Extreme Gradient Boosting (XGBoost) Algoritması

"Extreme Gradient Boosting" ‘Aşırı Gradyan Artırma’ anlamına gelen XGBoost, Gradient Boosting çerçevesi altında makine öğrenimi algoritmalarını uygulayan açık kaynaklı bir kitaplıktır. Tianqi Chen ve Carlos Guestrin tarafından geliştirilmiştir ve çeşitli makine öğrenimi zorlukları ve yarışmalarındaki mükemmel performansı nedeniyle veri bilimi topluluğunda önemli bir popülerlik kazanmıştır.

XGBoost Nedir?

XGBoost, denetimli bir topluluk makine öğrenimi algoritmasıdır. Gradyan artırma (Gradient Boosting) tekniğine dayalıdır ve her adımda mevcut modelin hatalarını düzelten yeni ağaçlar ekleyerek modeli güçlendirir. XGBoost, bu süreçte hız ve bellek verimliliği sağlayan optimizasyonlara sahiptir ve bu sayede büyük veri setlerinde etkili sonuçlar verir.

XGBoost Özellikleri

XGBoost, regresyon, sınıflandırma ve sıralama dahil olmak üzere çok çeşitli denetimli öğrenme problemlerini ele almak için tasarlanmıştır. XGBoost'u diğer makine öğrenimi algoritmalarından ayıran temel özelliklerden ve avantajlardan bazıları şunlardır:

- Ölçeklenebilirlik ve verimlilik: XGBoost yüksek düzeyde ölçeklenebilirdir ve büyük veri kümelerini ve yüksek boyutlu verileri verimli bir şekilde işleyebilir. Paralel ve dağıtılmış bilgi işlem yetenekleri sunarak büyük veri kümeleri üzerinde hızlı eğitime olanak tanır.
- Düzenlileştirme teknikleri: XGBoost, aşırı uyumu önlemeye ve modelin genelleştirme performansını iyileştirmeye yardımcı olan L1 ve L2 düzenlileştirme gibi yerleşik düzenlileştirme tekniklerini içerir.
- Eksik değerleri işleme: XGBoost, gerçek dünya veri kümelerinde yaygın bir sorun olan verilerdeki eksik değerleri işlemenin etkili bir yoluna sahiptir.
- Ağaç budama: XGBoost, modelin karmaşıklığını azaltmaya ve yorumlanabilirliğini artırmaya yardımcı olan ağaç budamasını destekler.
- Özellik önemi: XGBoost, modeldeki her özelliğin önemini ölçmek için bir yol sağlar, bu da özellik seçimi ve modelin davranışını anlamak için yararlı olabilir.

XGBoost'un Avantajları

- **Hızlı Eğitim**: XGBoost, paralel hesaplama ve diğer optimizasyonlar sayesinde büyük veri setleri üzerinde hızlı eğitim süresi sunar.
- **Overfitting Azaltma**: Düzenleme teknikleri sayesinde, modelin overfitting yapması önlenir.
- **Esneklik**: XGBoost, sınıflandırma, regresyon, sıralama ve daha birçok görevde esnek bir şekilde kullanılabilir.
- **Özellik Önem Sıralaması**: XGBoost, her özelliğin modele katkısını değerlendiren özellik önem sıralaması çıkarır.

Neden XGBoost?

XGBoost, bireylerin ve ekiplerin neredeyse her Kaggle yapılandırılmış veri yarışmasını kazanmasına yardımcı olmanın bir sonucu olarak son birkaç yılda önemli bir iyilik kazandı. Bu

yarışmalarda, şirketler ve araştırmacılar verileri yayınlar, ardından istatistikçiler ve veri madencileri, verileri tahmin etmek ve tanımlamak için en iyi modelleri üretmek için rekabet eder.Başlangıçta XGBoost'un hem Python hem de R uygulamaları oluşturuldu. Popülerliği nedeniyle, bugün XGBoost, Java, Scala, Julia, Perl ve diğer diller için paket uygulamalarına sahiptir. Bu uygulamalar, XGBoost kitaplığını daha da fazla geliştiriciye açtı ve Kaggle topluluğu genelinde çekiciliğini artırdı.

XGBoost Nerelerde Kullanılır?

