BOSSCODER ACADEMY

Walmart

Interview Experience

For Data Scientist

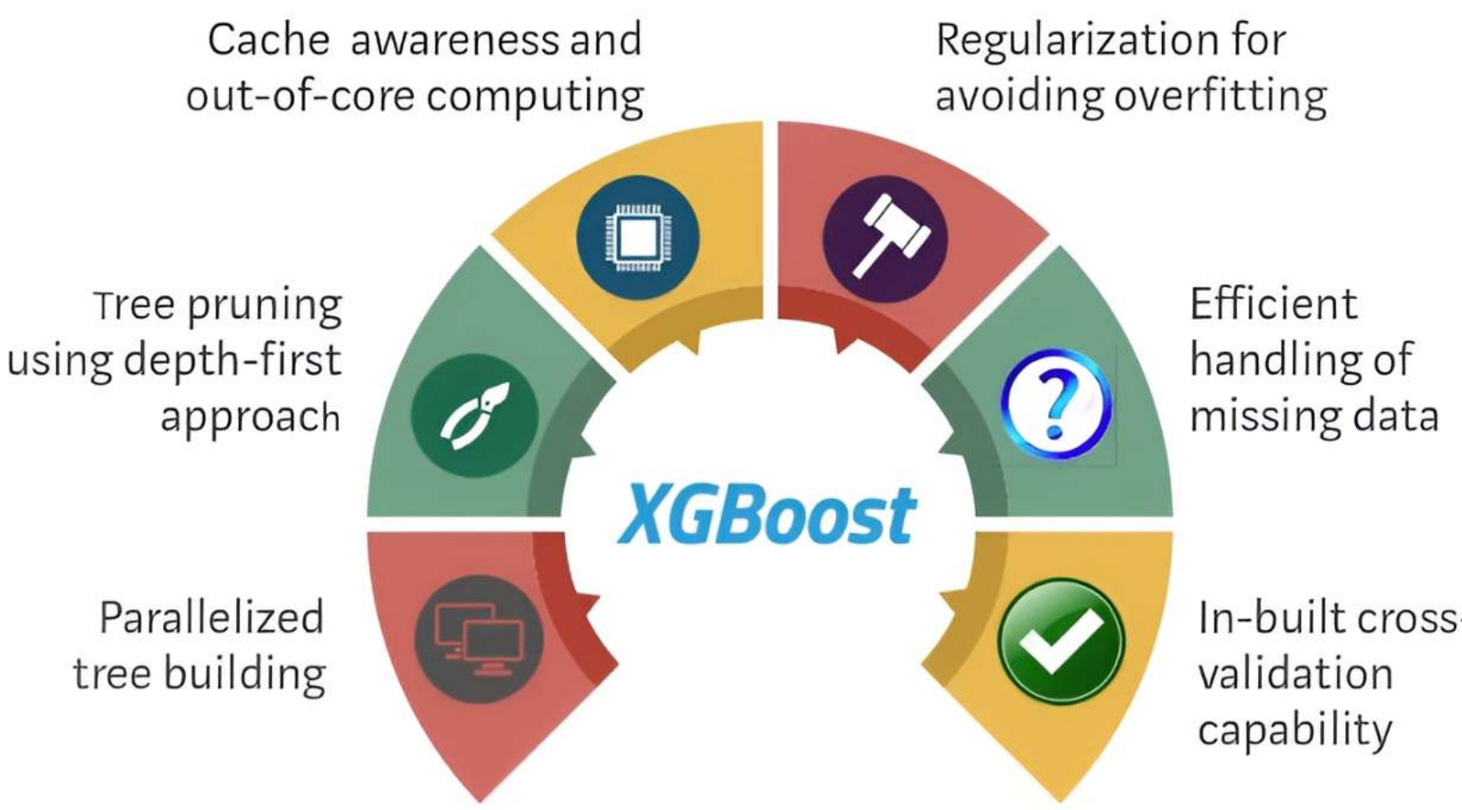
ROUND 1

Duration: **60 Min** Difficulty Level: **Medium**

TECHNICAL INTERVIEW

Question Asked:

1. How can you tune the hyper parameters of XGboost algorithm?
2. Explain the hyper parameters in XGboost Algorithm.



Question 1 - Problem Approach

XGBoost parameters are categorized into General, Booster, and Learning Task parameters. To tune the model effectively, follow these steps:

1. Set a relatively high learning rate (e.g., 0.1) and find the optimal number of trees using cross-validation with the 'cv' function.
2. Tune tree-specific parameters (max_depth, min_child_weight, gamma, subsample, colsample_bytree) for the chosen learning rate and number of trees.
3. Adjust regularization parameters (lambda, alpha) to reduce model complexity and improve performance.
4. Decrease the learning rate and fine-tune parameters for optimal results.

