MACHINE LEARNING

ASSIGNMENT - 5

Q1 to Q15 are subjective answer type questions, Answer them briefly.

1. R-squared or Residual Sum of Squares (RSS) which one of these two is a better measure of

goodness of fit model in regression and why?

Ans: R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively. R-squared measures the strength of the relationship between your model and the dependent variable on a convenient 0 – 100% scale.

.After fitting a linear regression model, you need to determine how well the model fits the data. Does it do a good job of explaining changes in the dependent variable? There are several key goodness-of-fit statistics for regression analysis. In this post, we'll examine R-squared (R2), highlight some of its limitations, and discover some surprises. For instance, small R-squared values are not always a problem, and high R-squared values are not necessarily good!

Linear regression identifies the equation that produces the smallest difference between all the observed values and their fitted values. To be precise, linear regression finds the smallest sum of squared residuals that is possible for the dataset.

Statisticians say that a regression model fits the data well if the differences between the observations and the predicted values are small and unbiased. Unbiased in this context means that the fitted values are not systematically too high or too low anywhere in the observation space.

However, before assessing numeric measures of goodness-of-fit, like R-squared, you should evaluate the residual plots. Residual plots can expose a biased model far more effectively than the numeric output by displaying problematic patterns in the residuals. If your model is biased, you cannot trust the results. If your residual plots look good, go ahead and assess your R-squared and other statistics.

Read my post about checking the residual plots.

R-squared and the Goodness-of-Fit

R-squared evaluates the scatter of the data points around the fitted regression line. It is also called the coefficient of determination, or the coefficient of multiple determination for multiple regression. For the same data set, higher R-squared values represent smaller differences between the observed data and the fitted values.

R-squared is the percentage of the dependent variable variation that a linear model explains.

{\displaystyle R^2 = \frac {\text{Variance explained by the model}}{\text{Total variance}}}

R-squared is always between 0 and 100%:

0% represents a model that does not explain any of the variation in the response variable around its mean. The mean of the dependent variable predicts the dependent variable as well as the regression model.

100% represents a model that explains all the variation in the response variable around its mean.

Usually, the larger the R2, the better the regression model fits your observations. However, this guideline has important caveats that I'll discuss in both this post and the next post.

Related post: What are Independent and Dependent Variables?

Visual Representation of R-squared

To visually demonstrate how R-squared values represent the scatter around the regression line, you can plot the fitted values by observed values.

Graph that illustrates a regression model with a low R-squared.Graph that illustrates a model with a high R-squared.

The R-squared for the regression model on the left is 15%, and for the model on the right it is 85%. When a regression model accounts for more of the variance, the data points are closer to the regression line. In practice, you'll never see a regression model with an R2 of 100%. In that case, the fitted values equal the data values and, consequently, all the observations fall exactly on the regression line.

R-squared has Limitations

You cannot use R-squared to determine whether the coefficient estimates and predictions are biased, which is why you must assess the residual plots.

R-squared does not indicate if a regression model provides an adequate fit to your data. A good model can have a low R2 value. On the other hand, a biased model can have a high R2 value!

Are Low R-squared Values Always a Problem?

No! Regression models with low R-squared values can be perfectly good models for several reasons.

Some fields of study have an inherently greater amount of unexplainable variation. In these areas, your R2 values are bound to be lower. For example, studies that try to explain human behavior generally have R2 values less than 50%. People are just harder to predict than things like physical processes.

Fortunately, if you have a low R-squared value but the independent variables are statistically significant, you can still draw important conclusions about the relationships between the variables. Statistically significant coefficients continue to represent the mean change in the dependent variable given a one-unit shift in the independent variable. Clearly, being able to draw conclusions like this is vital.

Linear regression identifies the equation that produces the smallest difference between all the observed values and their fitted values. To be precise, linear regression finds the smallest sum of squared residuals that is possible for the dataset.

Statisticians say that a regression model fits the data well if the differences between the observations and the predicted values are small and unbiased. Unbiased in this context means that the fitted values are not systematically too high or too low anywhere in the observation space.

However, before assessing numeric measures of goodness-of-fit, like R-squared, you should evaluate the residual plots. Residual plots can expose a biased model far more effectively than the numeric output by displaying problematic patterns in the residuals. If your model is biased, you cannot trust the results. If your residual plots look good, go ahead and assess your R-squared and other statistics.

