Introducing XGBoost

EXTREME GRADIENT BOOSTING WITH XGBOOST

What is XGBoost?

- Optimized gradient-boosting machine learning library
- Originally written in C++
- Has APIs in several languages:
  - **Python**
  - R
  - Scala
  - Julia
  - Java

What makes XGBoost so popular?

- Speed and performance
- Core algorithm is parallelizable
- Consistently outperforms single-algorithm methods
- State-of-the-art performance in many ML tasks

Using XGBoost: a quick example

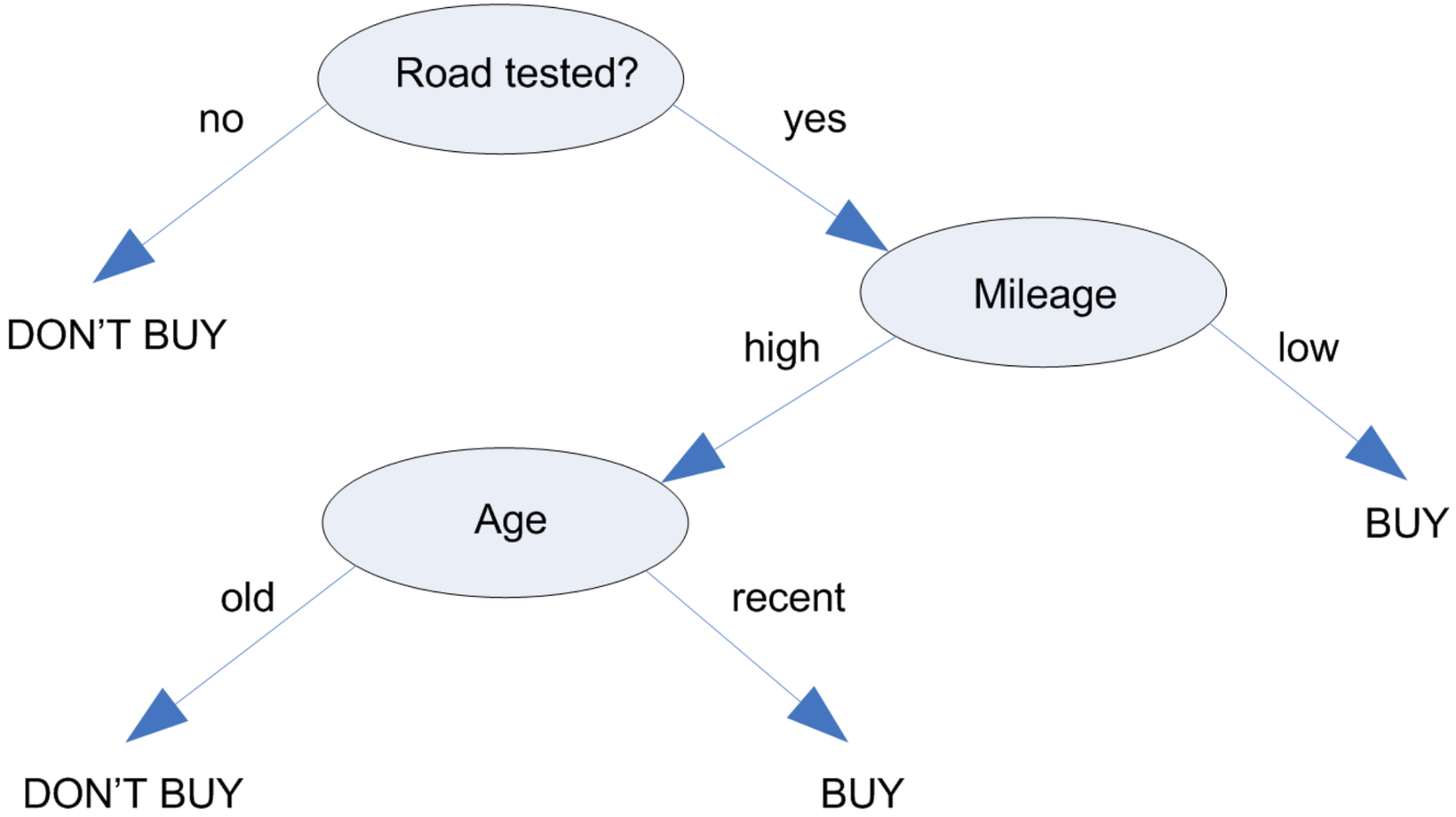
```
import xgboost as xgb
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
class_data = pd.read_csv("classification_data.csv")

X, y = class_data.iloc[:,:-1], class_data.iloc[:,-1]
X_train, X_test, y_train, y_test= train_test_split(X, y,
        test_size=0.2, random_state=123)
xg_cl = xgb.XGBClassifier(objective='binary:logistic',
        n_estimators=10, seed=123)
xg_cl.fit(X_train, y_train)

preds = xg_cl.predict(X_test)
accuracy = float(np.sum(preds==y_test))/y_test.shape[0]

print("accuracy: %f" % (accuracy))
```

```
accuracy: 0.78333
```

