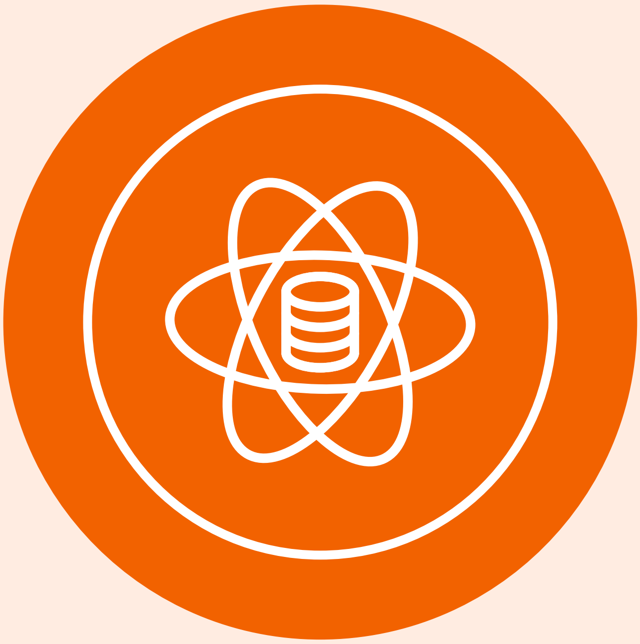
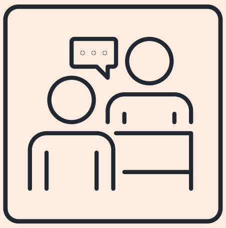
edureka! LEARNING CENTER



Data Science Program

Model Evaluation and Optimization

Interview Preparation



Model Selection and Model Evaluation Interview Preparation

The intersection of computational modeling and statistical analysis lies at the heart of the field of machine learning. The concept of algorithms or models, which are in reality, statistical estimations, is the crux of machine learning and serves as its central organizing principle.

Nevertheless, regardless of the data distribution, each and every model has a number of inherent constraints. Due to the fact that they are all only guesses, none of them can be absolutely precise. These shortcomings are commonly referred to as bias and variation in the scientific community.

The problem arises when the limitations are subtle, such as when we have to choose between a random forest algorithm and a gradient boosting algorithm or between two variations of the same decision tree algorithm. In these situations, we are faced with a choice between a random forest algorithm and a gradient boosting algorithm. Both are likely to have a large amount of variance but very little bias.

Here is where the processes of model selection and model evaluation come into play. Follow through to understand the interview related questions in this domain.

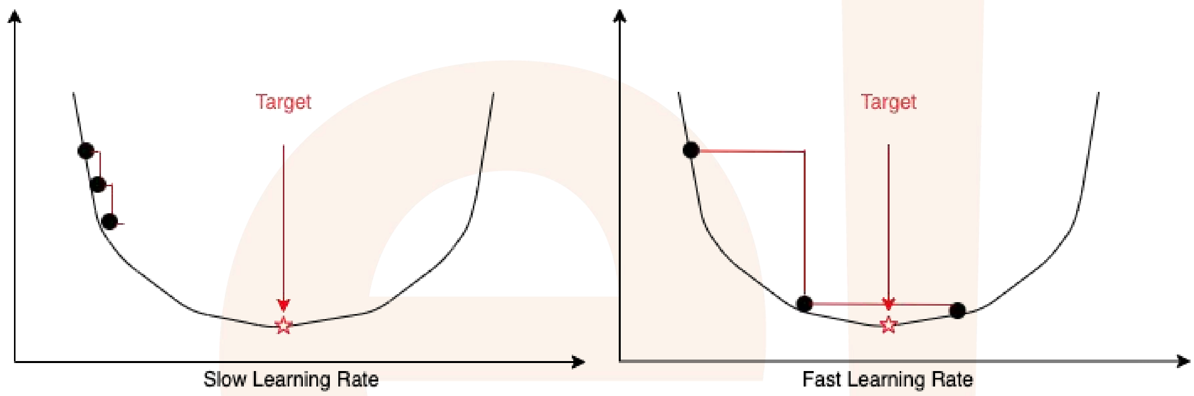
Q1. What are Hyper-Parameters in ML Model?

Answer: Every machine learning model has parameters and *can* additionally have hyper-parameters. **Hyper-parameters** are those parameters that cannot be directly learned from the regular training process. These parameters express **higher-level** properties of the model such as its *complexity* or *how fast it should learn*.

If machine learning model was an AM radio, the knobs for tuning the station would be its parameters but things like angle of antenna, height of antenna, volume knob would be hyperparameters.

