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?

   R-squared is always between 0 and 100%: 0% indicates that the model explains none of the variability of the response data around its mean. 100% indicates that the model explains all the variability of the response data around its mean. In general, the higher the R-squared, the better the model fits your data.

Residual Sum of Squares (RSS) is a statistical method that helps identify the level of discrepancy in a dataset not predicted by a regression model. Thus, it measures the variance in the value of the observed data when compared to its predicted value as per the regression model. Hence, RSS indicates whether the regression model fits the actual dataset well or not.

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.

In statistical data analysis the total sum of squares (TSS or SST) is a quantity that appears as part of a standard way of presenting results of such analyses. For a set of observations, , it is defined as the sum over all squared differences between the observations and their overall mean .:[1]

Explained sum of square (ESS) or Regression sum of squares or Model sum of squares is a statistical quantity used in modeling of a process. ESS gives an estimate of how well a model explains the observed data for the process.

It tells how much of the variation between observed data and predicted data is being explained by the model proposed. Mathematically, it is the sum of the squares of the difference between the predicted data and mean data.

Let $y_i = a + b_1x_{1i} + b_2x_{2i} + ... + \varepsilon_i$ is regression model, where:

y_i is the i^{th} observation of the response variable

x_{ji} is the i^{th} observation of the j^{th} explanatory variable

a and b_i are coefficients

i indexes the observations from 1 to n

ε_i is the i^{th} value of the error term

Then

$$\mathrm{ESS} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 .$$

This is usually used for regression models. The variation in the modeled values is contrasted with the variation in the observed data (total sum of squares) and variation in modeling errors (residual sum of squares). The result of this comparison is given by ESS as per the following equation:

ESS = total sum of squares – residual sum of squares

As a generalization, a high ESS value signifies greater amount of variation being explained by the model, hence meaning a better model.

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

Regularization describes methods for calibrating machine learning models to reduce the adjusted loss function and avoid overfitting or underfitting.

We can properly fit our machine learning model on a given test set using regularization, which lowers the errors in the test set.

A penalty or complexity term is added to the complex model during regularization. Let's consider the simple linear regression equation:

$y= \beta0+\beta1\times1+\beta2\times2+\beta3\times3+\cdots+\beta nxn +b$

In the above equation, Y represents the value to be predicted

Features for Y are X1, X2, and Xn.

β0,β1,…..βn are the weights or magnitude attached to the features, respectively. Here, stands for the model's bias, and b stands for the intercept.

Now, in order to create a model that can accurately predict the value of Y, we will add a loss function and optimize a parameter. The loss function for the linear regression is called RSS or residual square sum.

2 Regularization Techniques

Ridge Regularization and Lasso Regularization are the two main categories of regularization techniques.

Ridge Regularization

It is also referred to as Ridge Regression and modifies over- or under-fitted models by adding a penalty equal to the sum of the squares of the coefficient magnitude.

In other words, the mathematical function that represents our machine learning model is minimized, and coefficients are computed. It is squared and added how big the coefficients are. By reducing the number of coefficients, Ridge Regression applies regularization.

Lambda λ is used to represent the penalty term in the cost function. We can control the penalty term by modifying the values of the penalty function. The penalty's severity affects how much the coefficients are reduced. The parameters are trimmed. In order to avoid multicollinearity, it is used. Additionally, it causes coefficient shrinkage, which lessens the complexity of the model.

Lasso Regression

By introducing a penalty equal to the sum of the absolute values of the coefficients, it modifies the over- or under-fitted models.

Coefficient minimization is also carried out by lasso regression, but the true coefficient values are used rather than the squared coefficient magnitudes. In light of the fact that there are negative coefficients, the coefficient sum can therefore also be 0.

Key Difference Between Ridge Regression And Lasso Regression

- Ridge regression is mostly used to reduce overfitting in the model, and it includes all the features present in the model. The coefficients are shrunk, which lowers the model's complexity.
- Lasso regression helps to reduce the overfitting in the model as well as feature selection.

