicted instances and decreases the ones of the correctly predicted instances.  
  
**Gradient boosting** calculates the gradient (derivative) of the Loss Function with respect to the prediction (instead of the features). Gradient boosting increases the accuracy by minimizing the Loss Function (error which is difference of actual and predicted value) and having this loss as target for the next iteration. Gradient boosting algorithm builds first weak learner and calculates the Loss Function. It then builds a second learner to predict the loss after the first step. The step continues for third learner and then for fourth learner and so on until a certain threshold is reached.

15—

Linear regression is suitable for predicting output that is continuous value, such as predicting the price of a property. Its prediction output can be any real number, range from negative infinity to infinity. The regression line is generally a straight line.

Whereas logistic regression is for classification problems, which predicts a probability range between 0 to 1. For example, predict whether a customer will make a purchase or not.