Read my post about checking the residual plots.

R-squared and the Goodness-of-Fit

R-squared evaluates the scatter of the data points around the fitted regression line. It is also called the coefficient of determination, or the coefficient of multiple determination for multiple regression. For the same data set, higher R-squared values represent smaller differences between the observed data and the fitted values.

R-squared is the percentage of the dependent variable variation that a linear model explains.

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$$

R-squared is always between 0 and 100%:

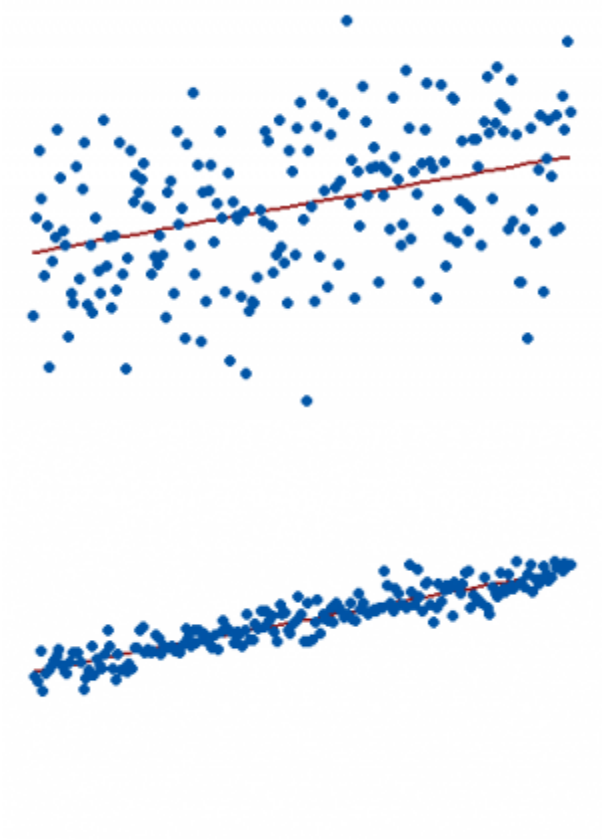
- 0% represents a model that does not explain any of the variation in the response variable around its mean. The mean of the dependent variable predicts the dependent variable as well as the regression model.
- 100% represents a model that explains all the variation in the response variable around its mean.

Usually, the larger the R^2, the better the regression model fits your observations. However, this guideline has important caveats that I'll discuss in both this post and the next post.

Related post: What are Independent and Dependent Variables?

Visual Representation of R-squared

To visually demonstrate how R-squared values represent the scatter around the regression line, you can plot the fitted values by observed values.



The R-squared for the regression model on the left is 15%, and for the model on the right it is 85%. When a regression model accounts for more of the variance, the data points are closer to the regression line. In practice, you'll never see a regression model with an R^2 of 100%. In that case, the fitted values equal the data values and, consequently, all the observations fall exactly on the regression line.

R-squared has Limitations

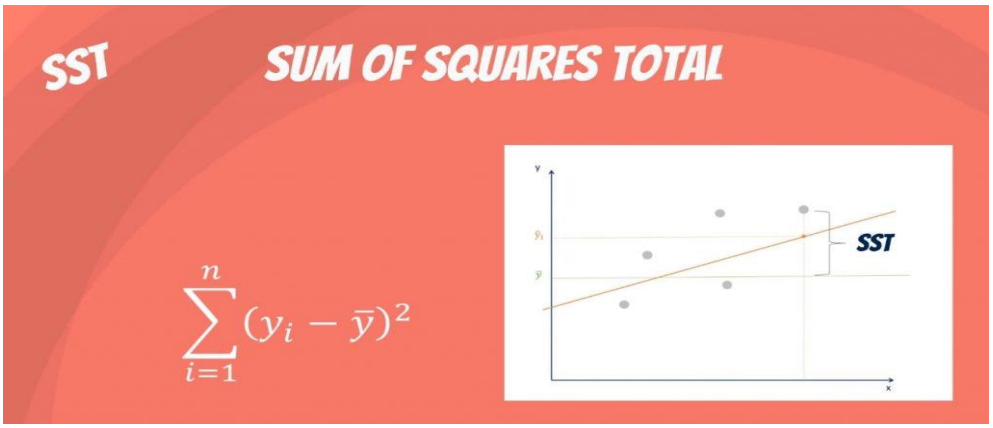
You cannot use R-squared to determine whether the coefficient estimates and predictions are biased, which is why you must assess the residual plots.

