Text2EmojiCategory Model Metrics

Text2EmojiCategory Model Metrics

출처, https://ils.unc.edu/courses/2013_spring/inls509_001/lectures/10-EvaluationMetrics.pdf, hands-on Machine Learning-한빛미디어

0. Introduction

Text2EmojiCategory의 목표: 주어진 문장에 대해서 적절한 스티커 카테고리를 추천한다. Text2EmojiCategory는 52개 카테고리 분류 문제인 multi-class 문제이다. 딥러닝 모델은 52개의 카테고리 중 하나를 추천하고 있고, 기존 추천 방식에 대해 살펴보면 다음과 같다.

- 주어진 문장에 대해서, 52개의 카테고리에 ranking을 부여한다.
- 52개의 카테고리 ranking 중 top-1 카테고리를 추천한다.

모델의 성능 측정을 위해서는 accuracy를 사용하고 있다. Accuracy는 다음과 같이 정의된다.

$$accracy = \frac{\text{모델이 카테고리를 맞춘 문장 개수}}{\text{전체 문장 개수}}$$

---

1. Binary-Class Metrics

후술할 개념들을 위해서 classification의 가장 기초인 binary classification을 살펴보자. binary classification이란 주어진 데이터을 두 그룹 중 하나로 분류하는 분류 문제이다. Binary classifier 모델을 학습, 예측시킨다면 다음과 같은 4가지 결과가 있을 수 있다. 편의상 두 그룹은 True와 False로 나누겠다(A그룹과 B그룹이 있다면, A그룹과 B그룹이라는 표현말고, A그룹인가? A그룹이 아닌가?라고 표현할 수 있기 때문에).

- True 데이터에 대해, True로 예측(정답)
- False 데이터에 대해, True로 예측(오답)
- True 데이터에 대해, False로 예측(오답)
- False 데이터에 대해, False로 예측(정답)

이렇게 예측한 것을 앞으로

- True Positive
- False Positive
- False Negative
- True Negative

로 칭한다.

1-1. Confusion Matrix

Confusion Matrix란 위에 서술한

- TP
- FP
- FN
- TN

에 대한 정보를 하나에 담은 Matrix다. Binary classification의 confusion matrix는 다음과 같이 표현될 수 있다.

1.

_	**모델이 True라고 예측**	**모델이 False라고 예측**
실제로 True	TP - True Positive	FN - False Negative
실제로 False	FP - False Positive	TN - True Negative

2.

Matrix를 아래와 같이도 표현할 수 있다.

_	**실제로 True**	**모델이 False라고 예측**
모델이 True라고 예측	TP - True Positive	FP - False Positive
모델이 False라고 예측	FN - False Negative	TN - True Negative

3.

물론 각각의 row나 columns의 순서를 바꿔서 표현할 수도 있다.

하지만 우리는 sklearn의 표기 방법에 따라 아래와 같이 표현한다. sklearn

_	**모델이 False라고 예측**	**모델이 True라고 예측**
실제로 False	TN - True Negative	FP - False Positive
실제로 True	FN - False Negative	TP - True Positive

1-2. Accuracy, Precision, Recall, F1-Score

Accuracy

Accuracy는 위에서 언급한 것과 마찬가지로 다음과 같이 정의된다.

$$accuracy = \frac{TN + TP}{TN + FP + FN + TP}$$

풀어 말하면 전체 데이터 중 맞은 것의 비율이라고 할 수 있다.

우리가 binary classification을 할 데이터가 1000개가 있다고 하자.

1. 데이터는 True 500개, False 500개로 구성되어 있고, 0.5의 확률로 True, 0.5의 확률로 False라고 예측하는 모델이 있다고 하자(모델1). 그렇다면 모델1의 accuracy는 0.5에 근접할 것이다.
2. 만약 데이터가 True 100개 False 900개로 구성되어있어도, 모델1의 accuracy는 0.5에 근접할 것이다.
3. 만약 데이터가 True 100개 False 900개로 구성되어있을 때, 1.0의 확률로 False를 예측하는 모델이 있다면(모델2), 이 모델의 accuracy는 0.9가 된다.

