CatBoost Report

- *Donal Loitam*

INTRODUCTION

- CatBoost algorithm is another member of the gradient boosting technique on decision trees.
- It yields state-of-the-art results without extensive data training typically required by other machine learning methods, and
- One of the many unique features that the CatBoost algorithm offers is the integration to work with diverse data types to solve a wide range of data problems faced by numerous businesses
- Not just that, but CatBoost also offers accuracy just like the other algorithm in the tree family.
- The term CatBoost is an acronym that stands for "Category" and "Boosting."
- CatBoost supports numerical, categorical, and text features but has a good handling technique for categorical data.

Features of CatBoost :

Robust

CatBoost can improve the performance of the model while reducing overfitting and the time spent on tuning.

CatBoost has several parameters to tune. Still, it reduces the need for extensive hyper-parameter tuning because the default parameters produce a great result.

Accuracy

The CatBoost algorithm is a high performance and greedy novel gradient boosting implementation.

Hence, CatBoost (when implemented well) either leads or ties in competitions with standard benchmarks.

Categorical Features Support

The key features of CatBoost is one of the significant reasons why it was selected by many boosting algorithms such as LightGBM, XGBoost algorithm ..etc

With other machine learning algorithms. After preprocessing and cleaning your data, the data has to be converted into numerical features so that the machine can understand and make predictions.

This is same like, for any text related models we convert the text data into to numerical data it is know as word embedding techniques.

This process of encoding or conversion is time-consuming. CatBoost supports working with non-numeric factors, and this saves some time plus improves your training results.

The Algorithm as compared to XGBoost :

Before we dive into the several differences that these algorithms possess, it should be noted that the CatBoost algorithm does not require the conversion of the data set to any specific format. Precisely numerical format, unlike XGBoost.

The algorithms differ from one another in implementing the boosted trees algorithm and their technical compatibilities and limitations.

Split

The split function is a useful technique, and there are different ways of splitting features for these three machine learning algorithms.

One right way of splitting features during the processing phase is to inspect the characteristics of the column.

The CatBoost algorithm introduced a unique system called **Minimal Variance Sampling (MVS)**, which is a weighted sampling version of the widely used approach to regularization of boosting models, Stochastic Gradient Boosting.

Also, Minimal Variance Sampling (MVS) is the new default option for subsampling in CatBoost.

With this technique, the number of examples needed for each iteration of boosting decreases, and the quality of the model improves significantly compared to the other gradient boosting models.

The features for each boosting tree are sampled in a way that maximizes the accuracy of split scoring.

In contrast to it, XGBoost does not utilize any weighted sampling techniques. This is the reason why the splitting process is slower compared to MVS of CatBoost.

Leaf Growth

A significant change in the implementation of the gradient boosting algorithms such as XGBoost, LightGBM CatBoost, is the method of tree construction, also called leaf growth.

The CatBoost algorithm grows a balanced tree. In the tree structure, the feature-split pair is performed to choose a leaf.

The split with the smallest penalty is selected for all the level's nodes according to the penalty function. This method is repeated level by level until the leaves match the depth of the tree.

By default, CatBoost uses symmetric trees ten times faster and gives better quality than non-symmetric trees.

However, in some cases, other tree growing strategies (Lossguide, Depthwise) can provide better results than growing symmetric trees.

The leaf-wise approach is a good choice for large datasets, which is one reason why XGBoost performs well.

Categorical Features Handling

CatBoost uses one-hot encoding for handling categorical features. By default, CatBoost uses one-hot encoding for categorical features with a small number of different values in most modes.

The number of categories for one-hot encoding can be controlled by the one_hot_max_size parameter in Python and R.

On the other hand, the CatBoost algorithm categorical encoding is known to make the model slower.

XGBoost was not engineered to handle categorical features. The algorithm supports only numerical features.