R-squared does not indicate if a regression model provides an adequate fit to your data. A good model can have a low R^2 value. On the other hand, a biased model can have a high R^2 value!

2. 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.

Ans: the two notations are SST, SSR, SSE, or TSS, ESS, RSS.

Mathematically, SST = SSR + SSE.



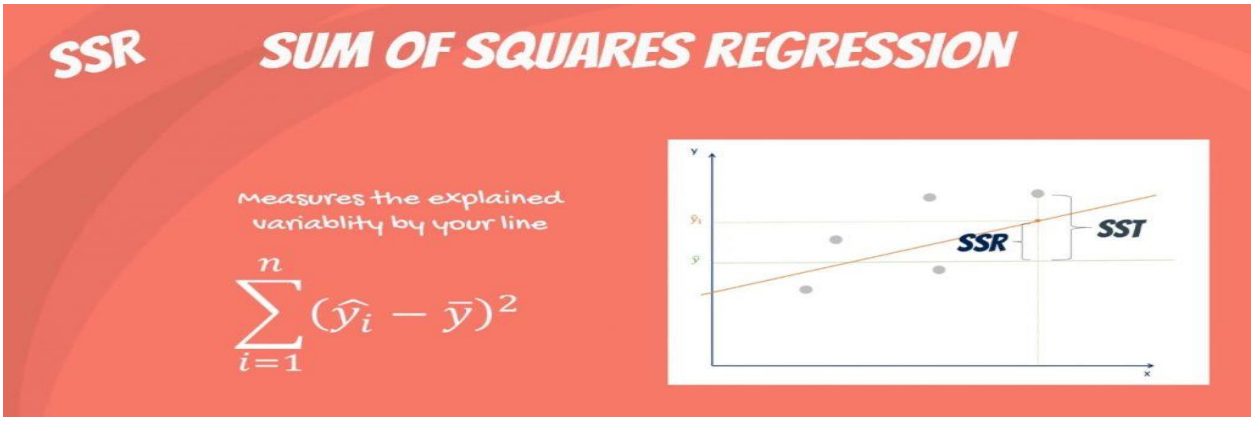
The sum of squares total, denoted SST, is the squared differences between the observed dependent variable and its mean. You can think of this as the dispersion of the observed variables around the mean – much like the variance in descriptive statistics.

It is a measure of the total variability of the dataset.

Side note: There is another notation for the SST. It is TSS or total sum of squares.

What is the SSR?

The second term is the sum of squares due to regression, or SSR. It is the sum of the differences between the predicted value and the mean of the dependent variable. Think of it as a measure that describes how well our line fits the data.

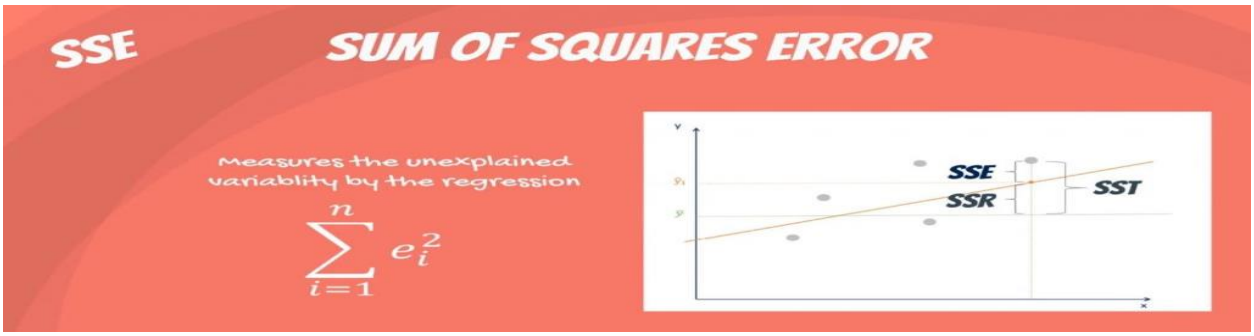


If this value of SSR is equal to the sum of squares total, it means our regression model captures all the observed variability and is perfect. Once again, we have to mention that another common notation is ESS or explained sum of squares.

