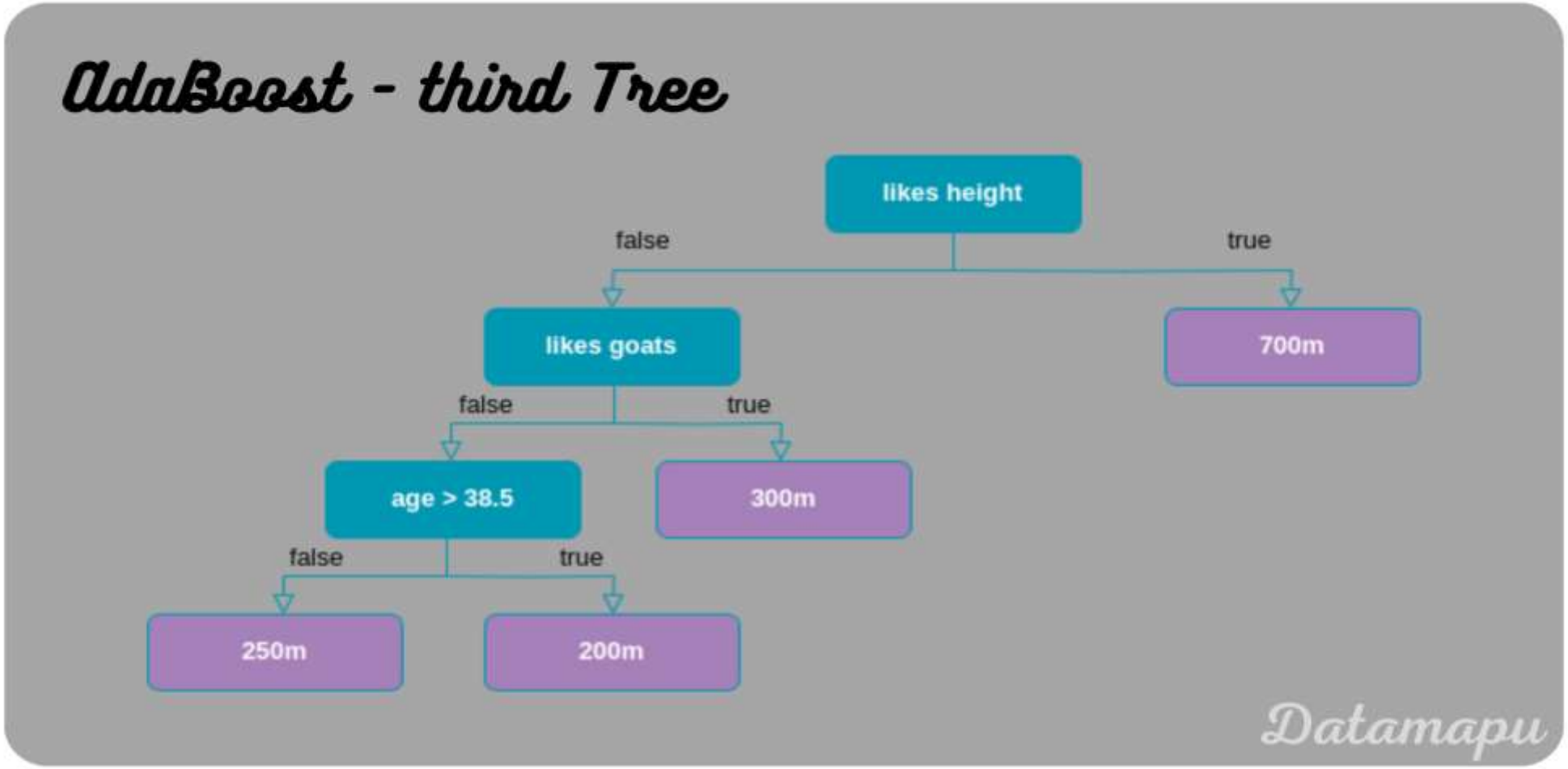
Boosting Algorithm Assignment

Ada Boosting

An AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases.