Q2. What is a model Learning Rate? Is a high learning rate always good?

Answer: The **Learning Rate** is a tuning parameter that determines the step size of each iteration (epoch) during model training. The step size is how fast (or slow) you update your neurons' weights in response to an estimated error. Model weights are updated using the backpropagation error method. So, the input will flow from the input nodes of your model through the neurons to the output nodes then the error is determined and backpropagated to update the neuron's (model) weights. How fast to update those neurons' weights is the learning rate.



- If the learning rate is **high**, thus the model weights are updated fast and frequently, then your model will converge fast, but it may overshoot the true error minima. **This means a faster but erroneous model.**
- If the learning rate is **low**, thus the model weights are updated slowly, then your model will take a long time to converge but will not overshoot the true error minima. **This means a slower but more accurate model.**

Q3. How to know whether your model is suffering from the problem of Exploding Gradients?

Answer: There are some subtle signs that you may be suffering from exploding gradients during the training of your network, such as:

- The model is unable to get traction on your training data (e g. poor loss).

- The model is *unstable*, resulting in large changes in loss from update to update.
- The model loss goes to NaN during training.

If you have these types of problems, you can dig deeper to see if you have a problem with exploding gradients. There are some less subtle signs that you can use to confirm that you have exploding gradients:

- The model weights quickly become very large during training.
- The model weights go to NaN values during training.
- The error gradient values are consistently above 1.0 for each node and layer during training.

Q4. What is a Confusion Matrix?

Answer: For every classification model prediction, a matrix called the ***confusion matrix*** can be constructed which demonstrates the number of test cases correctly and incorrectly classified.

It looks something like this (considering 1 -Positive and 0 -Negative are the target classes):

	Actual 0	Actual 1
Predicted 0	True Negatives (TN)	False Negatives (FN)
Predicted 1	False Positives (FP)	True Positives (TP)

- TN: Number of negative cases correctly classified
- TP: Number of positive cases correctly classified
- FN: Number of positive cases incorrectly classified as negative
- FP: Number of negative cases correctly classified as positive

Q5. What is model accuracy and how is it measured?

Answer: Accuracy is the simplest metric and can be defined as the number of test cases correctly classified divided by the total number of test cases.

$$Accuracy = (TP+TN)/(TP+TN+FP+FN)$$

It can be applied to most generic problems but is not very useful when it comes to unbalanced datasets.

For instance, if we are detecting frauds in bank data, the ratio of fraud to non-fraud cases can be 1:99. In such cases, if accuracy is used, the model will turn out to be 99% accurate by predicting all test cases as non-fraud. The 99% accurate model will be completely useless.

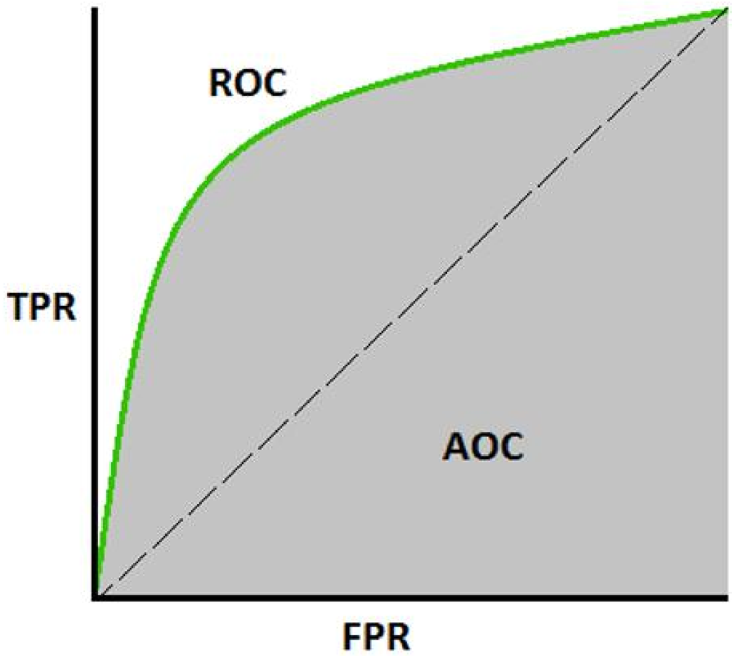
If a model is poorly trained such that it predicts all the 1000 (say) data points as non-frauds, it will be missing out on the 10 fraud data points. If accuracy is measured, it will show that that model correctly predicts 990 data points and thus, it will have an accuracy of (990/1000)*100 = 99%!

This is why accuracy is a false indicator of the model's health.