What is a decision tree?

EXTREME GRADIENT BOOSTING WITH XGBOOST

Visualizing a decision tree



[1] https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.modeler.help/nodes_treebuilding.htm

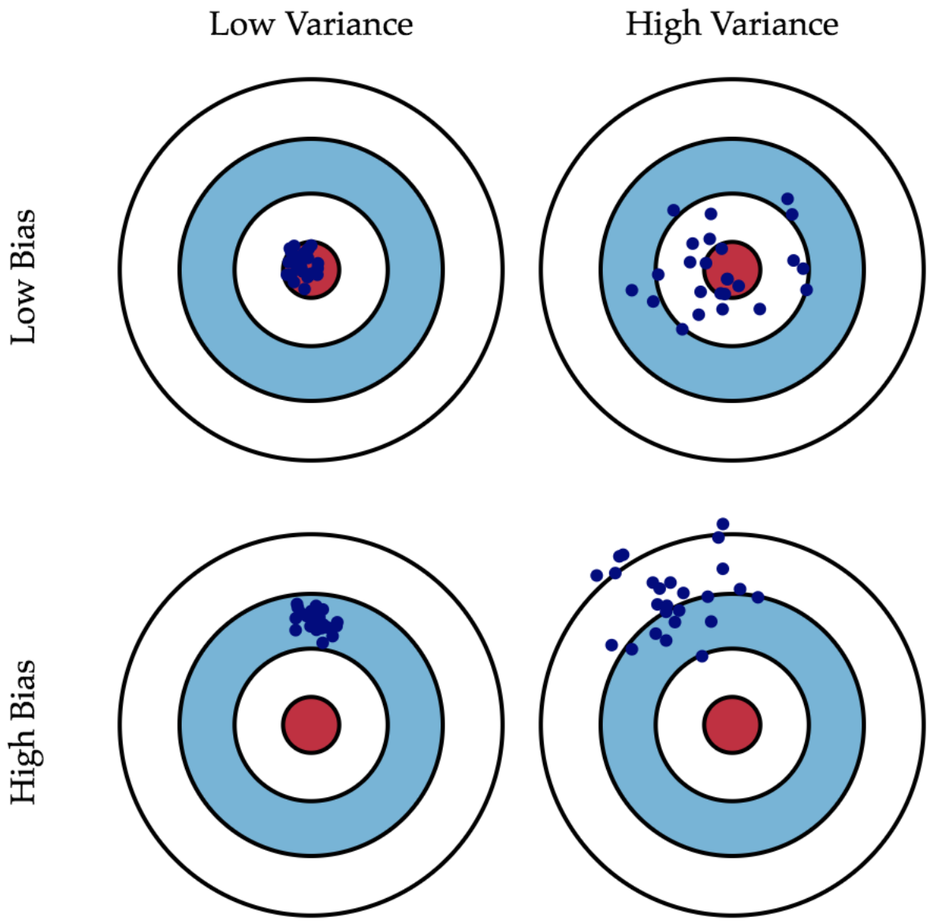
Decision trees as base learners

- Base learner - Individual learning algorithm in an ensemble algorithm
- Composed of a series of binary questions
- Predictions happen at the "leaves" of the tree

