Q1. Gradient boosting은 무엇인가?

- Gradient boosting은 머신러닝의 앙상블 학습 방법 중 하나로, 여러 개의 약한 예측기(weak learner)를 결합하여 강력한 예측기(strong learner)를 만드는 기법입니다. Gradient boosting은 오차의 경사(gradient)를 이용하여 모델을 학습시키는 방식으로 동작합니다. 일반적으로 gradient boosting은 트리 기반 모델인 Gradient Boosting Decision Tree (GBDT)를 사용합니다. GBDT는 이전 트리의 예측 오차(residuals)를 줄이는 새로운 트리를 순차적으로 학습하며, 이를 통해 전체 앙상블 모델을 구성합니다. 각 트리는 이전 트리의 오차에 집중하여 예측을 개선하는 방식으로 구성되기 때문에, 오차의 경사를 이용하여 모델이 학습되는 것입니다. 일반적으로 예측 오차를 줄이기 위해 경사 하강법(gradient descent)을 사용하며, 경사를 이용하여 트리의 분할점과 예측값을 조정합니다.

- Gradient boosting은 다양한 문제에 효과적인 예측 성능을 보이는데, 특히 분류(classification)와 회귀(regression) 문제에서 널리 사용됩니다. 그리고 gradient boosting은 다른 앙상블 방법과 달리 특성(feature)의 스케일 조정이나 이상치(outlier) 처리에 덜 민감하며, 변수의 중요도를 평가할 수도 있습니다.

- Gradient boosting에 대한 자세한 내용과 참고할 수 있는 논문 및 학술 자료는 아래와 같습니다.

  1. "Greedy Function Approximation: A Gradient Boosting Machine" - Jerome H. Friedman의 논문으로, gradient boosting의 개념과 알고리즘에 대한 기본 개념을 설명합니다. (논문 링크: http://statweb.stanford.edu/~jhf/ftp/trebst.pdf)
     → 이 논문은 그레디언트 부스팅 머신(Greedy Function Approximation, GFA) 알고리즘에 대한 개념적인 설명과 수학적인 기반을 제공합니다. 그레디언트 부스팅은 기존 모델의 오차를 보완하는 새로운 모델을 반복적으로 학습하여 예측 모델을 개선하는 기법입니다. 이 논문에서는 그레디언트 부스팅의 핵심 아이디어와 알고리즘의 세부 사항을 다루고 있습니다.

  2. "XGBoost: A Scalable Tree Boosting System" - Tianqi Chen 및 Carlos Guestrin의 논문으로, XGBoost라는 유명한 gradient boosting 알고리즘에 대해 자세히 설명합니다. (논문 링크: https://arxiv.org/abs/1603.02754)
     → 이 논문은 XGBoost 알고리즘에 대한 상세한 설명과 기술적인 내용을 다룹니다. XGBoost는 그레디언트 부스팅 결정 트리 기반의 알고리즘으로, 성능과 확장성을 개선하기 위해 여러 기법을 도입하였습니다. 이 논문에서는 XGBoost의 핵심 알고리즘, 정규화, 조기 중단 등에 대한 내용과 실험 결과를 제시하고 있습니다.

3. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree" - Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye 및 Tie-Yan Liu의 논문으로, 경량화된 gradient boosting 알고리즘인 LightGBM에 대해 설명합니다. (논문 링크: (https://papers.nips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf))
   → 이 논문은 LightGBM 알고리즘에 대한 설명과 성능 분석을 제시합니다. LightGBM은 빠른 학습과 예측 속도, 메모리 효율성을 갖는 그레디언트 부스팅 결정 트리 기반의 알고리즘입니다. 이 논문에서는 LightGBM의 분할 방법, 리프 중심 학습, 특징과 성능 분석 등에 대해 자세히 다루고 있습니다.