What Is The Purpose Of Regularization?

Variance, i.e. variability, is a characteristic of a standard least squares model. this model won't generalize well for a data set different than its training data. Regularization, significantly reduces the variance of the model, without a substantial increase in its bias. Therefore, the regularization techniques described above use the tuning parameter λ to control the effect of bias and variance. As the value of lambda increases, the value of the coefficients decreases, lowering the variance. *Up to a point, this increase in λ is advantageous because it only reduces variance (avoiding overfitting) while maintaining all of the data's significant properties.* But once the value reaches a certain point, the model begins to lose crucial characteristics, leading to bias and underfitting. As a result, care should be taken when choosing the value of &lambda.

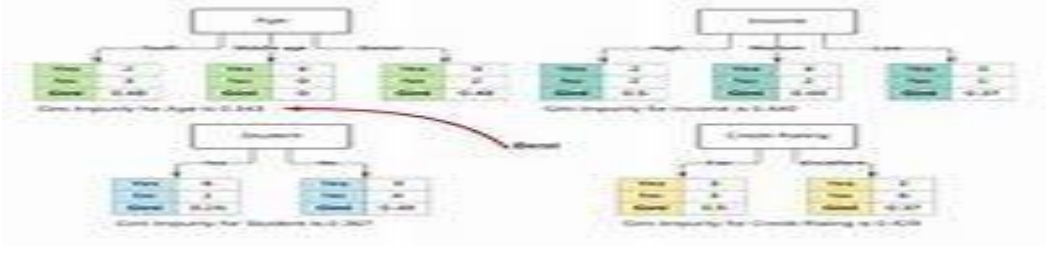
You won't require anything more fundamental to start with regularization than this. It is a practical method that can help increase the precision of your regression models. A popular library for implementing these algorithms is Scikit-Learn. It has a wonderful API that can get your model up and running with just a few lines of code in python.

If you liked this article, be sure to show your support by clapping for this article below and if you have any questions, leave a comment and In my best effort, I will respond.

For being more aware of the world of machine learning, follow me. It is the most effective way to learn when I publish new articles similar to this one.

4. What is Gini–impurity index?

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.

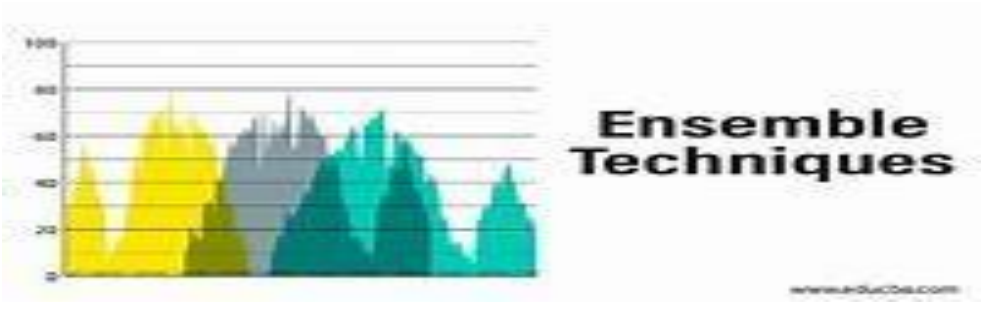


5. Are unregularized decision-trees prone to overfitting? If yes, why?

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.

6. What is an ensemble technique in machine learning?

Ensemble learning is a technique in machine learning which takes the help of several base models and combines their output to produce an optimized model. This type of machine learning algorithm helps in improving the overall performance of the model. Here the base model which is most commonly used is the Decision tree classifier.



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

difference between bagging and boosting is in how they are used. For example, bagging methods are typically used on weak learners that exhibit high variance and low bias, whereas boosting methods are leveraged when low variance and high bias is observed.

1.Bagging is a learning approach that aids in enhancing the performance, execution, and precision of machine learning algorithms. Boosting is an approach that iteratively modifies the weight of observation based on the last classification

2.It is the easiest method of merging predictions that belong to the same type.It is a method of merging predictions that belong to different types.