Decision trees and CART

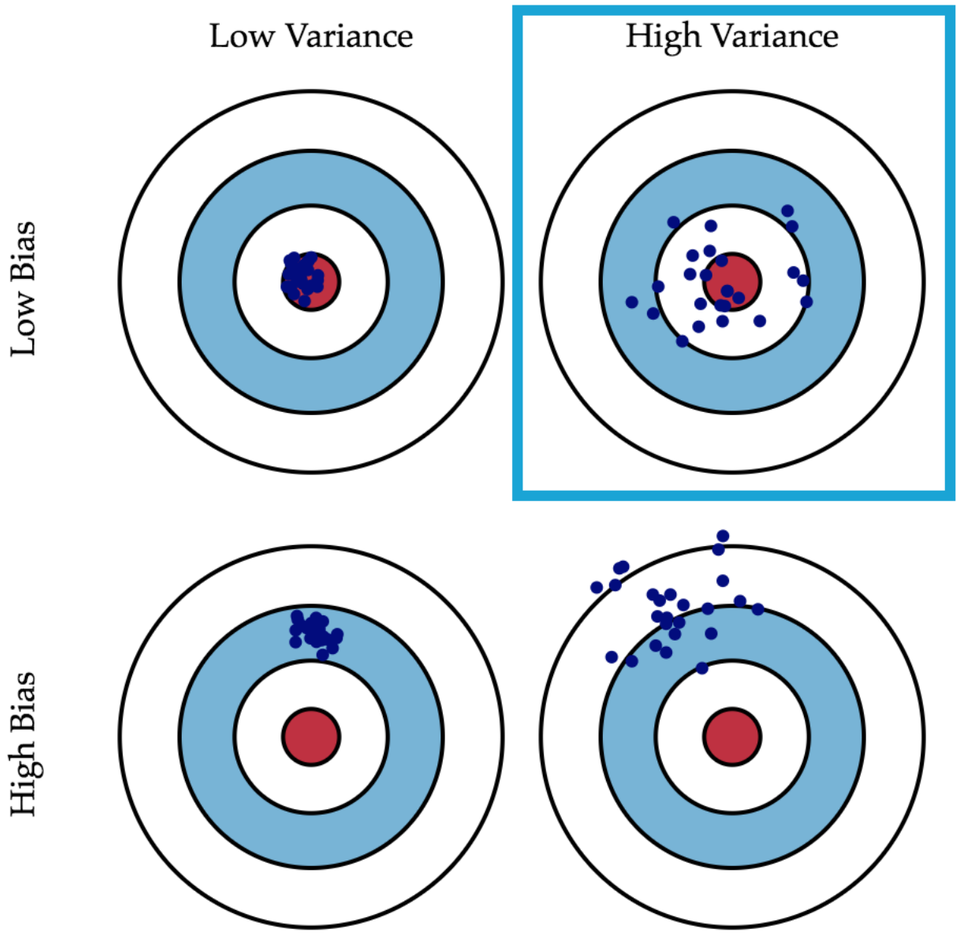
- Constructed iteratively (one decision at a time)
  - Until a stopping criterion is met

Individual decision trees tend to overfit



[1] http://scott.fortmann [2] roe.com/docs/BiasVariance.html

Individual decision trees tend to overfit



[1] http://scott.fortmann [2] roe.com/docs/BiasVariance.html

CART: Classification and Regression Trees

- Each leaf **always** contains a real-valued score
- Can later be converted into categories

What is Boosting?

EXTREME GRADIENT BOOSTING WITH XGBOOST

Boosting overview

- Not a specific machine learning algorithm
- Concept that can be applied to a set of machine learning models
  - "Meta-algorithm"
- Ensemble meta-algorithm used to convert many weak learners into a strong learner