4. "CatBoost: unbiased boosting with categorical features" - Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin의 논문으로, 범주형 변수를 다룰 수 있는 gradient boosting 알고리즘인 CatBoost에 대해 설명합니다. (논문 링크: https://arxiv.org/abs/1706.09516)
   → 이 논문은 CatBoost 알고리즘의 개요와 특징을 설명합니다. CatBoost는 범주형 변수의 처리에 특화된 그레디언트 부스팅 알고리즘입니다. 범주형 변수의 자동 처리, 순서 고려 분할, 대규모 데이터셋 처리 등의 기능을 가지고 있습니다. 이 논문에서는 CatBoost의 주요 특징과 성능 비교 실험 결과를 제시하고 있습니다.

위 논문들은 gradient boosting의 원리와 다양한 변종 알고리즘들에 대해 자세히 다루고 있고, 위 논문들은 각각 CatBoost, LightGBM, XGBoost에 대해 자세한 설명과 실험 결과를 제공하고 있으며, gradient boosting에 대한 이해를 높이는 데 도움이 될 것입니다.

Q2. LightGBM, XGBoost, CatBoost 모델 작동 방식의 차이점

- LightGBM 설명: LightGBM은 Microsoft에서 개발한 경량화된 gradient boosting 알고리즘입니다. LightGBM은 다른 트리 기반의 gradient boosting 알고리즘들에 비해 빠른 속도와 낮은 메모리 사용량을 가지고 있습니다. 이를 위해 LightGBM은 트리를 수직적으로 분할하면서, Leaf-wise(혹은 Best-first) 방식을 사용합니다. 즉, 가장 정보 획득이 큰 분할을 먼저 선택하여 트리를 성장시킵니다. 또한, LightGBM은 범주형 변수를 자동적으로 처리할 수 있는 기능도 가지고 있습니다.
- XGBoost 설명: XGBoost는 Tianqi Chen이 개발한 gradient boosting 알고리즘으로, eXtreme Gradient Boosting의 약자입니다. XGBoost는 다양한 트리 기반 모델을 합치는 앙상블 학습 방법으로, 일반적으로 GBDT와 함께 사용되며, 대부분의 gradient boosting 알고리즘의 기능과 성능을 갖추고 있습니다. XGBoost는 다양한 기능을 제공하며, 정규화(regularization), 조기 중단(early stopping), 가중치 조정(weight boosting), 결측값 처리 등의 기능을 지원합니다. 또한, 병렬 처리와 분산 학습을 지원하여 대규모 데이터셋에서도 효율적으로 작동할 수 있습니다.
- CatBoost 설명: CatBoost는 Yandex에서 개발한 gradient boosting 알고리즘으로, 범주형 변수를 다루는 데에 특화되어 있습니다. CatBoost는 범주형 변수에 대한 자동적인 처리를 수행하며, 범주형 변수의 순서를 고려한 분할 방법을 사용합니다. 이는 범주형 변수의 특성을 잘 반영하여 예측 성능을 향상시킵니다. 또한, CatBoost는 자체적으로 과적합을 줄이기 위한 기법인 Ordered Target Statistic을 적용하고 있습니다.

- LightGBM, XGBoost, CatBoost는 모두 gradient boosting을 기반으로 한 알고리즘으로, 기본적인 작동 방식은 유사합니다. 하지만 다음과 같은 차이점이 있습니다.

