LightGBM Algoritması

Atıl Samancıoğlu

1 Giriş

LightGBM, Microsoft tarafından geliştirilen, büyük veri setleri üzerinde hızlı ve verimli çalışan bir **Gradient Boosting** algoritmasıdır. Hafıza dostu olması, çok hızlı çalışması ve doğru sonuçlar vermesi ile popülerdir.

Bu dökümanda LightGBM'in nasıl çalıştığını örnekler üzerinden açıklayacağız ve aynı zamanda **XGBoost ile benzerliklerini ve farklarını** adım adım inceleyeceğiz.

2 LightGBM ile XGBoost Karşılaştırması

Özellik	**XGBoost**	**LightGBM**
Büyüme Şekli	Derinlik bazlı (Level-wise)	Yaprak bazlı (Leaf-wise)
Split Seçimi	Tüm ağaç için en iyi split	En çok bilgi kazancı olan yaprak
Hız	Yüksek	Çok daha yüksek
Bellek Kullanımı	Orta	Düşük
Categorical Destek	Kodlama gerekir	Native destek
Histogram Kullanımı	Opsiyonel	Zorunlu (binning yapılır)

Table 1: XGBoost ve LightGBM Karşılaştırması

LightGBM'in En Önemli Özelliği

Leaf-wise büyüme stratejisi sayesinde LightGBM daha hızlı ve daha düşük hata oranıyla çalışır. Ancak overfitting riski de daha yüksektir.

3 Bölüm 1: Yaprak Bazlı Büyüme (Leaf-wise Growth)

XGBoost'ta ağaçlar her seviyede dengeli büyür (*level-wise*). LightGBM ise her adımda yalnızca **en fazla bilgi kazancı sağlayan yaprağı** genişletir.

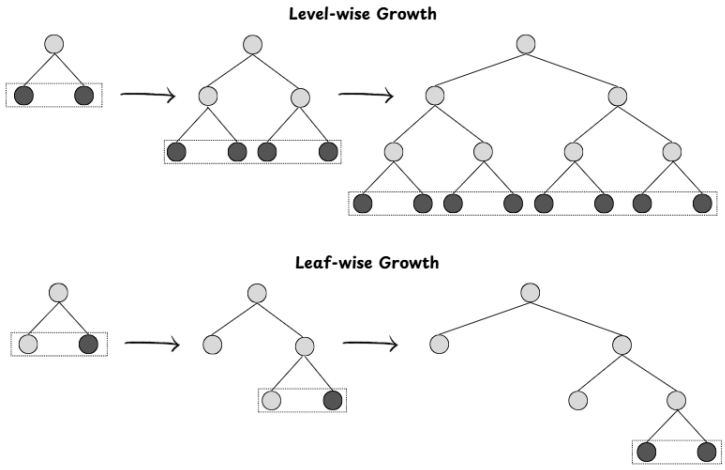

Figure 1: Oli Golfryd Level Wise Growth

Farkı Nasıl Hissediyoruz?

- XGBoost: Ağacın her tarafı dengeli büyür. Karar ağaçları daha "düzenli" görünür.
- LightGBM: Sadece en iyi yaprak büyütülür. Ağaçlar daha "asimetrik" olabilir ama genelde daha düşük hata üretir.

4 Bölüm 2: Histogram ve Binning Mantığı

LightGBM, sürekli değişkenleri kullanmadan önce histogram yapar. Bu hem hız hem de bellek kazancı sağlar.

Neden Histogram?

- Değişkenleri **önce gruplara (bin)** ayırır.
- Her split'te bu bin'ler üzerinden karar verilir.
- 256 bin gibi sabit sayı kullanılır, yani değer sayısı ne olursa olsun işlem sayısı düşer.

Örnek:

Gerçek gelir verileri: [40K, 45K, 46K, 60K, 75K]
LightGBM bunu şu şekilde bin'lere ayırabilir:

- Bin 1: 40K – 45K
- Bin 2: 45K – 60K
- Bin 3: 60K – 75K

Artık model "Gelir <Bin 2?" gibi kararlar verir. Sayı değil, bin numarası üzerinden çalışır.

5 Bölüm 3: Categorical Özelliklerin İşlenmesi

XGBoost'ta kategorik değişkenler sayıya çevrilmeden kullanılamaz. LightGBM ise onları doğal olarak işler.

Örnek:

`CreditScore` değişkeni şu değerleri alıyor:

- Düşük
- Orta
- Yüksek

LightGBM bunu doğrudan öğrenebilir. Split şöyle olabilir:

$$\texttt{CreditScore} \in \{\text{Düşük, Orta}\}$$

XGBoost'ta Ne Olurdu?

Bu değerleri sayıya çevirmek (örneğin LabelEncoder ile 0,1,2 yapmak) gerekirdi. Bu da modelin anlamlı split'ler öğrenmesini zorlaştırır.

6 Bölüm 4: Gain Hesaplaması Farkı

Hem **LightGBM** hem de **XGBoost** bir split (bölünme) yaparken ne kadar fayda sağladığını ölçmek için **gain** (kazanç) hesaplar. Ancak bu kazancı hesaplama yöntemleri birbirinden farklıdır.

XGBoost Gain Formülü:

$$Gain = \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}$$

- Burada G_L ve G_R: sol ve sağ taraftaki toplam **gradyan** değerlerini,
- H_L ve H_R: sol ve sağ taraftaki toplam **Hessian** (ikinci türev) değerlerini,

- λ: regularization (aşırı öğrenmeyi engelleyen ceza katsayısı) gösterir.

Bu formül, bir bölünme öncesi ve sonrası hataları karşılaştırarak ne kadar iyileşme olduğunu hassas şekilde ölçer. Bu nedenle genellikle daha doğru sonuç verir, ancak hesaplaması daha yavaştır.

LightGBM Gain Formülü:

$$Gain = \frac{(sum\_grad)^2}{sum\_hess}$$

- LightGBM, verileri histogramlara (bin) ayırır ve her bin için toplam gradyan ve Hessian değerlerini hesaplar.
- Hesaplama çok daha hızlıdır çünkü her olası eşik değeri tek tek denenmez.

Formül daha basit olduğu için işlem süresi oldukça kısadır. Bu da LightGBM'i büyük veri setleri üzerinde oldukça avantajlı kılar. Ancak bu sade yaklaşım, bazen çok küçük doğruluk kayıplarına yol açabilir.

Farkı Özetle

- XGBoost: Daha detaylı ve hassas gain hesabı yapar, ancak daha yavaştır.
- LightGBM: Daha hızlı çalışır, çünkü histogram (bin) yapısını kullanır ve her split için tüm eşikleri denemez.

Karşılaştırma Tablosu

Özellik	**XGBoost**	**LightGBM**
Gain Hesaplama	Daha detaylı, 3 terimli formül	Daha sade, tek oran formülü
Hız	Görece yavaş	Çok hızlı
Doğruluk	Genelde daha yüksek	Çok yakın, ama bazen az
Yapı	Tüm split noktaları denenir	Bin (aralık) bazlı işlem yapılır
Bellek Kullanımı	Daha fazla	Daha az

7 Sonuç

LightGBM:

- Boosting tabanlı en hızlı çözümlerden biridir,
- XGBoost'tan genellikle daha hızlıdır ama overfitting riski biraz daha yüksektir,
- Kategorik verilerle çalışmada daha kullanıcı dostudur,
- Büyük veriler için idealdir.