데이터 분석 업그레이드

학습 목표

▶ 머신러닝 기본에서 다루지 못한 추가적인 내용을 소개

▶ 이후의 학습에 도움을 주는 내용을 담아보기

- 참고 서적 : 데이터가 뛰어노는 AI 놀이터, 캐글

목차

02 분류의 평가지표

▶ 다중 클래스 분류(multi-class classification)

다중 클래스 로그 손실(multi-class logloss)

▶ 다중 레이블 분류(multi-label classification)

다중 레이블 분류(mean-F1 or macro-F1)

02 분류의 평가지표

▶ 추천 문제 – 순위를 매겨 제출

MAP@K – mean Average Precision at @ K의 약자

▶ 순위를 매기지 않을 때

mean-F1, macro-F1, micro-F1

=> F1-score 평가지표를 여러 개의 클래스로 확장

03 정형 데이터, 외부 데이터, 시계열 데이터

▶ 정형 데이터

- 정형 데이터(tabular data)로 표 형식 데이터라 불리기도 한다. 행과 열의 데이터를 가지고 보통 캐글에서는 csv와 같은 확장자를 가지는 데이터를 말함.

▶ 외부 데이터

▶ 시계열 데이터

- 시간의 흐름과 함께 관측된 데이터를 시계열 데이터라고 한다.

04 이진 분류의 평가지표(2)

▶ 로그 손실(logloss)

- 분류 문제의 대표적인 평가지표로서 교차 엔트로피(cross-entropy)라 불리기도 한다.

$$\text{logloss} = -\frac{1}{N}\sum_{i=1}^{n}(y_i \log p_i - (1 - y_i)\log(1 - p_i))$$

$$= -\frac{1}{N}\sum_{i=1}^{n}\log p_i'$$

y_i : 실제값(양성 :1, 음성 : 0)

p_i : 각 행 데이터가 양성일 예측 확률

p_i' : 실젯값을 예측하는 확률

* 로그 손실은 낮을수록 좋은 지표를 갖는다.

05 결측값 처리

▶ 결측값 처리하기

- 결측값을 채우지 않고 그대로 쓰기(GBDT 모델 가능, 랜덤 포레스트는 안됨.)
- 결측값을 -9999 처럼 값을 대입하여 처리하기(결측값을 채우지 않는 방법과 비슷)
- 해당 feature(특징)의 대푯값으로 채우기
  - 수치형 특징은 평균값, 중앙값 이용.
  - 범주형 특징은 가장 많은 수를 갖는 값으로 채우기
- 결측치 자체를 하나의 범줏값으로 보고 이를 새로운 범줏값으로 변경하기(범주형 특징)
- 결측값을 가진 특징(변수)가 다른 변수와 관련이 높을 때, 그 변수들로부터 원래의 값 예측(모델 만들기)

05 결측값 처리

▶ 결측값 처리하기

- 해당 결측값으로 새로운 특징(변수) 생성(0, 1 값을 갖는) 생성.

- 행 데이터마다 결측값이 있는 특징(변수)의 수를 카운팅하여 생성.

- 여러 개의 변수에서 결측값의 조합을 조사하여 몇 개의 패턴으로 분류할 수 있다면,
  어느 패턴인지 확인하는 특징(변수) 생성.

06 수치형 특징(변수) 변환

▶ 수치형 특징(feature) 변환하기

기본적으로 특징 그 자체로 사용할 수 있지만, 적절하게 변환하거나 가공하면 더 효과적인 특징을 만들어 낼 수 있다.

A. 표준화 (StandardScaler) – 평균 0, 분산 1로 변경

B. 정규화 (MinMaxScaler) - 최대 최소 스케일링

C. 로그 변환, log(x+1)변환, 절댓값 로그 변환

D. Box-Cox(박스 칵스 변환), Yeo-Johoson(여-존슨 변환)

07 수치형 특징 변환-clipping

▶ 클리핑(clipping)

상한과 하한을 설정한 뒤, 해당 범위를 벗어나는 값은 상한 값과 하한 값으로 치환한다.

Pandas 모듈이나 numpy 모듈의 clip 함수를 이용 가능.

07 수치형 특징 변환-구간 분할

▶ 구간 분할(binning)

수치형 변수를 구간별로 나누어 범주형 변수로 변환하는 방법.

A. 같은 간격으로 분할하는 방법

B. 분위점을 이용하여 분할하는 방법

C. 구간 구분을 지정하여 분할하는 방법

* Pandas 모듈의 cut함수, numpy모듈의 digitize 함수 이용 가능.

07 수치형 특징 변환-순위로 변환

▶ 순위 변환

수치형 변수를 대소 관계에 따른 순위로 변환하는 방법.

▶ RankGauss

수치형 변수를 순위로 변환한 뒤, 순서를 유지한 채 반강제로 정규분포가 되도록 변환.

08 범주형 특징 변환

▶ 범주형 변수 변환

● 특별히 대응하지 않아도 영향이 적음을 확인

- 테스트 데이터에만 존재하는 범주가 있을 때, 이 경우,행 데이터가 적을 때, 점수에 거의 영향을 끼치지 않는다.