Question 2 - Problem Approach

Hyperparameters guide an algorithm's learning process, and XGBoost, known for its power, relies on tuning these parameters.

In XGBoost, there are four main categories of hyperparameters: general, booster, learning task, and command line parameters.

Before running an XGBoost model, it's essential to set general, booster, and task parameters. Command line parameters are specific to the console version of XGBoost.

Key hyperparameters include:

1. **eta/learning_rate:** Controls the shrinkage factor (default 0.3).
2. **n_rounds/n_estimators:** Specifies the number of boosting rounds.
3. **max_depth:** Determines the maximum depth of the tree (default 6).
4. **subsample and colsample_bytree:** Manage data and predictor sampling to prevent overfitting.
5. **lambda/reg_lambda and alpha/reg_alpha:** Control regularization to avoid overfitting.

Optimizing these hyperparameters enhances the performance of XGBoost.

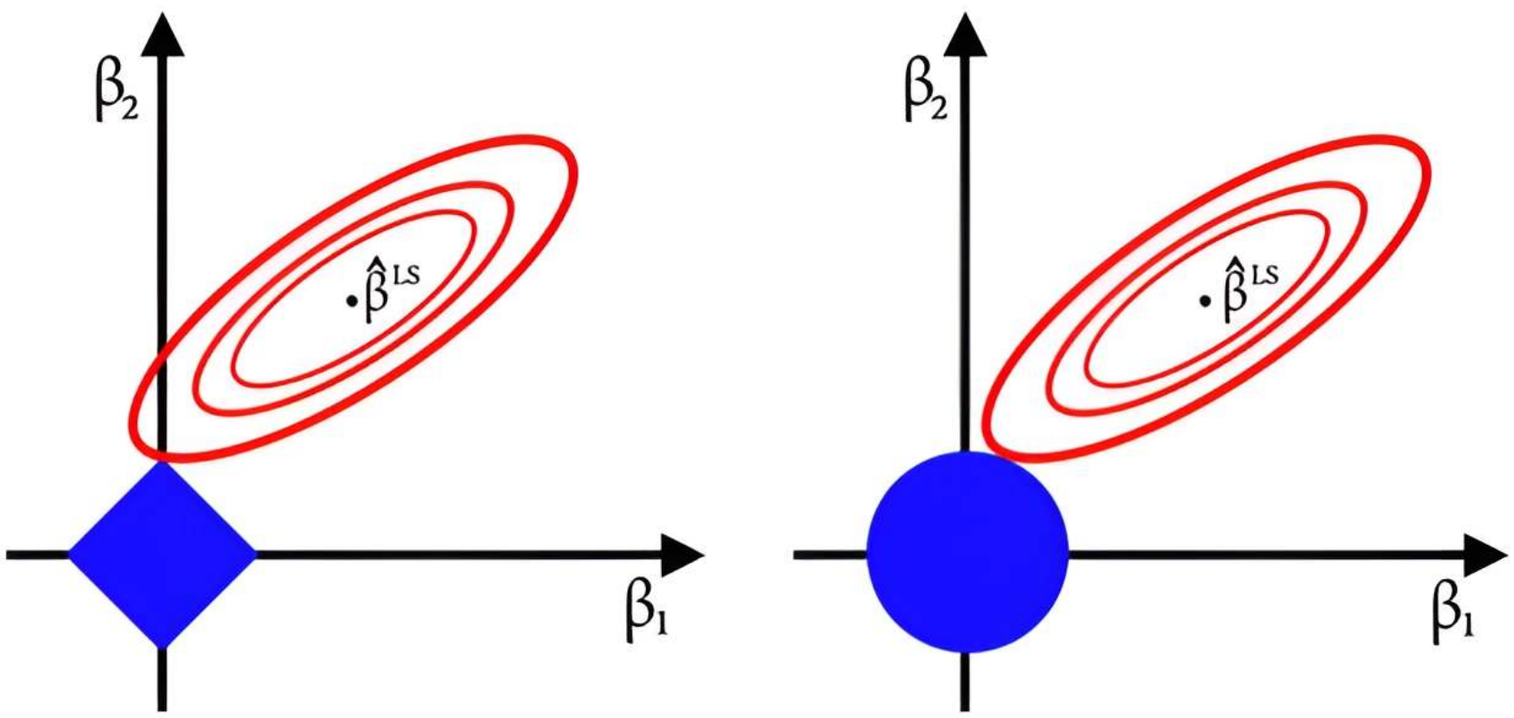
ROUND 2

Duration: **60 Min** Difficulty Level: **Medium**

TECHNICAL INTERVIEW

Question Asked:

- Difference between Ridge and LASSO .
- How to fit a time series model? State all the steps you would follow.