What is the SSE?

The last term is the sum of squares error, or SSE. The error is the difference between the observed value and the predicted value.

We usually want to minimize the error. The smaller the error, the better the estimation power of the regression. Finally, I should add that it is also known as RSS or residual sum of squares. Residual as in: remaining or unexplained.

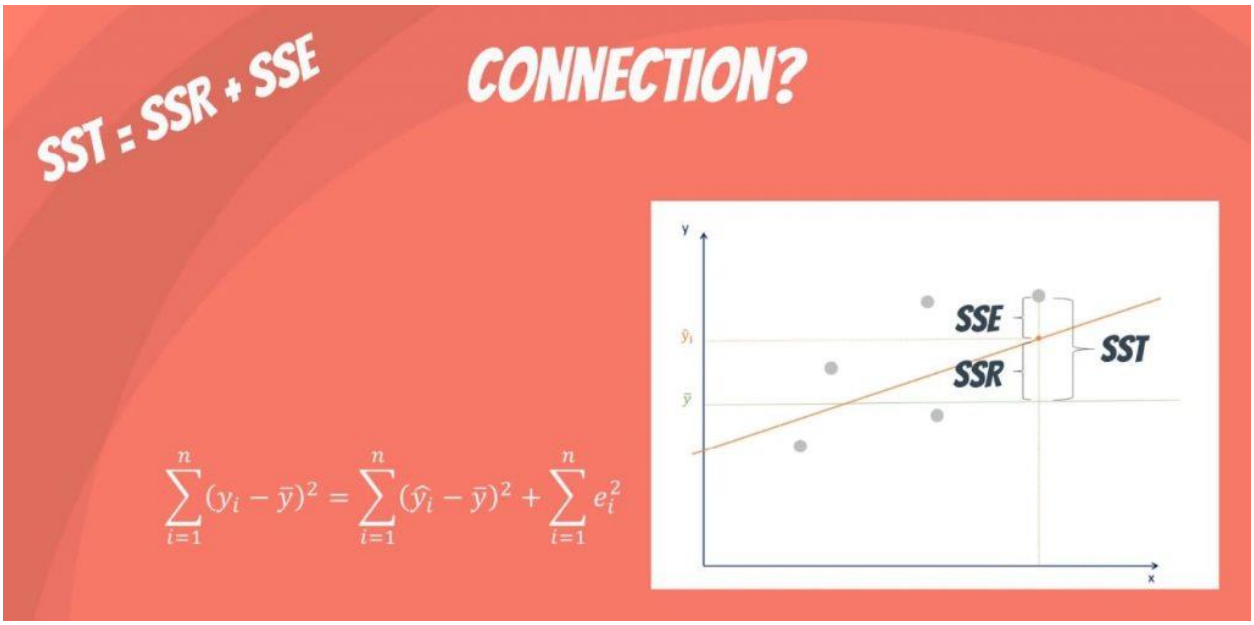


The rationale is the following: the total variability of the data set is equal to the variability explained by the regression line plus the unexplained variability, known as error.

.

How Are They Related?

Mathematically, **SST** = **SSR** + **SSE**.



Given a constant total variability, a lower error will cause a better **regression**. Conversely, a higher error will cause a less powerful **regression**. And that's what you must remember, no matter the notation.

3. What is the need of regularization in machine learning?

Ans: Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.

Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.

Pros of Regularization

Prevents Overfitting: A high-dimensional dataset having too many features can sometimes lead to overfitting (model captures both real and random effects).

Simplicity: An over-complex model having too many features can be hard to interpret especially when features are correlated with each other.

4. What is Gini–impurity index?

Ans: 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 used to measure the distribution of income across a population. A higher Gini index indicates greater inequality, with high-income individuals receiving much larger percentages of the population's total income.

5. Are unregularized decision-trees prone to overfitting? If yes, why?

Ans: Decision trees are a type of model used for both classification and regression. Trees answer sequential questions which send us down a certain route of the tree given the answer. The model behaves with “if this than that” conditions ultimately yielding a specific result. This is easy to see with the image below which maps out whether or not to play golf.