● 최빈값 또는 예측으로 보완

● 변환할 때 해당 변환의 평균에 가까운 값 입력

08 범주형 특징 변환 – 원핫 인코딩

▶ 원핫 인코딩(One Hot Encoding)

- 범주형 변수의 가장 대표적인 처리 방법

  A. 범주형 변수의 각 레벨에 대해 해당 레벨인지 여부를 나타내는 0과 1 두 값을 갖는 변수를 각각 생성

  B. N개의 레벨을 갖는 범주형 특징에 원-핫 인코딩을 적용하면 두 값(0,1)을 갖는 특징이 n개 만들어진다. 이들 두 값의 특징(변수)를 가변수(dummy variable)라 한다.

- 특징의 개수가 범주형 레벨 개수에 따라 증가한다는 중대한 결점 존재.

08 범주형 특징 변환 – 레이블 인코딩

▶ 레이블 인코딩(label encoding)

● 각 레벨을 단순히 정수로 변환.

A. 예를 들어 5개의 레벨이 있는 범주형 변수를 레이블 인코딩하면 각 값이 0에서 4까지 수치로 변경됨.

B. 보통은 레벨을 문자열로 보고 사전 순으로 나열된 인덱스로 변경

08 범주형 특징 변환 – 특징 해싱

▶ 특징 해싱(feature hashing)

● 원 핫 인코딩은 변환한 뒤, 특징의 수는 범주의 레벨 수와 같아진다. 특징 해싱(feature hashing)은 이의 수를 줄이는 방법이다.

A. 변환 후의 특징 수를 먼저 정해두고, 해시 함수를 이용하여 레벨별로 플래그를 표시할 위치를 결정.

● 범주형 변수가 많고 원-핫 인코딩에서 생성되는 특징이 지나치게 많을 때, 이용 가능.

다만, 경진대회에서는 GBDT로 학습하여 어느 정도 대응이 가능하므로 이 방법은 널리 쓰이지 않음.

08 범주형 특징 변환 – frequency encoding

▶ frequency encoding(프리퀀시 인코딩)

- 프리퀀시 인코딩은 각 레벨의 출현 횟수 혹은 출연 빈도로 범주형 변수를 대체하는 방법.
- 각 레벨의 출현 빈도와 목적변수 간의 관련성이 있을 때, 유효하다.

08 범주형 특징 변환 – 타깃 인코딩

▶ target encoding(타깃 인코딩)

- 타깃 인코딩은 목적 변수를 이용하여 범주형 변수를 수치형 변수로 변환하는 방법.

08 범주형 특징 변환 – 임베딩

▶ 임베딩(embedding)

- 자연어 처리에서 단어나 범주형 변수와 같은 이산적인 표현을 실수 벡터로 변환하는 방법.
- 임베딩을 다른 말로 분산 표현(distributed representation)이라고도 한다.
- 자연어 처리에서 단어에 대한 학습이 끝난 단어 임베딩 종류에는 word2Vec, GloVe, fastText등이 있다.

09 특징의 조합

▶ 특징(변수)의 조합

- 여러 개의 변수를 조합하여 변수 간 상호 작용을 표현하는 특징을 만들 수 있음.
- 특징끼리 기계적으로 무작정 조합하면 의미가 없는 변수가 대량 생산된다.

  따라서 **데이터에 관한 배경 지식을 활용**하여 어떤 식의 조합이 의미가 있을지 연구하고 특징을 만들어간다.

09 특징의 조합

▶ 수치형 특징 x 범주형 특징

- 범주형 변수의 레벨별로 수치형 변수의 평균이나 분산과 같은 통계량을 새로운 특징으로 생성

▶ 수치형 특징 x 수치형 특징

- 수치형 변수를 사칙 연산하여 새로운 특징 생성

  (예) Zillow Prize 대회에서는 집의 면적과 방의 개수라는 특징에 나눗셈을 적용하여 특징 생성.

09 특징의 조합

▶ 범주형 특징 x 범주형 특징

● 여러 개의 범주형 변수를 조합하여 새로운 범주형 변수를 만들 수 있음.

▶ 행의 통계량 구하기

● 행의 방향, 즉 행 데이터별로 여러 변수의 통계량을 구하는 방법.

- 전체 변수가 아닌 일부 변수로 범위를 좁히는 편이다.

- 결측치, 제로, 마이너스 값의 수를 계산함. 평균, 분산, 최대, 최소 등의 통계량 계산

10 자연어 처리 기법

▶ Bag-of-words(BoW)

- 문장 등의 텍스트를 단어로 분할하고, 각 단어의 출현 수를 순서에 상관없이 단순하게 세는 방식.

▶ n-gram

- BoW에서 분할하는 단위를, 단어가 아닌 연속되는 단어 뭉치 단위로 끊는 방법.