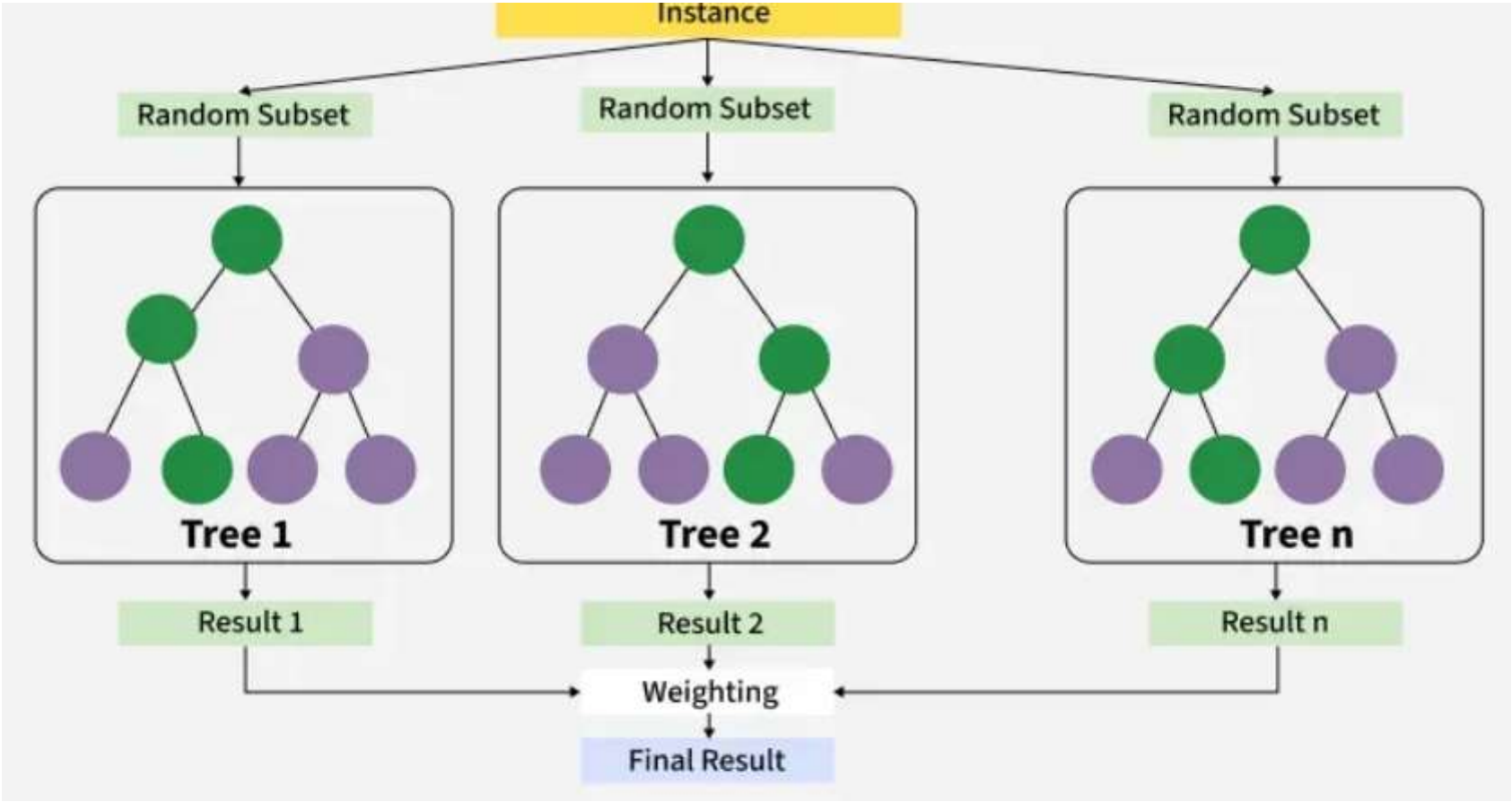


Ada Boosting

The above image shows how Boosting works. It starts with training on the original data. After each round more weight is given to misclassified points so the next model focuses on them. This process repeats and in the end all models are combined to make a final, more accurate prediction.