Decision trees are prone to overfitting, especially when a tree is particularly deep. This is due to the amount of specificity we look at leading to smaller sample of events that meet the previous assumptions.

This small sample could lead to unsound conclusions. An example of this could be predicting if the Boston Celtics will beat the Miami Heat in tonight’s basketball game.

The first level of the tree could ask if the Celtics are playing home or away. The second level might ask if the Celtics have a higher win percentage than their opponent, in this case the Heat. The third level asks if the Celtic’s leading scorer is playing? The fourth level asks if the Celtic’s second leading scorer is playing. The fifth level asks if the Celtics are traveling back to the east coast from 3 or more consecutive road games on the west coast. While all of these questions may be relevant, there may only be two previous games where the conditions of tonights game were met.

Using only two games as the basis for our classification would not be adequate for an informed decision. One way to combat this issue is by setting a max depth. This will limit our risk of overfitting; but as always, this will be at the expense of error due to bias.

Thus if we set a max depth of three, we would only ask if the game is home or away, do the Celtics have a higher winning percentage than their opponent, and is their leading scorer playing. This is a simpler model with less variance sample to sample but ultimately will not be a strong predictive model.

6. What is an ensemble technique in machine learning?

Ensemble methods are techniques that aim at improving the accuracy of results in models by combining multiple models instead of using a single model.

The combined models increase the accuracy of the results significantly. This has boosted the popularity of ensemble methods in machine learning.

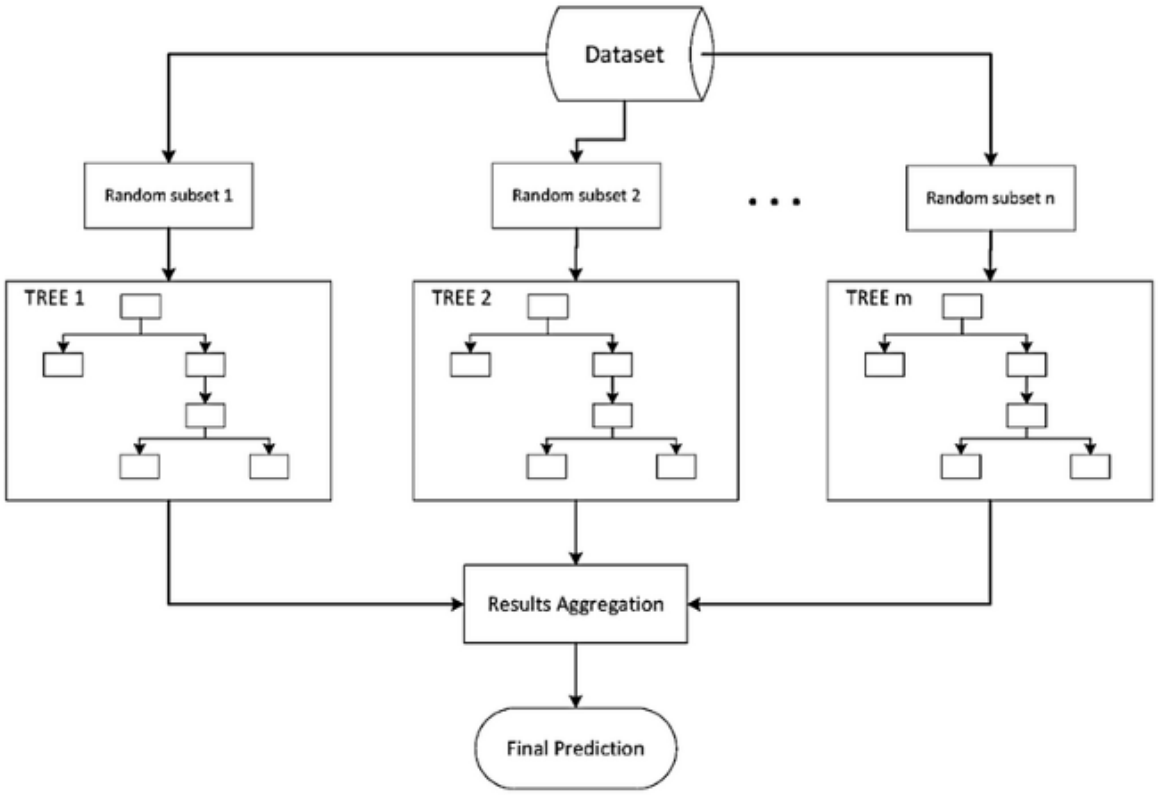
7. What is the difference between Bagging and Boosting techniques?