1) LightGBM: LightGBM은 Leaf-wise(혹은 Best-first) 방식을 사용하여 트리를 성장시킵니다. 이는 현재 상황에서 정보 획득이 가장 큰 분할을 선택하고, 그에 따라 트리를 성장시키는 방식입니다. 이 방식은 기존의 Level-wise 방식보다 더 빠른 학습 속도와 낮은 메모리 사용량을 제공합니다. LightGBM은 일반적으로 히스토그램 기반의 분할 방법을 사용하여 데이터를 이산화(discretization)합니다. 이를 통해 범주형 변수를 자동으로 처리하고 메모리 사용량을 줄일 수 있습니다. LightGBM은 Leaf-wise 분할로 인해 오버피팅이 발생할 가능성이 있으므로, 일반적으로 학습률(learning rate)과 트리 깊이(max_depth) 등의 하이퍼파라미터를 조절하여 제어합니다.
2) XGBoost: XGBoost는 Level-wise 방식을 사용하여 트리를 성장시킵니다. 이는 현재까지의 정보 획득을 동일한 깊이의 모든 노드에 적용한 다음, 가장 정보 획득이 큰 분할을 선택하여 트리를 성장시키는 방식입니다. XGBoost는 정규화(regularization)를 통해 오버피팅을 줄이는 기능을 제공합니다. 정규화 항은 트리의 복잡도를 제어하고, 조기 중단(early stopping)을 통해 적절한 반복 횟수를 찾아내는 데에 사용됩니다. XGBoost는 가중치 조정(weight boosting)과 결측값 처리 등의 다양한 기능을 지원합니다.
3) CatBoost: CatBoost는 범주형 변수를 다루는 데에 특화된 gradient boosting 알고리즘입니다. CatBoost는 범주형 변수의 순서를 고려한 분할 방법을 사용하여 트리를 성장시킵니다. CatBoost는 자체적으로 과적합을 줄이기 위한 기법인 Ordered Target Statistic을 적용합니다. 이는 트리를 성장시킬 때, 범주형 변수의 특성을 고려하여 예측 성능을 향상시킵니다. CatBoost는 범주형 변수의 자동 처리 기능을 제공합니다. 이는 범주형 변수를 자동으로 이진 분할하여 처리하고, 명시적인 인코딩이 필요하지 않습니다.

[요약]

1. 학습 알고리즘: LightGBM은 Leaf-wise 방식으로 트리를 성장시키는 반면, XGBoost는 Level-wise 방식을 사용합니다. CatBoost는 Ordered Target Statistic을 사용하여 범주형 변수의 순서를 고려한 분할을 수행합니다.
2. 속도와 메모리 사용량: LightGBM은 효율적인 분할 전략과 병렬 처리 기능을 통해 빠른 학습 속도와 낮은 메모리 사용량을 가지고 있습니다. XGBoost와 CatBoost도 속도와 메모리 사용량이 일반적인 gradient boosting 알고리즘보다 향상되었지만, LightGBM보다는 상대적으로 느릴 수 있습니다.
3. 범주형 변수 처리: LightGBM와 XGBoost는 범주형 변수를 자동으로 처리할 수 있는 기능을 제공하지 않지만, CatBoost는 범주형 변수 처리에 특화되어 있습니다.

Q3. 데이터셋의 크기나 특징별로 모델마다 성능의 차이가 나타나는 이유가 무엇인가?

1. LightGBM, XGBoost, CatBoost 모델은 데이터셋의 크기와 특징에 따라 성능의 차이가 발생하는 이유는 다음과 같습니다.

- 데이터셋 크기:
    1) LightGBM: LightGBM은 Leaf-wise 방식으로 트리를 성장시키는데, 이는 적은 수의 분할로 더 깊은 트리를 구성할 수 있게 합니다. 이로 인해 대규모 데이터셋에서도 빠른 학습 속도를 보이는 특징이 있습니다. 또한, LightGBM은 메모리 사용량을 최적화하여 대용량 데이터셋을 처리할 수 있습니다.
    2) XGBoost: XGBoost는 Level-wise 방식으로 트리를 성장시키는데, 이는 트리의 모든 레벨에서 분할을 수행하므로 작은 데이터셋에서는 성능이 우수합니다. 그러나 대규모 데이터셋에서는 LightGBM보다 메모리 사용량이 더 많아지고 학습 시간이 늘어날 수 있습니다.
    3) CatBoost: CatBoost는 대규모 데이터셋에서도 뛰어난 성능을 보입니다. 이는 샘플의 순서를 고려한 분할 방법과 범주형 변수의 자동 처리 기능을 사용하기 때문입니다.
- 데이터셋 특징:
    1) LightGBM: LightGBM은 데이터셋의 특징이 다양한 경우에 효과적입니다. 특징이 다양한 데이터셋에서는 각 분할에서 더 적은 레벨을 만들고 더 정확하게 예측할 수 있습니다. 또한, LightGBM은 희소 데이터셋에서도 효율적으로 처리할 수 있는 기능을 가지고 있습니다.
    2) XGBoost: XGBoost는 정규화(regularization) 기능을 제공하여 이상치나 노이즈에 대해 더 강건한 모델을 만들 수 있습니다. 이는 트리의 복잡도를 제어하고 오버피팅을 방지하는 데 도움이 됩니다. 따라서, 데이터셋에 이상치나 노이즈가 있는 경우에 XGBoost가 성능이 우수할 수 있습니다.
    3) CatBoost: CatBoost는 범주형 변수의 처리에 특화되어 있습니다. 범주형 변수의 순서를 고려한 분할 방법을 사용하여 범주형 변수를 효과적으로 처리할 수 있습니다. 이는 범주형 변수가 많거나 중요한 데이터셋에서 CatBoost가 성능이 우수한 이유입니다.