XG Boost

Traditional machine learning models like decision trees and random forests are easy to interpret but often struggle with accuracy on complex datasets. XGBoost short form for eXtreme Gradient Boosting is an advanced machine learning algorithm designed for efficiency, speed and high performance.



XG Boost

It is an optimized implementation of Gradient Boosting and is a type of ensemble learning method that combines multiple weak models to form a stronger model.

XGBoost uses decision trees as its base learners and combines them sequentially to improve the model's performance. Each new tree is trained to correct the errors made by the previous tree and this process is called boosting.

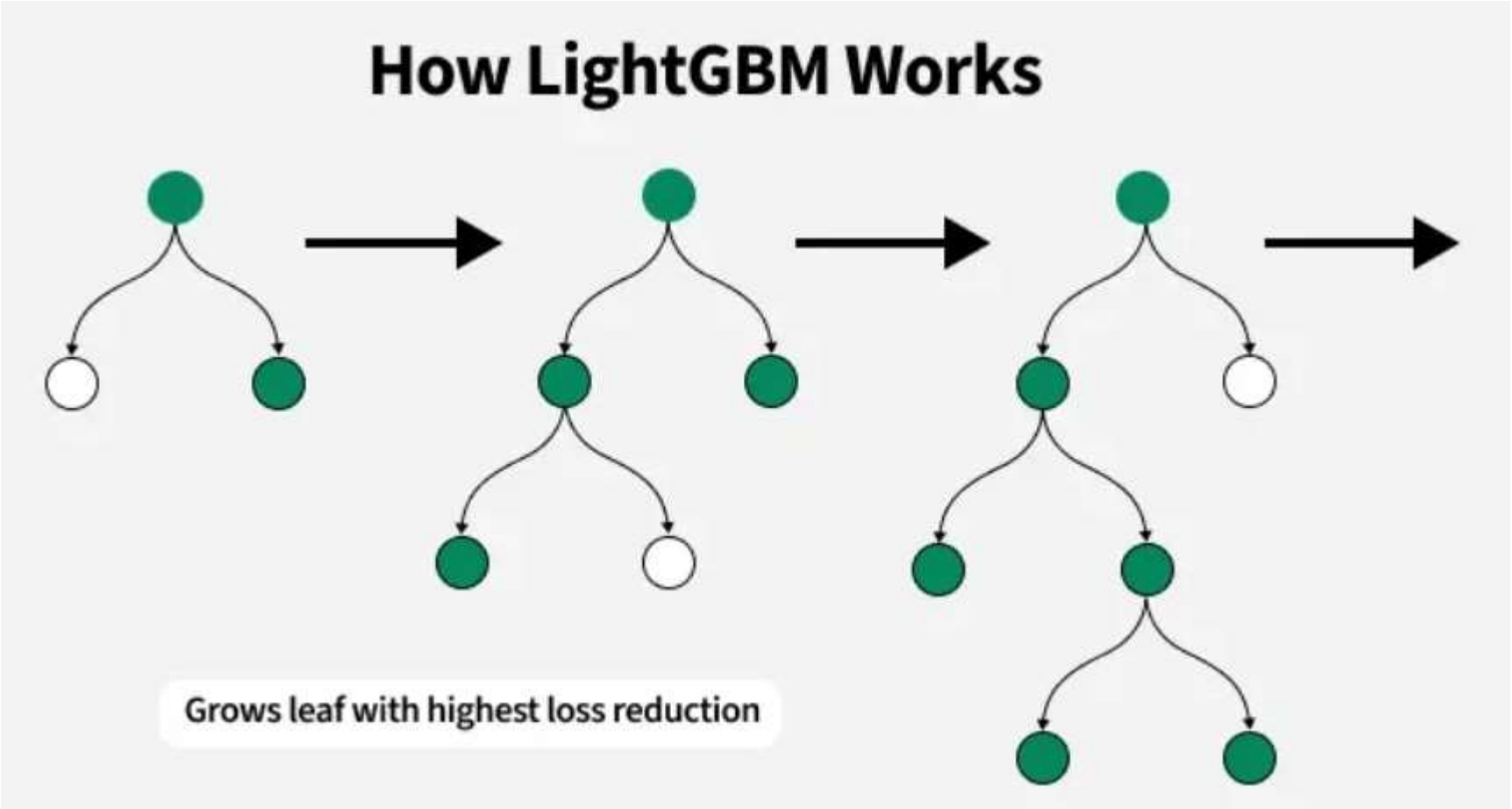
It has built-in parallel processing to train models on large datasets quickly. XGBoost also supports customizations allowing users to adjust model parameters to optimize performance based on the specific problem.