3번 예제에서 모델2는 accuracy는 0.9가 나오지만 과연 이 모델이 좋다고 할 수 있을지는 의문이다. 이 예제는 정확도를 분류기의 성능 측정 지표로 선호하지 않는 이유를 보여준다. 특히 imbalanced data, 즉 데이터 개수가 불균형한 데이터셋을 다룰 때 더욱 그러하다.

Precision and Recall

Binary classifier의 성능을 측정할 때는 confusion matrix를 조사하는 것도 하나의 방법이다.

- 예제 1의 confusion matrix

_	**모델이 False라고 예측**	**모델이 True라고 예측**
실제로 False	250	250
실제로 True	250	250

- 예제 2의 confusion matrix

_	모델이 False라고 예측	모델이 True라고 예측
실제로 False	450	450
실제로 True	50	50

- 예제 3의 confusion matrix

_	모델이 False라고 예측	모델이 True라고 예측
실제로 False	900	0
실제로 True	100	0

Confusion matrix를 보게 되면 많은 정보를 얻을 수 있지만 요약된 정보도 필요하다. 이때 주로 사용되는 것이 precision 그리고 recall이다.

precision은 다음과 같이 정의된다.

$$
\begin{aligned}
precision &= \frac{TP}{TP+FP} \\
&= \frac{\text{그 중 진짜 True 개수}}{\text{모델이 True로 예측한 데이터 개수}}
\end{aligned}
$$

recall은 다음과 같이 정의된다.

$$
\begin{aligned}
recall &= \frac{TP}{TP+FN} \\
&= \frac{\text{그 중 True로 예측한 개수}}{\text{실제 True 데이터 개수}}
\end{aligned}
$$

- 참고

_	모델이 False라고 예측	모델이 True라고 예측
실제로 False	TN - True Negative	FP - False Positive
실제로 True	FN - False Negative	TP - True Positive

precision과 recall을 풀어말하면 다음과 같다.

- precision(FP) - (True라고 예측했는데, 틀린 데이터의 개수)에 대한 정보를 가지고 있음
- recall(FN) - (실제 True인데, True라고 예측 못한 데이터의 개수)에 대한 정보를 가지고 있음

Precision/Recall Trade-Off and F1-Score

Precision과 Recall은 서로 trade-off 관계에 놓여있다.

문장	^^?	흑	아 제	와 신난	제	아	힝..	너무 좋	행벅	행복해

		흑	발	다	길	싸	ㅠ	아	ㅋ	^^
Confidence	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
True Label	X	X	X	O	X	O	X	O	O	O

- 모델은 주어진 문장이 True에 속할 확률을 추정할 수 있는데 이를 Confidence라고 한다.

위의 예시에선 모델이 주어진 10개의 문장에 대해 각각의 문장이 happy 카테고리에 속할 확률을 추정했다. 여기에 threshold라는 개념을 추가하여 한 문장의 추청 확률이 threshold의 이상이 되면 모델은 그 문장을 happy 카테고리에 속한다고 예측한다. 일반적인 binary classification의 threshold 초기값은 0.5이다.

- Threshold = 0.5

문장	^^?	흑 흑	낰좈 앜	와 신난 다	제 길	아 싸	힝.. ㅠ	너무 좋 아	행벅 ㅋ	행복해 ^^
Confidence	0.0	0.1	0.2	0.3	0.4	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
True Label	X	X	O	O	X	O	X	O	O	O
Model Prediction	X	X	X	X	X	**O**	**O**	**O**	**O**	**O**

  - Precision = 4/5
  - recall = 4/6
- Threshold = 0.8

문장	^^?	흑 흑	낰좈 앜	와 신난 다	제 길	아 싸	힝.. ㅠ	너무 좋 아	행벅 ㅋ	행복해 ^^
Confidence	0.0	0.1	0.2	0.3	0.4	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
True Label	X	X	O	O	X	O	X	O	O	O
Model Prediction	X	X	X	X	X	**X**	**X**	**X**	**O**	**O**