3.Here, every model has equal weight.Here, the weight of the models depends on their performance.

4.In bagging, each model is assembled independently. In boosting, the new models are impacted by the implementation of earlier built models.

5.It helps in solving the over-fitting issue.It helps in reducing the bias.

6.In the case of bagging, if the classifier is unstable, then we apply bagging.In the case of boosting, If the classifier is stable, then we apply boosting.

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

Out-of-Bag Error in Random Forest The out-of-bag error is the average error for each predicted outcome calculated using predictions from the trees that do not contain that data point in their respective bootstrap sample. This way, the Random Forest model is constantly being validated while being trained.

Outlook	Temperature	Humidity	Wind	Play Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Hot	High	Weak	Yes
Windy	Cold	Low	Weak	Yes

Bootstrap sample

9. What is K-fold cross-validation?

Cross validation is an evaluation method used in machine learning to find out how well your machine learning model can predict the outcome of unseen data. It is a method that is easy to comprehend, works well for a limited data sample and also offers an evaluation that is less biased, making it a popular choice.

The data sample is split into ‘k’ number of smaller samples, hence the name: K-fold Cross Validation. You may also hear terms like four fold cross validation, or ten fold cross validation, which essentially means that the sample data is being split into four or ten smaller samples respectively.

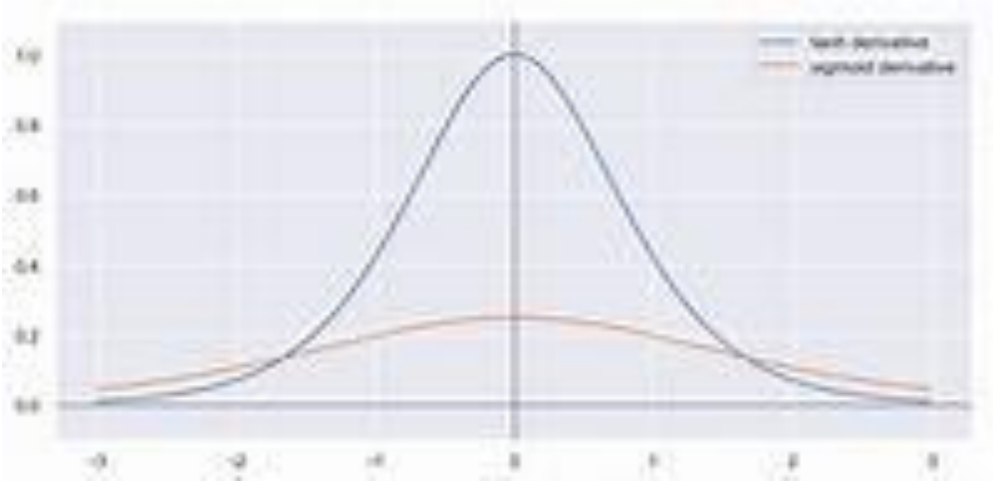
10. What is hyper parameter tuning in machine learning and why it is done?

Hyperparameters are the knobs or settings that can be tuned before running a training job to control the behavior of an ML algorithm. They can have a big impact on model training as it relates to training time, infrastructure resource requirements (and as a result cost), model convergence and model accuracy.