LG Boost - Light Gradient Boosting Machine

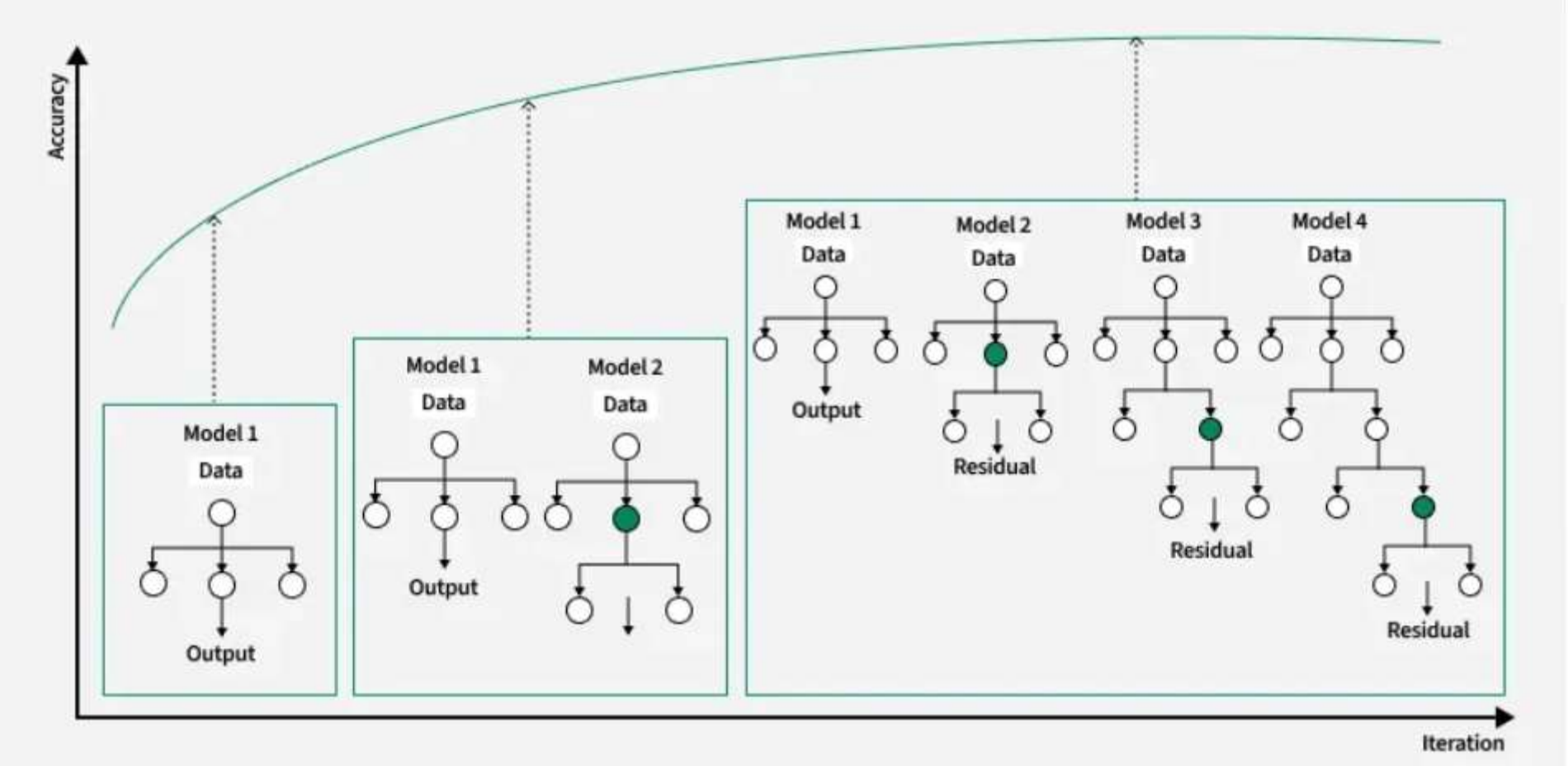
LightGBM is an open-source high-performance framework developed by Microsoft. It is an ensemble learning framework that uses gradient boosting method which constructs a strong learner by sequentially adding weak learners in a gradient descent manner.

