

Nội dung



1. Giới thiệu Boosting
2. AdaBoost
3. Thuật toán AdaBoost
4. Xây dựng AdaBoost
5. XGBoost



XGBoost



- ❑ XGBoost là một trong những thuật toán machine learning rất phổ biến hiện nay. Có thể thực hiện cả 2 công việc là regression hay classification.
- ❑ XGBoost là một loại Boosting cung cấp giải pháp tốt trong việc phối hợp với thuật toán cơ sở (mặc định là classification and regression trees CART)



XGBoost



- ❑ **XGBoost (Extreme Gradient Boosting)** thuộc về nhóm các thuật toán tăng cường và sử dụng gradient boosting (GBM) framework làm cốt lõi. Nó là một thư viện tăng cường phân phối tối ưu hóa (optimized distributed gradient boosting library)
- ❑ **Cài đặt: pip install xgboost**



XGBoost



- ❑ **XGBoost khá phổ biến vì:**
 - Tốc độ và hiệu suất: Ban đầu được viết bằng C++, nó tương đối nhanh hơn các phân loại tập hợp khác.
 - Core algorithm song song: vì core XGBoost algorithm song song nên nó có thể khai thác sức mạnh của máy tính đa lõi (multi-core computers). Nó cũng có thể thực hiện song song trên GPU và trên các mạng máy tính khiến việc huấn luyện trên các bộ dữ liệu rất lớn cũng khả thi.



XGBoost



- Vượt trội so với các algorithm method khác: cho thấy hiệu suất tốt hơn trên nhiều bộ dữ liệu.
- Nhiều tham số điều chỉnh: bên trong XGBoost có các tham số cho cross-validation, regularization, user-defined objective functions, missing values, tree parameters...



XGBoost



□ Áp dụng: XGBClassifier

```
import xgboost as xgb

xgb_model = xgb.XGBClassifier(random_state=42)
xgb_model.fit(X, y)

y_pred = xgb_model.predict(X)

scores2 = cross_val_score(xgb_model, X, y, cv=5)
scores2
array([0.96666667, 0.96666667, 0.93333333, 0.9        , 1.        ])
```



XGBoost



□ Áp dụng: XGBRegressor

```
import xgboost as xgb
from sklearn.metrics import mean_squared_error
```

```
xgb_model = xgb.XGBRegressor(random_state=42)
xgb_model.fit(X, y)
```

```
y_pred = xgb_model.predict(X)
print("Model r^2 score:", xgb_model.score(X,y))
```

Model r^2 score: 0.9917113219896987

```
mse=mean_squared_error(y, y_pred)
print(mse)
```

0.025632114175041725