2. LightGBM과 XGBoost 비교:
    1) 성능: 많은 비교 연구에서는 LightGBM과 XGBoost가 유사한 성능을 보입니다. 두 모델 모두 앙상블 학습과 그레디언트 부스팅을 기반으로 하기 때문에 훈련 오차를 지속적으로 감소시키는 데 강점을 가지고 있습니다.
    2) 속도: LightGBM은 Leaf-wise 분할 방식과 효율적인 분할 전략을 사용하여 대규모 데이터셋에서 빠른 학습 속도를 제공합니다. 반면 XGBoost는 Level-wise 분할 방식을 사용하므로 작은 데이터셋에서 빠른 학습 속도를 보입니다.
    3) 메모리 사용량: LightGBM은 효율적인 메모리 사용량을 가지고 있어 대규모 데이터셋을 처리할 때 유리합니다. XGBoost는 대규모 데이터셋에서 더 많은 메모리를 사용하는 경향이 있습니다.

3. LightGBM과 CatBoost 비교:
    1) 성능: LightGBM과 CatBoost는 비슷한 성능을 보이는 경향이 있습니다. 그러나 데이터셋에 따라 상이할 수 있습니다. CatBoost는 범주형 변수의 처리에 특화되어 있고 범주형 변수가 많은 데이터셋에서 성능이 우수합니다. LightGBM은 다양한 특징을 가진 데이터셋에서 좋은 성능을 보입니다.
    2) 범주형 변수 처리: CatBoost는 범주형 변수의 자동 처리 기능과 범주형 변수의 순서를 고려한 분할 방법을 사용하여 범주형 변수를 효과적으로 처리합니다. LightGBM은 범주형 변수를 자동으로 처리할 수 있는 기능을 가지고 있지만, CatBoost에 비해 약간의 성능 하락이 있을 수 있습니다.

4. XGBoost와 CatBoost 비교:
    1) 성능: XGBoost와 CatBoost는 유사한 성능을 보입니다. 그러나 데이터셋에 따라 상이할 수 있습니다. CatBoost는 범주형 변수의 처리에 특화되어 있고 범주형 변수가 많거나 중요한 데이터셋에서 성능이 우수합니다. XGBoost는 정규화(regularization) 기능을 제공하여 이상치나 노이즈에 대해 더 강건한 모델을 만들 수 있습니다.
    2) 속도: XGBoost는 일반적으로 CatBoost보다 학습 시간이 짧습니다.
    3) 메모리 사용량: CatBoost는 대규모 데이터셋에서도 효율적인 메모리 사용을 보장합니다. XGBoost는 메모리 사용량이 큰 경향이 있습니다.

기타 참고 자료 및 웹사이트

CatBoost v. XGBoost v. LightGBM - Kaggle

CatBoost Vs XGBoost Vs LightGBM - YouTube

XGBoost vs. CatBoost vs. LightGBM: How Do They Compare?

When to Choose CatBoost Over XGBoost or LightGBM ...

https://www.researchgate.net/publication/351133481_Comparison_of_Gradient_Boosting_Decision_Tree_Algorithms_for_CPU_Performance

https://scholar.google.co.kr/scholar?q=A+Comparative+Study+of+Gradient+Boosting+Decision+Trees%22&hl=ko&as_sdt=0&as_vis=1&oi=scholart