It's designed for efficiency, scalability and high accuracy particularly with large datasets. It uses decision trees that grow efficiently by minimizing memory usage and optimizing training time. Key innovations like Gradient-based One-Side Sampling (GOSS), histogram-based algorithms and leaf-wise tree growth enable LightGBM to outperform other frameworks in both speed and accuracy.

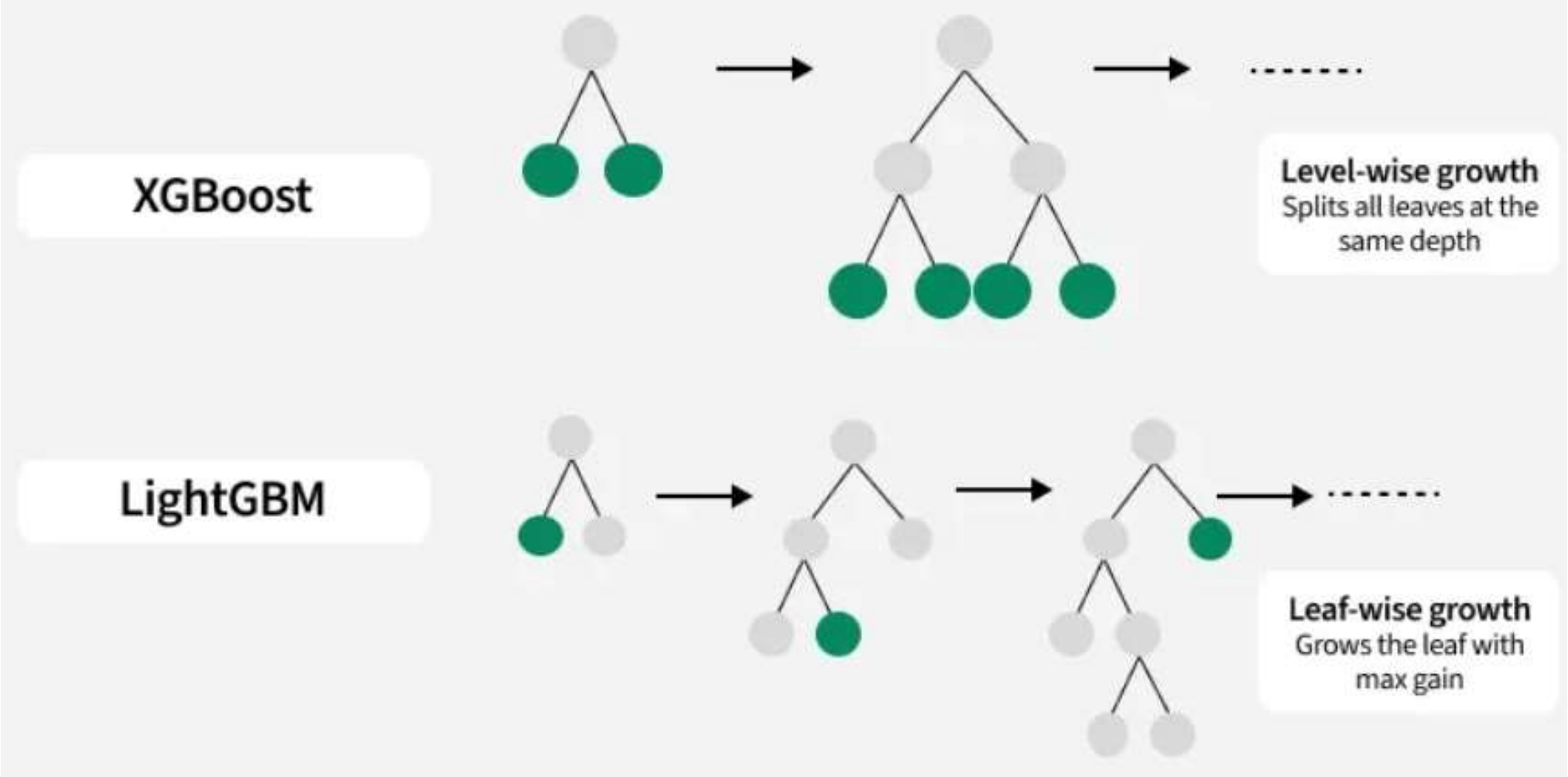
LG Boost - Light Gradient Boosting Machine