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

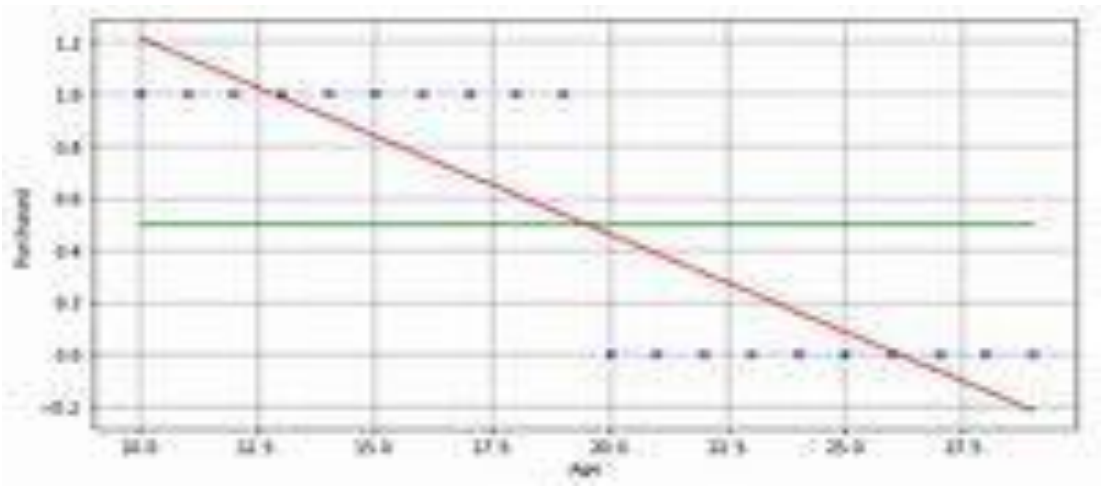
The learning rate can seen as step size, η. As such, gradient descent is taking successive steps in the direction of the minimum. If the step size η is too large, it can (plausibly) "jump over" the minima we are trying to reach, ie. we overshoot.

Gradient descent explodes if learning rate is too large



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

Yes, it might work, but logistic regression is more suitable for classification task and we want to prove that logistic regression yields better results than linear regression. Let’s see how logistic regression classifies our dataset. Logistic regression model, a sigmoid curve that fit the training dataset



13. Differentiate between Adaboost and Gradient Boosting.

Boosting makes decision trees cool again. Here, gradient boosting and adaboost are the most common boosting techniques for decision tree based machine learning. In this post, we are going to compare those two boosting techniques and explain the similar and different parts of them.

Naming

Gradient boosting comes from boosting results with gradient descent algorithm. Some sources mention this gradient boosting machines or gradient boosting decision trees.

Adaboost is the acronym of the adaptive boosting.

Boosting

The both gradient boosting and adaboost build many decision trees. I mean that they build a tree and build another one with the error of the previous one.

The way gradient boosting boosting its results is to find the difference between prediction and actual value of an instance. Then, the target label of that instance will be replaced with this subtraction in the next round. Difference between actual and prediction comes from the derivative of the mean squared error as a loss function.

Adaboost applies a similar procedure. It builds a decision tree, then it will increase the target label for incorrectly predicted ones, and it will decrease the target label value for correctly predicted ones.

In this way, predictions with high error will be more important in the next rounds in those techniques both.

Weights

In gradient boosting, each tree has a same weight. To make a final decision, we will find the sum of the predictions of those sequential trees.

On the other hand, trees have weights in adaboost. Each tree will contribute to the prediction with respect to its weight.

Decision tree algorithm

The both gradient boosting and adaboost run regression trees. No matter what kind of a data set you have. It does not matter having a regression or classification problem. You have to transform classification data set to regression task firstly.

Tree depth

Adaboost builds one-depth regression tree (or decision stumps). It just expects 51% accuracy score as a prerequisite.

On the other hand, gradient boosting build trees with higher depth. For instance, lightgbm and xgboost are popular gradient boosting implementations. The maximum depth of a tree is set to 5 in the default configuration for these libraries.

Linearity

Built trees are linear models in adaboost because it builds a tree with one depth. However, predictions will be made with the combination of many linear models and it will be non-linear anyway.

On the other hand, built trees are already non-linear in gradient boosting. So, final predictions will be non-linear as well.

Adoption

Nowadays, gradient boosting is highly adopted in daily data science competitions. It can even battle with deep learning in many Kaggle challenges.

On the other hand, adaboost is a legacy technique and it is not adopted as common as gradient boosting. Still, it appears in the face and eye detection module (haar cascade) of opencv.