  - Precision = 2/2
  - recall = 2/6
- Threshold = 0.3

문장	^^?	흑 흑	낰좈 앜	와 신난 다	제 길	아 싸	힝.. ㅠ	너무 좋 아	행벅 ㅋ	행복해 ^^
Confidence	0.0	0.1	0.2	0.3	0.4	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
True Label	X	X	O	O	X	O	X	O	O	O
Model	X	X	X	O	O	**O**	**O**	**O**	**O**	**O**

Prediction										

- Precision = 5/7
- Recall = 5/6

이렇게 Threshold를 어떻게 설정하냐에 따라서 Precision과 Recall은 반대로 움직이며, 상황에 따라 Precision을 높게 하는 것을 목표로 할지, Recall을 높게 하는 것을 목표로할 지 판단해야 한다.

- Threshold가 높다면,
  - precision이 높고, recall이 낮다
  - 주어진 문장에 대해, True라고 말한다면 거의 틀리지 않는다.
  - 실제 True인 문장에 대해서도 많은 경우 False라도 답한다.
- Threshold가 낮다면,
  - recall이 높고, precision이 낮다.
  - 주어진 문장에 대해, True라고 말하지만 틀릴 가능성이 높다.
  - 실제 True인 문장에 대해서 True라고 말할 가능성이 높다.

Precision과 recall을 둘 다 고려한 수치인 F-Measure가 있다.

$$\text{F-Measure} = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

만약 $\alpha = 1/2$라면, F1-Score가 된다.

$$\text{F1-Score} = \frac{2}{P^{-1} + R^{-1}}$$

- precision이 recall보다 중요한 상황이라면 α값을 낮추고,
- recall이 precision보다 중요한 상황이라면 α값을 높여야 한다.

---

2. Multi-Class Metrics

Text2EmojiCategory는 주어진 문장에 대해서 적절한 스티커 카테고리를 추천하는 서비스다. 따라서, 우리는 binary classification 문제를 해결하는 것이 목표가 아니라 multi classification 문제를 해결하는 것이 목표이다.

2-1. Confusion Matrix

multi classification 문제의 confusion matrix는 다음과 같다.

_	모델이 Class1이라고 예측	모델이 Class2라고 예측	모델이 Class3라고 예측
실제로 Class1			
실제로 Class2			
실제로 Class3			

임의로 예시를 만들어보자.

_	모델이 None이라고 예측	모델이 Sad라고 예측	모델이 Happy라고 예측
실제로 None	300	100	110
실제로 Sad	4	20	5
실제로 Happy	6	7	20

이 경우 accuracy는 다음과 같다.

$$accuracy = \frac{300 + 20 + 20}{(300 + 100 + 110) + (4 + 20 + 5) + (6 + 7 + 20)}$$

Imbalanced data이므로 precision과 recall을 봐야한다. Precision과 recall은 각각의 class별로 볼 수 있다.

- Class None

_	모델이 None이 아니라고 예측	모델이 None이라고 예측
실제로 None이 아님	관심 X	4 + 6
실제로 None	100+110	300

  - precision_none = 300 / (300 + 4 + 6)
  - recall_none = 300 / (300 + 100 + 110)
- Class Sad

_	모델이 Sad가 아니라고 예측	모델이 Sad라고 예측
실제로 Sad가 아님	관심 X	100 + 7
실제로 Sad	4 + 5	20

- Class Happy

_	모델이 Happy가 아니라고 예측	모델이 Happy라고 예측
실제로 Happy가 아님	관심 X	110 + 5
실제로 Happy	6 + 7	20

2-2. Micro-Average, Macro-Average, Weighted-Average

이렇게 구한 class 별 precision, recall을 어떻게 평균을 내느냐도 하나의 선택 사항이다.