LG Boost - Light Gradient Boosting Machine



LG Boost - Light Gradient Boosting Machine



Ada Boost – model creation steps

R2 score = 92%

[13]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(independent, dependent, test_size=0.30, random_state=0)
```

[26]:
```
from sklearn.ensemble import AdaBoostRegressor
regressor = AdaBoostRegressor(n_estimators=100, random_state=0)
regressor.fit(X_train, y_train)
```

```
D:\ProgramData\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1339: DataConversionWarning: A column-vector y was passed when a 1d array was expe
cted. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

[26]:
```
AdaBoostRegressor
AdaBoostRegressor(n_estimators=100, random_state=0)
```

[28]:
```
y_predict = regressor.predict(X_test)
```

[30]:
```
from sklearn.metrics import r2_score
r_score = r2_score(y_test, y_predict)
r_score
```

[30]:
```
0.9268799485154495
```

XG Boost – model creation steps

R2 score = 92%

[21]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(independent, dependent, test_size=0.30, random_state=0)
```

[23]:
```
regressor = xg.XGBRegressor(objective ='reg:linear', n_estimators = 10, seed = 123)
regressor.fit(X_train, y_train)
```

```
C:\Users\GracyMano\AppData\Roaming\Python\Python312\site-packages\xgboost\training.py:183: UserWarning: [13:55:17] WARNING: C:\actions-runner\_work\xgboo
st\xgboost\src\objective\regression_obj.cu:245: reg:linear is now deprecated in favor of reg:squarederror.
  bst.update(dtrain, iteration=i, fobj=obj)
```

[23]:
```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             feature_weights=None, gamma=None, grow_policy=None,
             importance_type=None, interaction_constraints=None,
             learning_rate=None, max_bin=None, max_cat_threshold=None,
             max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
             max_leaves=None, min_child_weight=None, missing=nan,
             monotone_constraints=None, multi_strategy=None, n_estimators=10,
             n_jobs=None, num_parallel_tree=None, ...)
```

[25]:
```
y_predict = regressor.predict(X_test)
```

[27]:
```
from sklearn.metrics import r2_score
r_score = r2_score(y_test, y_predict)
r_score
```

[27]:
```
0.9276081919670105
```

Light Gradient Boosting Machine– model creation steps

R2 score = -4%

```
[37]: import lightgbm as lgbm
```

```
[49]: regressor = lgbm.LGBMRegressor(objective='regression', metric='rmse', boosting_type='gbdt')
      regressor.fit(X_train, y_train)
```

```
[LightGBM] [Warning] There are no meaningful features which satisfy the provided configuration. Decreasing Dataset parameters min_data_in_bin or min_da
ta_in_leaf and re-constructing Dataset might resolve this warning.
[LightGBM] [Info] Total Bins 0
[LightGBM] [Info] Number of data points in the train set: 35, number of used features: 0
[LightGBM] [Info] Start training from score 110225.320145
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
[LightGBM] [Warning] Stopped training because there are no more leaves that meet the split requirements
```

```
[51]: y_predict = regressor.predict(X_test)
```

```
[53]: from sklearn.metrics import r2_score
      r_score = r2_score(y_test, y_predict)
      r_score
```

```
[53]: -0.03677248112167808
```