1. **Sınıflandırma**: Spam e-posta filtreleme, müşteri churn tahmini.
2. **Regresyon**: Ev fiyat tahmini, satış tahmini.
3. **Sıralama**: Arama motoru sıralamaları, öneri sistemleri.
4. **Anomali Tespiti**: Kredi kartı dolandırıcılığı tespiti.

XGBoost Kullanımı

XGBoost, Python, R ve birçok diğer programlama diliyle uyumludur. Python'da XGBoost kullanmak için önce xgboost kütüphanesini kurmanız gerekir: pip install xgboost

Basit Bir Python Örneği

```
import xgboost as xgb
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error


# Veri seti
data = pd.read_csv('dataset.csv')
X = data.drop('target', axis=1)
y = data['target']


# Eğitim ve test setlerine ayırma
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# XGBoost dataset formatına dönüştürme
train_data = xgb.DMatrix(X_train, label=y_train)
test_data = xgb.DMatrix(X_test, label=y_test)
```

```
# Modelin parametreleri
params = {
  'objective': 'reg:squarederror',
  'eval_metric': 'rmse',
  'learning_rate': 0.1,
  'max_depth': 6,
  'min_child_weight': 1,
  'subsample': 0.8,
  'colsample_bytree': 0.8,
  'n_estimators': 100
}

# Modeli eğitme
model = xgb.train(params, train_data, evals=[(test_data, "Test")],
early_stopping_rounds=10)

# Tahmin yapma
y_pred = model.predict(test_data)
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
```

XGBoost'un Önemli Parametreleri

- objective: Modelin amacını belirler (örneğin, regresyon, sınıflandırma).
- eval_metric: Değerlendirme metriğini belirtir (örneğin, RMSE, logloss).
- learning_rate: Her adımda modelin ne kadar öğrenmesini gerektiğini belirler.
- max_depth: Bir ağacın maksimum derinliği.
- min_child_weight: Bir yaprak düğümünde bulunması gereken minimum örnek ağırlığı.
- subsample: Her iterasyonda kullanılan rastgele veri alt kümesinin oranı.
- colsample_bytree: Her iterasyonda kullanılan rastgele özelliklerin oranı.
- n_estimators: Toplam ağaç sayısı.

XGBoost'da Overfitting'i Azaltma Yolları

- **max_depth ve min_child_weight**: Daha düşük değerler, modelin karmaşıklığını azaltarak overfitting'i engeller.

- **subsample ve colsample_bytree**: Her iterasyonda sadece veri ve özelliklerin bir kısmını kullanarak overfitting'i azaltabilir.
- **early_stopping_rounds**: Eğitim sırasında performans iyileşmediğinde erken durma sağlayarak overfitting'in önüne geçer.

SONUÇ

Yukarıda anlatılan sebepler ve uygulama göz önüne alındığında XGBoost hem tahmin gücü hem de algoritma çalışma hızı açısında mevcuttaki en iyi algoritmalardan biridir. Ancak her ne kadar güçlü bir algoritma olsa da işin sırrı hiperparametreleri optimize etmektedir. Düşük öğrenim oranı ve gamma ile başlayıp, yavaş yavaş bu parametreler güncellenerek optimum sayılar bulunabilir.

Bu denemeler için "GridSearchCV" çok büyük kolaylık sağlamaktadır. Eklenen hiperparametrelerin tüm kombinasyonları kadar model kurulup test edileceği göz önüne alınmalıdır. Aksi taktirde uzun saatler sürebilecek bir çalışma başlatılabilir.

XGBoost çok iyi bir algortima olsa da her zaman en iyi sonucu verecek diye bir kaide yoktur. Mümkün olduğunca diğer algoritmalar da denenmelidir.

Kaynakça

- What is XGBoost | XGBoosting
- What is XGBoost? An Introduction to XGBoost Algorithm in Machine Learning | Simplilearn
- What is XGBoost and how to use it. | by EdvardOlsen | Medium
- XGBoost – What Is It and Why Does It Matter? (nvidia.com)
- XGBoost Nasıl Çalışır? Neden İyi Performans Gösterir? - Veri Bilimi Okulu - Veri Bilimi Okulu
- ChatGPT