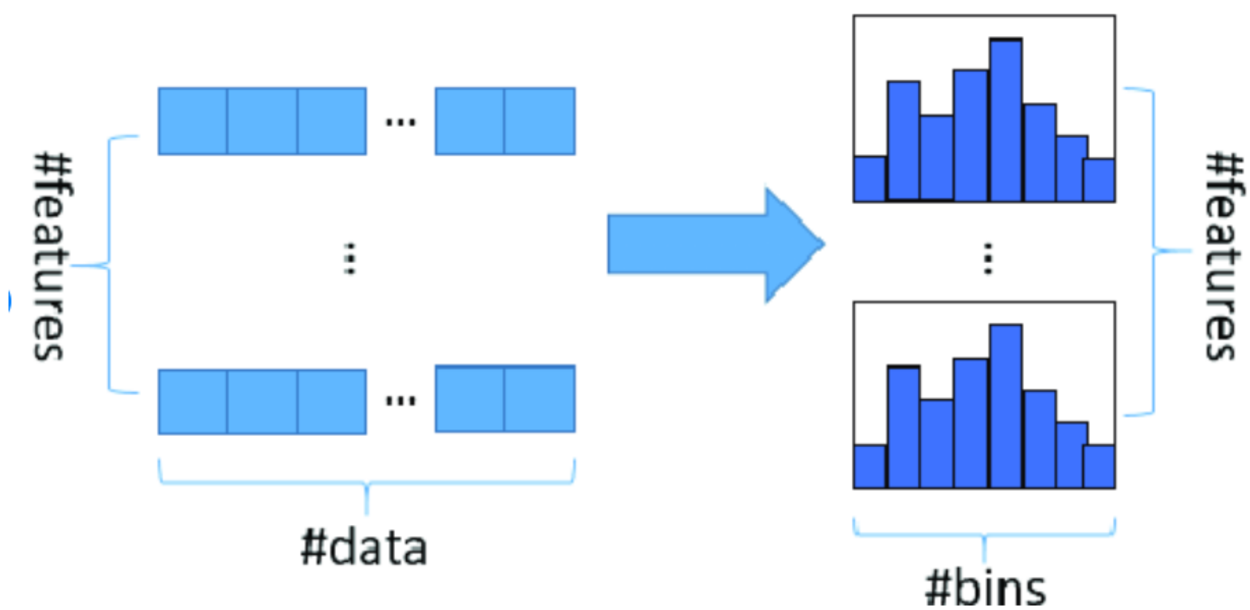
LightGBM

Aaryan
CO21BTECH11001

LightGBM is a gradient boosting technique which aims to reduce the time complexity of other gradient boosting algorithms. It does so by using Histogram based split finding, GOSS and EFB. These are explained below:

Histogram based splitting:

Histogram based algorithm buckets continuous features into discrete bins to construct feature histograms during training.



It costs $O(\#data\ *\ \#feature)$ for histogram building and $O(\#bin\ *\ \#feature)$ for split point finding. As $bin\ <<\ data$ histogram building will dominate the computational complexity.
This method is very efficient as compared to the one used by GBM.

GOSS:

GOSS (Gradient Based One Side Sampling) is a sampling method which down samples the instances on basis of gradients. As we know instances with small gradients are well trained (small training error) and those with large gradients are under trained. Since discarding instances with small gradients and only focusing on instances with large gradient alters the data distribution, GOSS retains instances with large gradients while performing random sampling on instances with small gradients.

Inputs:

a: Sampling ratio of large gradient data.
b: Sampling ratio of small gradient data.
Steps are as follows:

1. Sort the instances according to absolute gradients in a descending order.
2. Select the top $a \times 100\%$ instances.
3. Randomly sample $b \times 100\%$ instances from rest of the data.
4. Amplify the sampled data with small gradients by a constant $\frac{1-a}{b}$, when calculating the information gain.

By doing so, we put more focus on the under-trained instances without changing the original data distribution by much.

EFB (Exclusive Feature Bundling):

In a sparse feature space, many features are mutually exclusive, i.e., they never take nonzero values simultaneously. We can bundle exclusive features into a single feature (which we call a feature bundle). By a feature scanning algorithm, we can build the same feature histograms from the feature bundles as those from individual features. In this way, the complexity of histogram building changes from

$O(\#data \times \#feature)$ to $O(\#data \times \#bundle)$, while $\#bundle << \#feature$.

Algorithm for bundling of features:

Input:

F: features, K: max conflict count

1. Construct a graph with weighted (measure of conflict between features) edges. Conflict is measure of the fraction of exclusive features which have overlapping nonzero values.
2. Sort the features by count of non-zero instances in descending order.
3. Loop over the ordered list of features and assign the feature to an existing bundle (if conflict $<$ threshold) or create a new bundle (if conflict $>$ threshold).

Algorithm to merge exclusive features:

Input:

$numData$: number of data
F: One bundle of exclusive features

1. Calculate the offset to be added to every feature in feature bundle.
2. Iterate over every data instance and feature.
3. Initialize the new bucket as zero for instances where all features are zero.
4. Calculate the new bucket for every nonzero instance of a feature by adding respective offset to original bucket of that feature.

<u>**Questions:**</u>

1. LightGBM is
   (a) Gradient Boosting Algorithm
   (b) Bagging Algorithm
   (c) Supervised learning algorithm

(d) Unsupervised learning algorithm
Ans. (a), (c)

2. LightGBM prevents the alteration of data distribution by:
   (a) Ignoring instances with low value of gradient
   (b) Scales them with a constant
   (c) Random sampling without amplification
   (d) Random sampling with amplification
   Ans. (b), (d)

3. What is the major assumption of LightGBM?
   Ans. It assumes that the best next model minimizes the total prediction error when merged with past models. The central concept is to define the desired outcomes for this next model to reduce error.

4. What are the advantages of LightGBM?
   Ans. It is fast and efficient. Its accuracy is better than any other boosting algorithm. Also, it is less computationally and space expensive.

5. What are the disadvantages of LightGBM?
   Ans. With small datasets, it is prone to overfitting.