Therefore, for such a case, a metric is required that can focus on the ten fraud data points which were completely missed by the model.

Q6. Explain AUC-ROC Curve.

Answer: ROC curve is a plot of true positive rate (recall) against false positive rate (TN / (TN+FP)). AUC-ROC stands for Area Under the Receiver Operating Characteristics and the higher the area, the better is the model performance.
If the curve is somewhere near the 50% diagonal line, it suggests that the model randomly predicts the output variable.



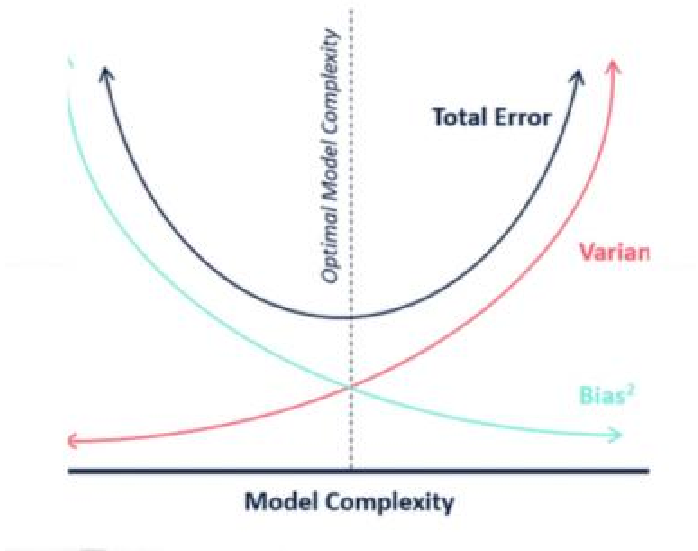
Q7. Explain the bias-variance trade-off.

Answer: Bias occurs when a model is strictly ruled by assumptions – like the linear regression model assumes that the relationship of the output variable with the independent variables is a straight line. This leads to underfitting when the actual values are non-linearly related to the independent variables.

Variance is high when a model focuses on the training set too much and learns the variations very closely, compromising on generalization. This leads to overfitting.

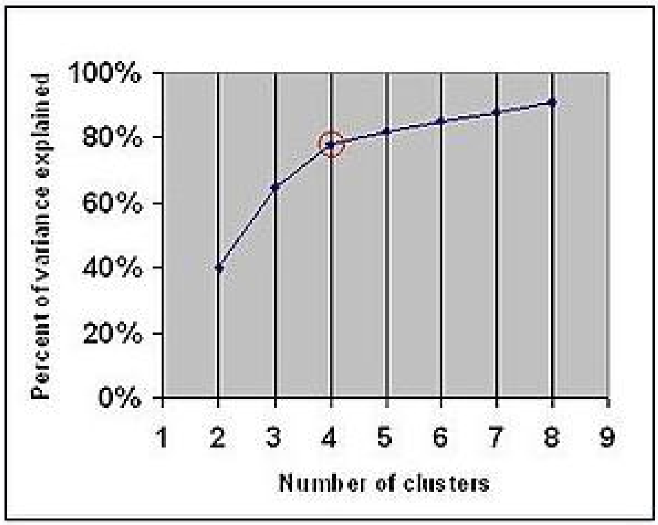
An optimal model is one that has the lowest bias and variance and since these two attributes are indirectly proportional, the only way to achieve this is through a tradeoff between the two. Therefore, the model selection should be such that the bias and variance intersect like in the image below.

This can be achieved by iteratively tuning the hyperparameters of the model in use (Hyperparameters are the input parameters that are fed to the model functions). After every iteration, the model evaluation must take place with the use of a suitable metric.



Q8. What is the elbow method?

Answer: The elbow method is used to determine the number of clusters in a dataset by plotting the number of clusters on the x-axis against the percentage of variance explained on the y-axis. The point in x-axis where the curve suddenly bends (the elbow) is considered to suggest the optimal number of clusters.



Explained variance. The "elbow" is indicated by the red circle. The number of clusters chosen should therefore be 4.

Q9. What is a F1-Score?

Answer: F1 score is the harmonic mean of Recall and Precision and therefore, balances out the strengths of each.

$$F1Score = 2 * ((precision * recall)/(precision + recall))$$

It is useful in cases where both recall and precision can be valuable – like in the identification of plane parts that might require repairing. Here, precision will be required to save on the company's cost (because plane parts are extremely expensive) and recall will be required to ensure that the machinery is stable and not a threat to human lives.

Q10. What is Overfitting in Machine Learning?

Answer: Overfitting refers to a model that models the training data too well.

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the model's ability to generalize.