Conclusion

So, we have mentioned two important boosting techniques in tree-based machine learning. The both gradient boosting and adaboost are very similar to the moving heavy rocks. A poor employee cannot move a heavy rock but poor employees come together and move a heavy rock. This is the idea behind boosting!

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

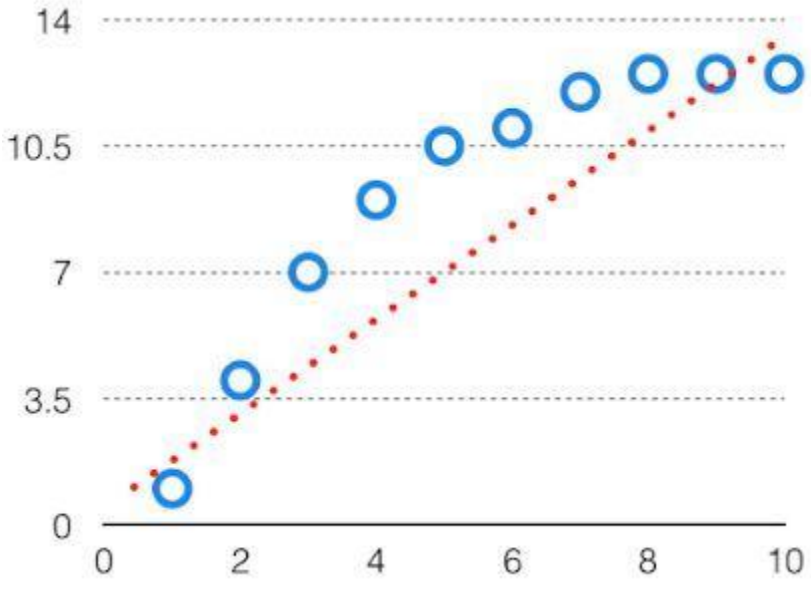
It is important to understand prediction errors (bias and variance) when it comes to accuracy in any machine learning algorithm. There is a tradeoff between a model’s ability to minimize bias and variance which is referred to as the best solution for selecting a value of Regularization constant. Proper understanding of these errors would help to avoid the overfitting and underfitting of a data set while training the algorithm.

Bias

The bias is known as the difference between the prediction of the values by the ML model and the correct value. Being high in biasing gives a large error in training as well as testing data. Its recommended that an algorithm should always be low biased to avoid the problem of underfitting.

By high bias, the data predicted is in a straight line format, thus not fitting accurately in the data in the data set. Such fitting

is known as Underfitting of Data. This happens when the hypothesis is too simple or linear in nature. Refer to the graph given below for an example of such a situation.



HighBias

In such a problem, a hypothesis looks like follows.