10 자연어 처리 기법

▶ tf-idf

- BoW에서 작성한 단어- 문서 카운트 행렬을 변환하는 기법.

▶ 단어 임베딩

- 단어를 수치 벡터로 변환하는 방법을 단어 임베딩(word embedding)라 한다.

11 검증 방법

▶ 홀드 아웃 검증

● 가장 간단한 방법으로 학습 데이터의 일부를 학습에 사용하지 않고, 검증용으로 남겨두는 것.

▶ 교차 검증

● 학습 데이터를 분할하고 홀드아웃 검증 절차를 여러 번 반복.

11 검증 방법

▶ 층화 K-겹 검증

- 클래스의 비율을 맞추어 검증을 수행. (StratifiedKFold)

▶ 그룹 k-겹 검증

- GroupKFold 클래스를 준비하여 검증.

11 검증 방법

▶ LOO 검증

- leave one out 검증

12 리더 보드의 정보 활용

▶ 캐글의 'TRUST YOUR CV'

● 교차 검증을 신용하라. Public Leaderboard의 점수에 현혹되지 말고, 검증에 의해 모델을 평가하고 일반화 성능이 좋은 것을 찾아내는 것이 중요

12 리더 보드의 정보 활용

▶ 검증과 Leaderboard 점수 차이를 고찰

● 검증 점수와 Public Leaderboard의 점수가 잘 맞는다면, 학습 데이터와 테스트 데이터의 성질이 가깝고 검증도 잘 진행됨. 이때 안심하고 검증 진행.

● 그렇지 않다면, 다음과 같은 가능성을 생각해 볼 수 있음.

(1) 우연에 의한 점수의 편차

(2) 검증 데이터와 테스트 데이터의 분포가 다름

(3) 검증 설계가 부적절하여 일반화 성능을 제대로 평가 못함.

13 검증 데이터와 Public Leaderboard의 과적합

▶ 지나친 시도에 따른 과적합

● 매개 변수 튜닝으로 수많은 시도를 반복하는 등 검증 데이터의 점수를 참조하여 지나치게 취사선택하면 검증 데이터에 과적합 할 수 있음.

=> 매번 제출할 때, 현재의 검증 점수와 Public Leaderboard의 점수를 플롯하여 감각적으로 불균형의 영향을 파악

13 검증 데이터와 Public Leaderboard의 과적합

▶ 교차 검증의 분할을 변경

● 지나친 매개변수 튜닝에 따른 검증 데이터의 과적합을 방지하기 위해 매개변수 튜닝에 이용하는 교차 검증의 분할과 모델의 좋고 나쁨을 평가하는 분할을 서로 바꾸어 보기

A. 한 분할에 의한 교차 검증에 따라 매개변수 튜닝 후, 최적의 매개변수 선택.

B. A와는 다른 분할을 이용한 교차 검증에 따라 A에서 선택한 매개변수로 모델을 평가.

14 모델 튜닝 – 매개변수 튜닝

▶ 하이퍼 파라미터 탐색 방법

- 수동으로 매개 변수 조정
- 그리드 서치와 랜덤 서치(grid search, random search)
- 베이즈 최적화(Bayesian optimization)
  - hyperopt라는 라이브러리를 많이 사용.
  - 2018년 optuna라는 라이브러리 공개.

14 모델 튜닝 – 매개변수 튜닝

▶ 매개 변수 튜닝의 포인트

● 중요한 매개변수와 크게 중요하지 않는 매개변수 중에 중요한 매개변수부터 조정해 나간다.

● 매개 변수의 어느 범위를 탐색했을 때, 그 상한 또는 하한 매개변수에 좋은 점수 평가가 집중되어 있다면 범위를 넓혀 탐색하는 게 좋음.

● 학습시의 난수 시드를 지정.

● 모델의 난수 시드와 폴드 분할의 난수 시드를 변경 시의 점수 변화를 보고, 매개변수를 변경했을 때, 점수 변화가 단순한 랜덤성인지 아니면 매개변수의 변경으로 개선된 결과인지 추측 가능.

15 앙상블 – 간단한 앙상블 기법

▶ 평균과 가중 평균

- 모델의 성능을 보면서 적절하게 결정.

  - Public leaderboard를 보면서 성능이 높은 모델에는 다른 모델의 3배에 달하는 가중치를 주는 식으로 결정.

- 점수가 가장 높아지도록 최적화 – scipy.optimize 모듈 등을 사용.

▶ 스태킹

- 둘 이상의 모델을 조합하여 효율적이면서도 성능이 높아지도록 예측하는 방법.

15 앙상블 – 간단한 앙상블 기법

▶ 모델을 선택할 때,Tip

● 검증 결과를 로그로 출력하고 각 모델의 점수를 파악할 수 있도록 한다.

● 모델의 다양성을 평가하고자 모델 예측값의 상관계수를 계산하거나 다른 모델의 예측값 간 산포도를 플롯한다.

● 모델 검증에서의 점수와 해당 모델을 단독으로 public leaderboard 제출시의 점수를 플롯한다.