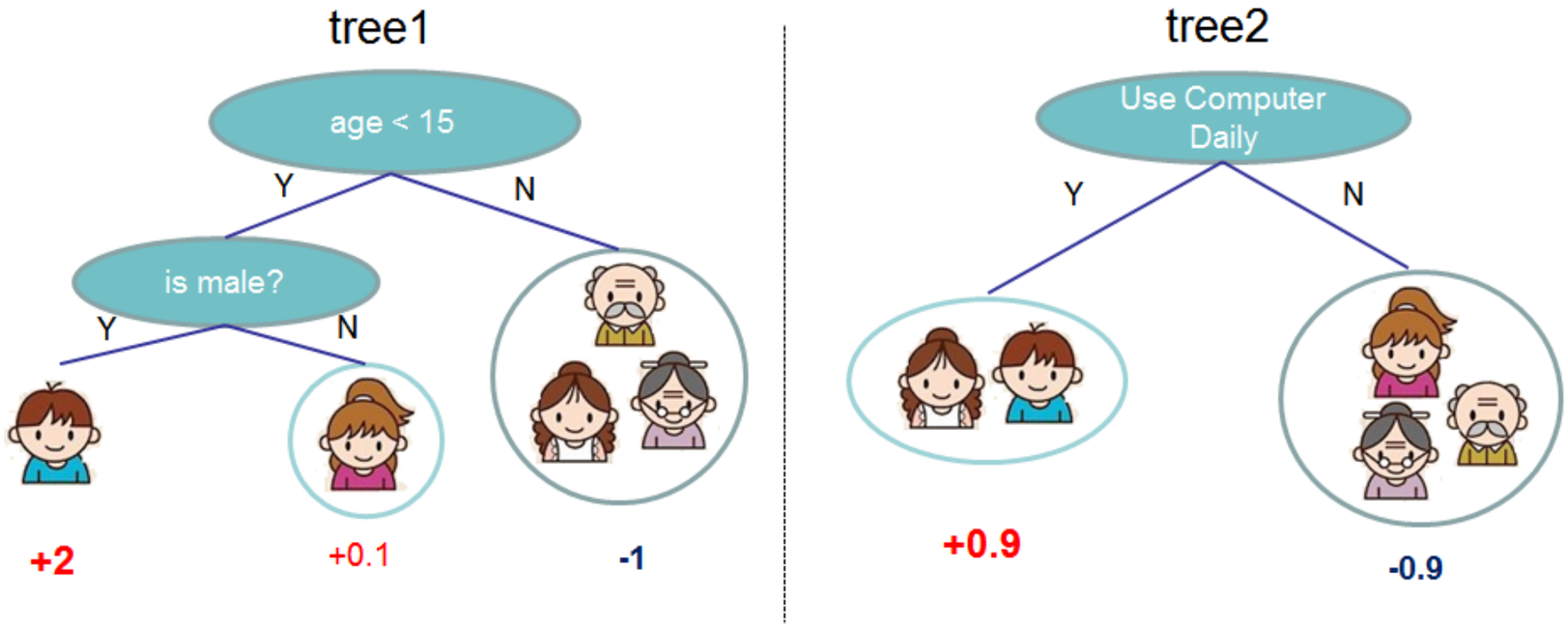
Weak learners and strong learners

- Weak learner: ML algorithm that is slightly better than chance
  - Example: Decision tree whose predictions are slightly better than 50%
- Boosting converts a collection of weak learners into a strong learner
- Strong learner: Any algorithm that can be tuned to achieve good performance

How boosting is accomplished

- Iteratively learning a set of weak models on subsets of the data
- Weighing each weak prediction according to each weak learner's performance
- Combine the weighted predictions to obtain a single weighted prediction
- ... that is much better than the individual predictions themselves!

Boosting example



f( ) = 2 + 0.9= 2.9 f( )= -1 - 0.9 = -1.9

[1] https://xgboost.readthedocs.io/en/latest/model.html

Model evaluation through cross-validation

- Cross-validation: Robust method for estimating the performance

of a model on unseen data

- Generates many non-overlapping train/test splits on training data
- Reports the average test set performance across all data splits

Cross-validation in XGBoost example

```
import xgboost as xgb
import pandas as pd
churn_data = pd.read_csv("classification_data.csv")
churn_dmatrix = xgb.DMatrix(data=churn_data.iloc[:,:-1],
        label=churn_data.month_5_still_here)
params={"objective":"binary:logistic","max_depth":4}
cv_results = xgb.cv(dtrain=churn_dmatrix, params=params, nfold=4,
        num_boost_round=10, metrics="error", as_pandas=True)
print("Accuracy: %f" %((1-cv_results["test-error-mean"]).iloc[-1]))
```

```
Accuracy: 0.88315
```

When should I use XGBoost?

EXTREME GRADIENT BOOSTING WITH XGBOOST

When to use XGBoost

- You have a large number of training samples
  - Greater than 1000 training samples and less 100 features
  - The number of features < number of training samples
- You have a mixture of categorical and numeric features
  - Or just numeric features

When to NOT use XGBoost

- Image recognition
- Computer vision
- Natural language processing and understanding problems
- When the number of training samples is significantly smaller than the number of features