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

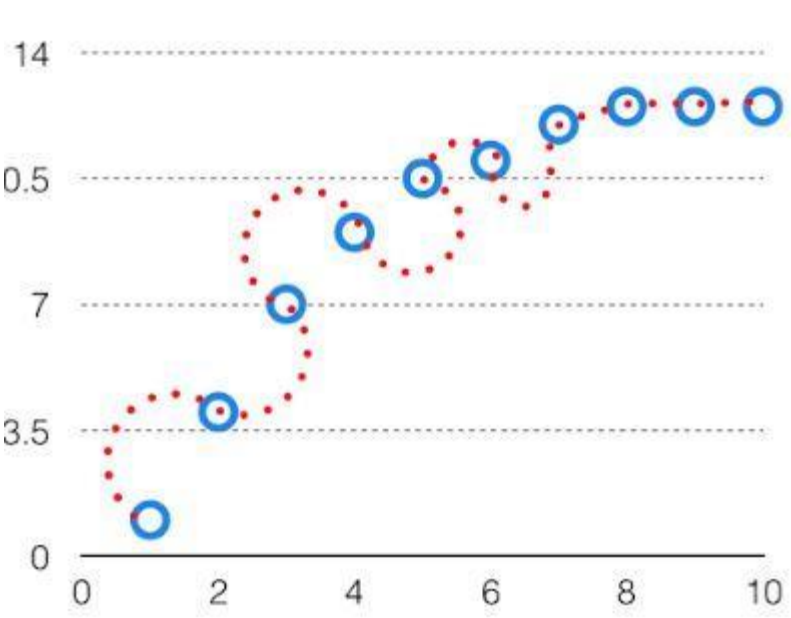
Variance

The variability of model prediction for a given data point which tells us spread of our data is called the variance of the model. The model with high variance has a very complex fit to the training data and thus is not able to fit accurately on the data which it hasn’t seen before. As a result, such models perform very well on training data but has high error rates on test data.

When a model is high on variance, it is then said to as Overfitting of Data. Overfitting is fitting the training set accurately via complex curve and high order hypothesis but is not the solution as the error with unseen data is high.

While training a data model variance should be kept low.

The high variance data looks like follows.



High Variance

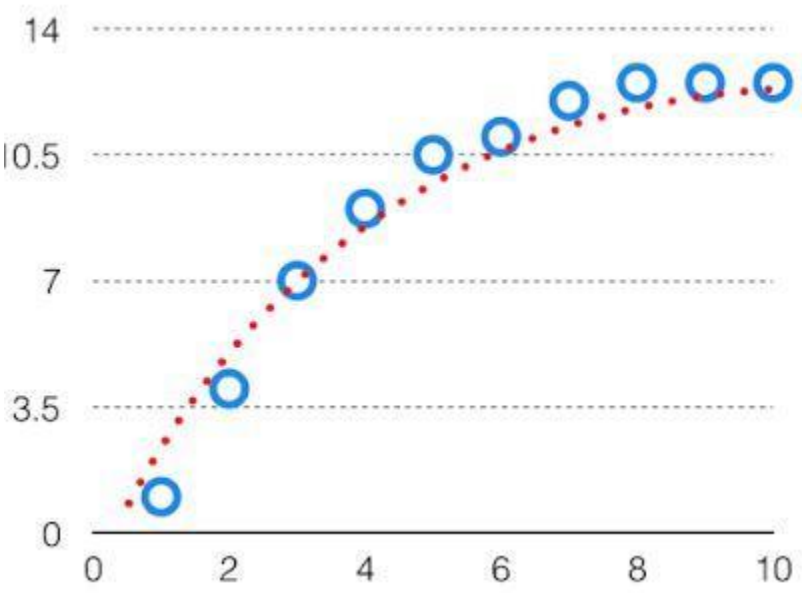
In such a problem, a hypothesis looks like follows.

