XGBoost is a tree-based model with a different implementation compared to other tree-based models. The ensemble it creates is comprised of weak learners, meaning that each tree in the ensemble can barely make accurate predictions. But with enough of these models, it actually creates an ensemble that usually outperforms random forests. A benefit to XGBoost is that it provides a way to highly optimize the models. This is done by offering a large number of hyperparameters to tune. The downside to all these optimizations, and with ensemble models in general, is that they are hard to interpret.

## Code Example of XGBoost

```
import pandas as pd
import xgboost as xgb

df = pd.DataFrame(
    [[1, 2, 0], [3, 4, 1], [5, 6, 0], [7, 8, 1]],
    columns=["num", "amount", "target"]
)
df_xgb = xgb.DMatrix(
    df[["num", "amount"]], label=df["target"]
)
params = {"eval_metric": "logloss", "objective": "binary:hinge"}
bst = xgb.train(params, df_xgb)

bst.predict(df_xgb)

# output
array([0., 1., 0., 1.], dtype=float32)
```

## Additional Resources

- If you want to learn more about XGBoost, we recommend reading their **documentation**.
- To learn more about the research that was involved, check out the **research paper** that first published the algorithm.