Lasso regression Ridge regression

Question 1 - Problem Approach

1. Ridge and Lasso regression uses two different penalty functions. Ridge uses L2 where as lasso go with L1. In ridge regression, the penalty is the sum of the squares of the coefficients and for the Lasso, it's the sum of the absolute values of the coefficients. It's a shrinkage towards zero using an absolute value (L1 penalty) rather than a sum of squares(L2 penalty).

2. As we know that ridge regression can't have zero coefficients. Here, we can either select all the coefficients or none of them whereas LASSO does both parameter shrinkage and variable selection automatically because it zero out the co-efficients of collinear variables. Here it helps to select the variable(s) out of given n variables while performing lasso regression.

Question 2 - Problem Approach

Fitting a time series forecasting model requires 5 steps . The steps are explained below :

1. **Data preparation:** Data preparation is usually the first step where we load all the essential packages and data into a time series object.

2. **Time series decomposition:** Decomposition basically means deconstructing and visualizing the series into its component parts.

3. **Modelling:** The actual model building is a simple 2-lines code using auto.arima() function. auto.arima will take care of the optimum parameter values, we just need to specify a few boolean parameters.

4. **Forecasting:** Making an actual forecast is the simplest of all the steps above . We are using forecast() function and passing the model above and specifying the number of time steps into the future we want to forecast.

5. **Model evaluation:** This is an extra step for model evaluation and accuracy tests.

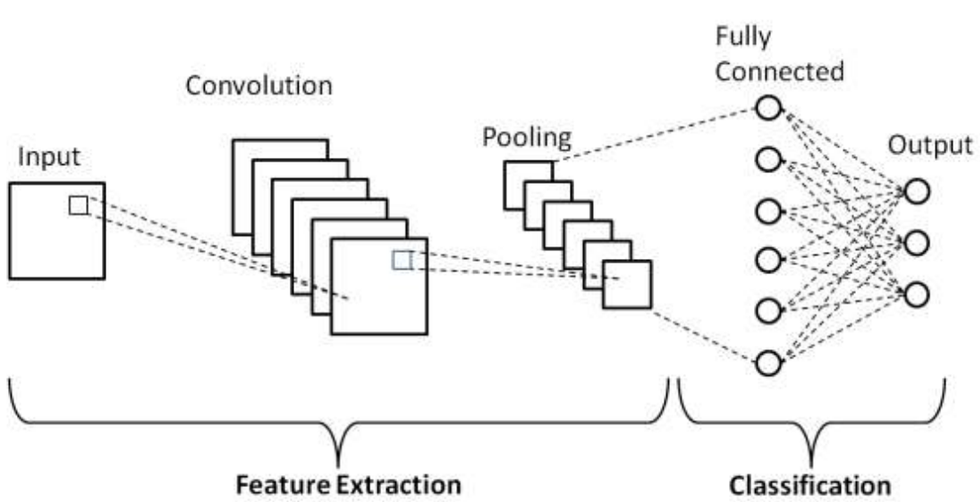
ROUND 3

Duration: **60 Min** Difficulty Level: **Hard**

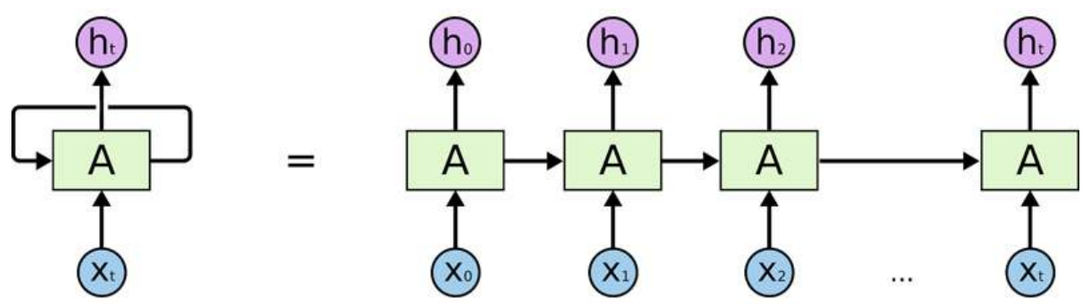
TECHNICAL INTERVIEW

Question Asked:

- RNN,CNN and difference between these two.
- What are outlier values and how do you treat them?



CNN



RNN

Question 1 - Problem Approach

- The main difference between CNN and RNN is the ability to process temporal information or data that comes in sequences, such as a sentence for example. Moreover, convolutional neural networks and recurrent neural networks are used for completely different purposes, and there are differences in the structures of the neural networks themselves to fit those different use cases.
- CNNs employ filters within convolutional layers to transform data. Whereas, RNNs reuse activation functions from other data points in the sequence to generate the next output in a series.

Question 2 - Problem Approach

Outliers are data points significantly different from others in a dataset. They can be identified using univariate or graphical analysis.

For handling multiple outliers, replacing them with the 99th or 1st percentile values is common. Treatment options include adjusting outliers to fit within a range or removing them entirely.