Adaboost,LightGBM,extremeGBM: Hypertuning parameter’s

pip is the package installer for Python packages. It is used to install packages(consist of classes and function) which do not come with python.
Pip – pip install packages.

Boosting Algorithm as follows:
Boosting Algorithm helps machine learning models to improve their Prediction accuracy and performance by converting multiple weak learner to a single robust learning model. Machine learning models can be weak or strong learners.

XGBoosting:
Root node: The topmost node in the tree, where the decision process begins.
Decision node: A node that splits based on a feature and leads to multiple branches based on the decision.
Terminal node(leaf node) : A node at the end of a branch, representing the final prediction or outcome.
max_depth means maximum depth of the tree from top to bottom
min_samples_split minimum number of samples a node must contain to perform splitting,Higher values make the tree simpler, prevents over fitting. Default is 2, but it's often tuned for better performance.
n_estimator – total number of decision trees/models in the algorithm.

S.no	Hypertuning parameter	R2 value
1	GradientBoostingRegressor(n_estimators=500, max_depth=4,min_samples_split=5,learning_rate=0.01,loss="squared_error")	.8821
2	GradientBoostingRegressor(n_estimators=400, max_depth=3,min_samples_split=4,learning_rate=0.01,loss="squared_error")	.8915
3	GradientBoostingRegressor(n_estimators=300, max_depth=3,min_samples_split=3,learning_rate=0.01,loss="squared_error")	.8907
4	GradientBoostingRegressor(n_estimators=500, max_depth=4,min_samples_split=5,learning_rate=0.01,loss="absolute_error")	.8337
5	GradientBoostingRegressor(n_estimators=400, max_depth=3,min_samples_split=4,learning_rate=0.01,loss="absolute_error")	.7693
6	GradientBoostingRegressor(n_estimators=300, max_depth=3,min_samples_split=3,learning_rate=0.01,loss="absolute_error")	.7463
7	GradientBoostingRegressor(n_estimators=500, max_depth=4,min_samples_split=5,learning_rate=0.01,loss="huber")	.8929
8	GradientBoostingRegressor(n_estimators=400, max_depth=3,min_samples_split=4,learning_rate=0.01,loss="huber")	.8923
9	GradientBoostingRegressor(n_estimators=300, max_depth=3,min_samples_split=3,learning_rate=0.01,loss="huber")	.8909
10	GradientBoostingRegressor(n_estimators=500, max_depth=4,min_samples_split=5,learning_rate=0.01,loss="quantile")	.5274
11	GradientBoostingRegressor(n_estimators=400, max_depth=3,min_samples_split=4,learning_rate=0.01,loss="quantile")	.5779
12	GradientBoostingRegressor(n_estimators=300, max_depth=3,min_samples_split=3,learning_rate=0.01,loss="quantile")	.5368

LGBM:

What is Gradient ? General meaning of Gradient – means measure of steepness or slope of road or line.

The slope of a function, or the rate of change in the output value relative to the input value.
Learning rate?
A hyperparameter that controls how much the algorithm moves in the direction of the gradient

A histogram of the distribution is used by LightGBM to bucket data into bins. Instead of using every data point, the bins are used to iterate, calculate the gain, and divide the data

A sparse dataset is a dataset with large number of missing values or consist of zero values. Most machine learning algorithms are developed for dense features, so extra techniques may be needed to handle sparse features.

LightGBM Hypertuning parameter's:

S.NO	**Hypertuning parameter's**	**R2 score**
1	LGBMRegressor(objective = "regression", boosting_type = "gbdt",num_leaves = 31,max_depth = 3,n_estimators=110,learning_rate = 0.05,feature_fraction = 0.9)	.8967
2	LGBMRegressor(objective = "regression", boosting_type = "gbdt",num_leaves = 50,max_depth = 3,n_estimators=110,learning_rate = 0.05,feature_fraction = 0.9)	.8967
3	LGBMRegressor(objective = "regression", boosting_type = "dart",num_leaves = 31,max_depth = 3,n_estimators=110,learning_rate = 0.05,feature_fraction = 0.9)	.7716
4	LGBMRegressor(objective = "regression", boosting_type = "dart",num_leaves = 50,max_depth = 3,n_estimators=110,learning_rate = 0.05,feature_fraction = 0.9)	.7716

Adaboost Hypertuning parameter's:
AdaBoost regression is often combined with decision tree regression because decision trees are considered "weak learners" that are easily adaptable to the boosting process of AdaBoost, allowing it to conditionally improve the model's accuracy.

S.NO	**Hypertuning parameter's**	R2score
1	AdaBoostRegressor(n_estimators=50,loss="linear",learning_rate=1.0, random_state=0)	.8549
2	AdaBoostRegressor(n_estimators=50,loss="linear",learning_rate=2.0, random_state=0)	.8560
3	AdaBoostRegressor(n_estimators=50,loss="linear",learning_rate=4.0, random_state=0)	-0.0360
4	AdaBoostRegressor(n_estimators=50,loss="square",learning_rate=1.0, random_state=0)	.5467
5	AdaBoostRegressor(n_estimators=50,loss="square",learning_rate=2.0, random_state=0)	.5295
6	AdaBoostRegressor(n_estimators=50,loss="exponential",learning_rate=1.0, random_state=0)	.6295
7	AdaBoostRegressor(n_estimators=50,loss="exponential",learning_rate=2.0, random_state=0)	.5466

Conclusion:
Best Model found in LighGradientBoosting Method : .8967