Ans: Bagging is a method of merging the same type of predictions. Boosting is a method of merging different types of predictions. Bagging decreases variance, not bias, and solves over-fitting issues in a model. Boosting decreases bias, not variance.

Thus, on the one hand, in a Random Forest model, Bagging is used, and the AdaBoost model implies the Boosting algorithm.

Types of Ensemble Methods

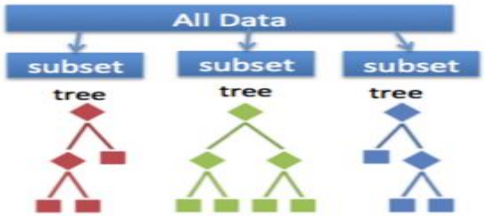
1. ***BAGG***ing, or ***B***ootstrap ***AGG***regating. **BAGG**ing gets its name because it combines ***B***ootstrapping and ***Agg***regation to form one ensemble model. Given a sample of data, multiple bootstrapped subsamples are pulled. A Decision Tree is formed on each of the bootstrapped subsamples. After each subsample Decision Tree has been formed, an algorithm is used to aggregate over the Decision Trees to form the most efficient predictor. The image below will help explain:



Given a Dataset, bootstrapped subsamples are pulled. A Decision Tree is formed on each bootstrapped sample. The results of each tree are aggregated to yield the strongest, most accurate predictor.

2. **Random Forest** Models. Random Forest Models can be thought of as **BAGG**ing, with a slight tweak. When deciding where to split and how to make decisions, BAGGed Decision Trees have the full disposal of features to choose from. Therefore, although the bootstrapped samples may be slightly different, the data is largely going to break off at the same features throughout each model. In contrary, Random Forest models decide where to split based on a random selection of features. Rather than splitting at similar features at each node throughout, Random Forest models implement a level of differentiation because each tree will split based on different features. This level of differentiation provides a greater

ensemble to aggregate over, ergo producing a more accurate predictor. Refer to the image for a better understanding.



A random forest takes a random subset of features from the data, and creates n random trees from each subset. Trees are aggregated together at end.

Similar to BAGGing, bootstrapped subsamples are pulled from a larger dataset. A decision tree is formed on each subsample. HOWEVER, the decision tree is split on different features (in this diagram the features are represented by shapes).

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

Ans: The out-of-bag (OOB) error is the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample. This allows the Random Forest Classifier to be fit and validated whilst being trained

9. What is K-fold cross-validation?

Ans: K-fold Cross-Validation is when the dataset is split into a K number of folds and is used to evaluate the model's ability when given new data. K refers to the number of groups the data sample is split into. For example, if you see that the k-value is 5, we can call this a 5-fold cross-validation.

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

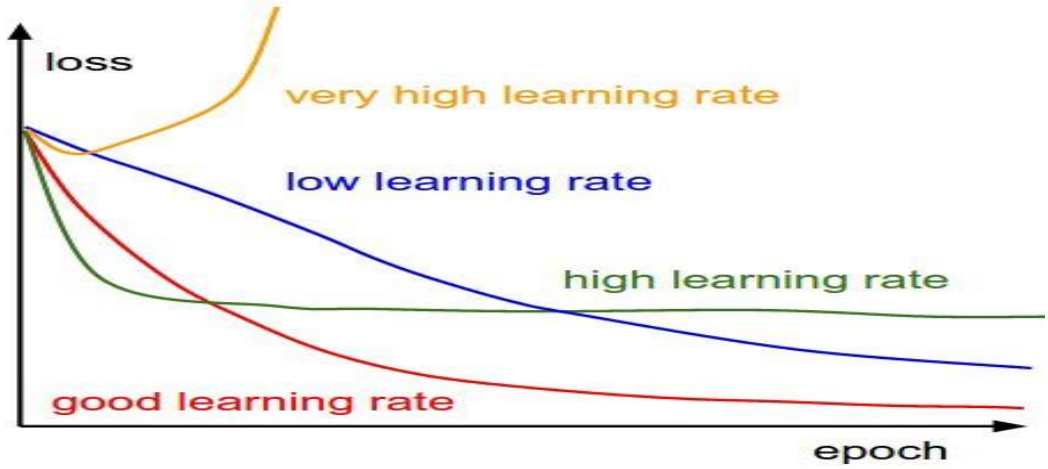
Ans: Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set. That combination of hyperparameters maximizes the model's performance, minimizing a predefined loss function to produce better results with fewer errors.