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Bias Variance Tradeoff

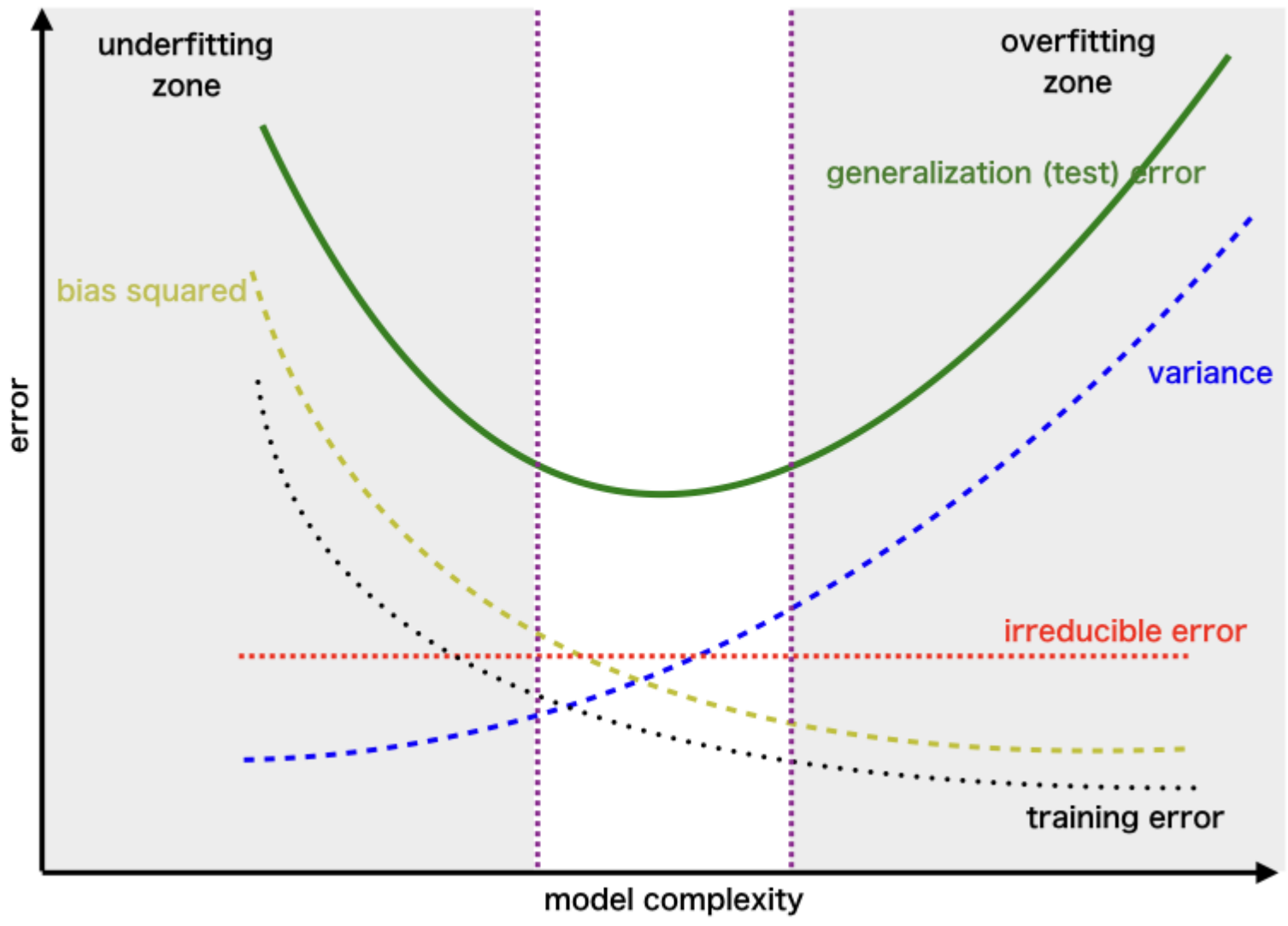
If the algorithm is too simple (hypothesis with linear eq.) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree eq.) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as Trade-off or Bias Variance Trade-off.

This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like.



The best fit will be given by hypothesis on the tradeoff point.

The error to complexity graph to show trade-off is given as –



This is referred to as the best point chosen for the training of the algorithm which gives low error in training as well as testing data.

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

SVM is a famous supervised machine learning algorithm used for classification as well as regression algorithms. However, mostly it is preferred for classification algorithms. It basically separates different target classes in a hyperplane in n-dimensional or multidimensional space.

The main motive of the SVM is to create the best decision boundary that can separate two or more classes(with maximum margin) so that we can correctly put new data points in the correct class.

Wait!

Why is it known as SVM?

Because It chooses extreme vectors or support vectors to create the hyperplane, that’s why it is named so. In the below sections let's understand in more detail.

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.

$$K\left(\overline{x}\right) = \begin{matrix} 1 & \text{if} \left\|\overline{x}\right\| \leq 1 \\ 0 & \text{otherwise} \end{matrix}$$

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

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

F(x, xj) = tanh(αxay + 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

$$k(x,y) = \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$$

Bessel function kernel

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

Bassel Kernel Formula

$$k(x,y) = \frac{J_{v+1}(\sigma||x-y||)}{||x-y||^{-n(v+1)}}$$

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

$$k(x, y) = \sum_{k=1}^{n} \exp(-\sigma(x^k - y^k)^2)^d$$