- Micro-Average Precision

$$\frac{\text{total true positives}}{\text{total true positives} + \text{total false postives}}$$

- Macro-Average Precision

$$\frac{1}{Y} \cdot \sum_{k=1}^{Y} Precision_k$$

  - where Y is the number of classes.
- Weighted-Average Precision

$$\sum_{k=1}^{Y} w_k \cdot Precision_k$$

  - where $w_k = \frac{\#\text{ of data in the class k}}{\#\text{ of total data}}$
- Micro, macro, weighted average recall도 같은 방식으로 구한다.

2-3. 결론

Text2EmojiCategory는 주어진 문장에 대해서 적절한 스티커 카테고리를 추천하는 서비스다.

주어진 문장에 대해서 하나의 카테고리를 추천했을 때,

- 틀리지 않고 안전하게 대답을 하는 모델에게 높은 점수를 주고 싶다면 precision의 비중을 높여야 한다.
  - 하지만, '낰좈앜'같이 'happy' 카테고리에 속하는 문장들에게 아예 추천을 포기할 수 있다.
- 틀릴 수 있지만 어찌 됐든 대답을 하는 모델에게 높은 점수를 주고 싶다면 recall의 비중을 높여야 한다.
  - 하지만, '제길'같이 'happy' 카테고리에 속하지 않는 문장들도 'happy'로 추천할 수 있다.

---

3. Multi-Label Metrics

[문장-카테고리]의 관계는 1:1 관계가 아니라 1:N 관계로 생각할 수 있다. 주어진 문장에 대해서 여러가지 감정 태그가 부여되고, 이를 맞추는 문제가 된다면 이는 multi-label 문제라고 할 수 있다. 하나의 데이터에 대해 Multi-label 모델의 성능을 평가하는 방식은 다음과 같은게 있다.

3-1. Precision At K, Recall At K

- Precision at K, P@K: proportion of top-K categories that are relevent.

$$P@K = \frac{\text{top-K에 속한 데이터와 관련있는 카테고리의 수}}{K}$$

- Recall at K, R@K: proportion of relevant documents that in the top-K

$$R@K = \frac{\text{top-K에 속한 데이터와 관련있는 카테고리의 수}}{\text{데이터와 관련있는 카테고리의 수}}$$

그림 p16

모델의 성능을 평가할 때, 데이터의 문장 별 P@K, R@K의 평균을 구해 사용할 수 있다.

- P@K와 R@K의 장단점
  - 장점
    - 쉽게 계산 가능하다.
    - 쉽게 해석이 가능하다.
  - 단점
    - K의 값이 metric의 큰 영향을 미친다.
    - 적절한 K를 고르기 어렵다.
    - K 이내의 Ranking은 중요하게 여겨지지 않는다.

3-2. AP: Average Precision

이상적으로 우리는 K의 다양한 값에 높은 precision을 원하는데, metric 중 하나인 average precision은 K를 선택하지 않고 precision과 recall을 반영할 수 있다.

AP는 다음과 같이 구할 수 있다.

1. Ranking을 위에서부터 하나 씩 내려온다.
2. 만약 t 번째 category가 주어진 문장과 관계가 있다면, P@K를 구한다.
3. 1-2번을 반복하다고, R@K가 1이 되면 구한 P@K의 평균을 구한다.

그림 p34

- AP의 장단점
  - 장점
    - K를 선택하지 않아도 된다.
    - precision과 recall을 모두 고려할 수 있다.
    - Ranking의 윗부분에서의 실수가 높은 영향끼친다.

        - Ranking의 아랫부분에서의 실수도 영향을 끼친다.
    - 단점
        - P@K, R@K를 해석하기 쉽지 않다.

3-3. MAP: Mean Average Precision

위에 서술한 AP의 개념은 한 문장에 대한 AP를 구하는 방법이다. 모든 문장에 대해 AP를 구하고 평균을 내면 이를 모델 metric으로서 사용할 수 있다. 이를 MAP(Mean Average Precision)이라고 하며, information retrieval 시스템을 평가할 때, 가장 많이 쓰이는 metric 중 하나이다.