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

Ans: A learning rate that is too large can cause the model to converge too quickly to a suboptimal solution, whereas a learning rate that is too small can cause the process to get stuck.

The challenge of training deep learning neural networks involves carefully selecting the learning rate.

If the learning rate is very large we will skip the optimal solution. If it is too small we will need too many iterations to converge to the best values. So using a good learning rate is crucial.



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

Ans: No, Logistic Regression has traditionally been used as a linear classifier, i.e. when the classes can be separated in the feature space by linear boundaries.

That can be remedied however if we happen to have a better idea as to the shape of the decision boundary.

Logistic regression is known and used as a linear classifier. It is used to come up with a hyperplane in feature space to separate observations that belong to a class from all the other observations that do not belong to that class. The decision boundary is thus linear. Robust and efficient implementations are readily available (e.g. scikit-learn) to use logistic regression as a linear classifier.

While logistic regression makes core assumptions about the observations such as IID (each observation is independent of the others and they all have an identical probability distribution), the use of a linear decision boundary is not one of them. The linear decision boundary is used for reasons of simplicity following the Zen mantra – when in doubt simplify. In those cases where we suspect the decision boundary to be nonlinear, it may make sense to formulate logistic regression with a nonlinear model and evaluate how much better we can do.

13. Differentiate between Adaboost and Gradient Boosting.

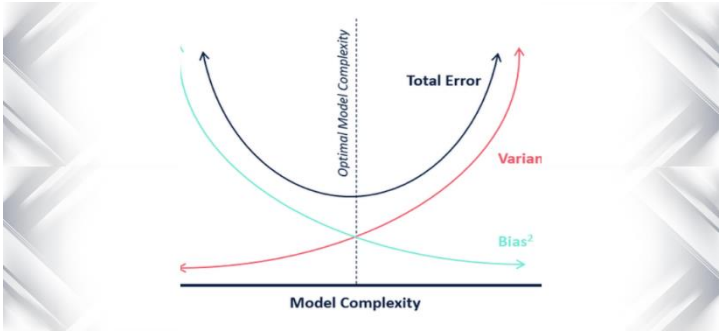
Ans:

AdaBoost	GradientBoost
Both AdaBoost and Gradient Boost use a base weak learner and they try to boost the performance of a weak learner by iteratively shifting the focus towards problematic observations that were difficult to predict. At the end, a strong learner is formed by addition (or weighted addition) of the weak learners.	
In AdaBoost, shift is done by up-weighting observations that were misclassified before.	Gradient boost identifies difficult observations by large residuals computed in the previous iterations.
In AdaBoost "shortcomings" are identified by high-weight data points.	In Gradientboost "shortcomings" are identified by gradients.
Exponential loss of AdaBoost gives more weights for those samples fitted worse.	Gradient boost further dissect error components to bring in more explanation.
AdaBoost is considered as a special case of Gradient boost in terms of loss function, in which exponential losses.	Concepts of gradients are more general in nature.

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

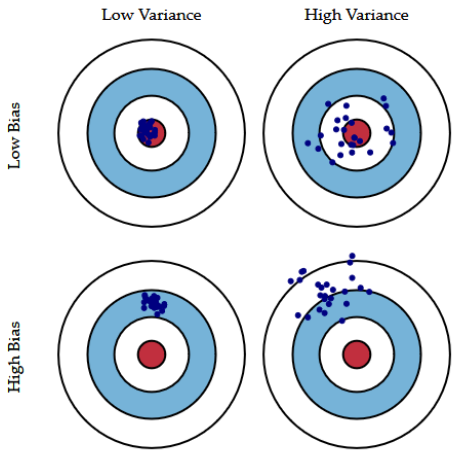
Ans: In statistics and machine learning, the bias–variance tradeoff is the property of a model that the variance of the parameter estimated across samples can be reduced by increasing the bias in the estimated parameters.

Bias is the simplifying assumptions made by the model to make the target function easier to approximate. Variance is the amount that the estimate of the target function will change given different training data. Trade-off is tension between the error introduced by the bias and the variance.



Though some points are classified incorrectly, the model generally fits most of the datapoints accurately. The balance between the Bias error and the Variance error is the **Bias-Variance Tradeoff**.

The following bulls-eye diagram explains the tradeoff better:



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

Kernel plays a vital role in classification and is used to analyze some patterns in the given dataset.

They are very helpful in solving a no-linear problem by using a linear classifier.

Later the svm algorithm uses kernel-trick for transforming the data points and creating an optimal decision boundary.

Kernels help us to deal with high dimensional data in a very efficient manner.

We have various svm kernel functions to convert the non-linear data to linear.

listed 7 such popular svm kernel functions.

SVM algorithms use a group of mathematical functions that are known as kernels. The function of a kernel is to require data as input and transform it into the desired form.

Different SVM algorithms use differing kinds of kernel functions. These functions are of different kinds—for instance, linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.

The most preferred kind of kernel function is RBF. Because it's localized and has a finite response along the complete x-axis.

The kernel functions return the scalar product between two points in an exceedingly suitable feature space. Thus by defining a notion of resemblance, with a little computing cost even in the case of very high-dimensional spaces.

Linear Kernel

It is the most basic type of kernel, usually one dimensional in nature. It proves to be the best function when there are lots of features. The linear kernel is mostly preferred for **text-classification problems** as most of these kinds of classification problems can be linearly separated.

Linear kernel functions are **faster** than other functions.

Linear Kernel Formula

$$F(x, xj) = sum(x.xj)$$

Here, **x, xj** represents the data you're trying to classify.

Polynomial Kernel

It is a more generalized representation of the linear kernel. It **is not** as preferred as other kernel functions as it is **less efficient** and accurate.

Polynomial Kernel Formula

$$F(x, xj) = (x.xj+1)^d$$

Here '.' shows the **dot product** of both the values, and **d** denotes the degree.

F(x, xj) representing the **decision boundary** to separate the given classes.

Gaussian Radial Basis Function (RBF)

It is one of the most preferred and used kernel functions in svm. It is usually chosen for non-linear data. It helps to make proper separation when there is no prior knowledge of data.

Gaussian Radial Basis Formula

$$\mathbf{F(x, xj) = exp(-gamma * ||x - xj||^2)}$$

The value of gamma varies from **0 to 1**. You have to manually provide the value of gamma in the code. The most preferred value for **gamma is 0.1**.

Sigmoid Kernel

It is mostly preferred for **neural networks**. This kernel function is similar to a two-layer perceptron model of the neural network, which works as an **activation function** for neurons.

It can be shown as,

Sigmoid Kenel Function

$$\mathbf{F(x, xj) = tanh(\alpha xay + c)}$$

Gaussian Kernel

It is a commonly used kernel. It is used when there is no prior knowledge of a given dataset.

Gaussian Kernel Formula

Bessel function kernel

It is mainly used for removing the cross term in mathematical functions.

Bassel Kernel Formula

Here J is the Bessel function.

ANOVA kernel

It is also known as a radial basis function kernel. It usually performs well in multidimensional regression problems.

Anova Kernel Formula

to choose the best SVM kernel for your dataset

Because a linear kernel takes less training time when compared to other kernel functions.

- The **linear kernel** is mostly preferred for text classification problems as it performs well for large datasets.
- **Gaussian kernels** tend to give good results when there is no additional information regarding data that is not available.
- **Rbf kernel** is also a kind of Gaussian kernel which projects the high dimensional data and then searches a linear separation for it.

- **Polynomial kernels** give good results for problems where all the training data is normalized.

Advantages of SVM

- It works well on a dataset having many features.
- It provides a clear margin of separation.
- It is very effective for the dataset where the number of features are greater than the data points.
- You can specify different kernel functions to make a proper decision boundary.

Disadvantages of SVM

- It requires very high training time, hence not recommended for large datasets.
- It is very sensitive to outliers.