아마존 구매리뷰를 통한 제품 만족도 평가

빅데이터경영통계전공
20162522 송민규

INDEX

Summary

아마존 구매내역으로 만든 데이터셋을 이용하여 분석하였다. 핸드폰 구매내역, 악기 구매내역 데이터를 가져왔으며, 여러가지 모델들을 적용하여 감성분석을 실시해 보았다. 그 결과, 핸드폰 구매내역 데이터 에서는 다층신경망 모형이 가장 정확도가 높게 나왔으며 , 악기 구매내역 데이터에서는 tensorflow 모형이 가장 정확도가 높게 나왔다. 이로써 데이터마다 적용시켜야 하는 모델이 다른 것을 알게 되었고, 성능을 올리기 위하여 여러가지 시도를 함으로써 조금이나마 성능을 올릴 수 있는 방법을 찾아내었다.

Introduction

최근 소비자들은 직접 가서 물품을 구매하기 보다 온라인으로 쇼핑하고 배송하는 시스템을 선호한다. 보통 소비자들이 물품을 구매할 때 상품 정보만 보고 사기에는 정보가 부족하여 구매자의 리뷰를 보게 되는데, 이 과정이 물품 구매에 큰 영향을 끼친다. 그래서 상품 리뷰 중 추천 리뷰와 비추천 리뷰를 자동으로 구분하여 따로따로 보여준다면 소비자들이 상품을 구매할 때 더 나은 판단을 할 수 있다. 리뷰가 추천/비추천인지 파악하여 구분 할 수 있는 모델을 여러 개 만들어보고 성능이 좋고 시스템에 활용할 수 있는 모델을 선정해 볼 것이다.

Main Subject – Data

- Source : http://jmcauley.ucsd.edu/data/amazon/
- Preprocessing : column이 reviewerID, asin, reviewerName, helpful, reviewText, overall, summary, unixReviewTime, reviewTime으로 구성 되어 있어 필요한 column인 reviewText, overall만 추출했다. 그 후 비교를 위한 sentiment column을 생성하였고 overall column은 drop하였다.
- About Data : May 1996 - July 2014까지의 data이며, 같은 제품에 대하여 최소 5개 이상의 리뷰가 남겨져 있다.

Main Subject – tensorflow 감성분석

핸드폰 구매내역
(Cell)

```
38868/38868 [==============================] - 1s 18us/sample - loss: 0.3559 - accur
acy: 0.8544
Out[24]: [0.35592715552747034, 0.8544304]
```

악기 구매내역의 accuracy가 더 높음.

악기 구매내역
(musical)

```
2051/2051 [==============================] - 0s 36us/sample - loss: 0.3025 - accurac
y: 0.8903
Out[24]: [0.3025156772141047, 0.8902974]
```

성능높이기 위한 시도

Optimizer =
RMSprop 변경

```
2051/2051 [==============================] - 0s 37us/sample - loss: 0.3297 - accurac
y: 0.8874
Out[24]: [0.32973978485728633, 0.887372]
```

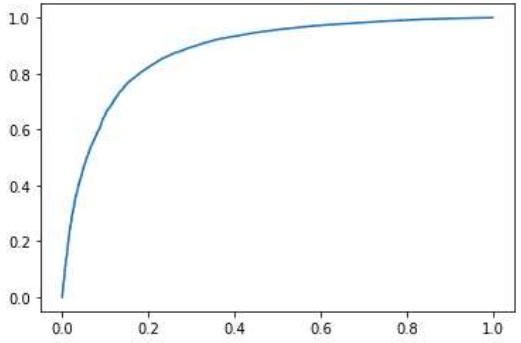
Optimizer =
nadam 변경

```
2051/2051 [==============================] - 0s 36us/sample - loss: 0.3015 - accurac
y: 0.8908
Out[49]: [0.301541697381038, 0.890785]
```

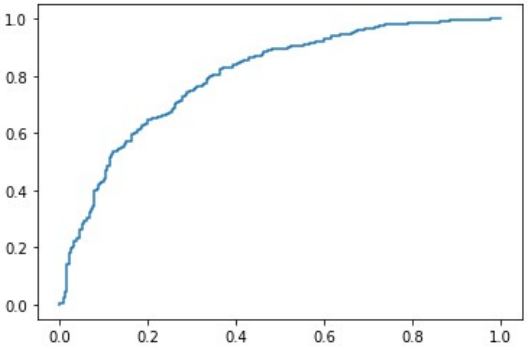
Nadam으로 변경 후에 성능이 조금 향상됨.

Main Subject – confusion matrix

Cell – accuracy_score : 0.85443037
precision_score : 0.87395114
recall_score : 0.94676818
f1_score : 0.90890355
ROC_AUC_score : 0.88250593



Musical – accuracy_score : 0.89029741
precision_score : 0.8965
recall_score : 0.99006073
f1_score : 0.94096037
ROC_AUC_score : 0.79843778



다른 모델과의 비교를 위해 accuracy_score를 보는 것으로 결정.

Main Subject – 다층신경망

Main Subject – 신경망 언어모형

핸드폰 구매내역
(Cell)

```
Train on 194340 samples
194340/194340 [==============================] - 75s 386us/sample - loss: 0.5467 - a
cc: 0.7641
```

악기 구매내역의 accuracy가 더 높음.

악기 구매내역
(musical)

```
Train on 10254 samples
10254/10254 [==============================] - 2s 211us/sample - loss: 0.4477 - acc:
0.8792
```

성능높이기 위한 시도

Dense(32),
Dense(16) 추가

```
Train on 10254 samples
10254/10254 [==============================] - 2s 202us/sample - loss: 0.4275 - acc:
0.8783
```

Dense를 두 층을 쌓았으나 성능이 약간 떨어짐.

Main Subject – FastText

성능높이기 위한 시도

핸드폰 구매내역 (Cell)

```
Train on 1000 samples
1000/1000 [==============================] - 0s 199us/sample - loss: 0.6597 - accura
cy: 0.6340
```

악기 구매내역 (musical)

```
Train on 1000 samples
1000/1000 [==============================] - 0s 202us/sample - loss: 0.5302 - accura
cy: 0.8130
```

악기 구매내역의 accuracy가 더 높음.

여러 번 잘 나올 때까지 시도

```
Train on 1000 samples
1000/1000 [==============================] - 1s 521us/sample - loss: 0.6131 - accura
cy: 0.6820
```

여러 번 잘 나올 때까지 시도

```
Train on 1000 samples
1000/1000 [==============================] - 0s 203us/sample - loss: 0.4721 - accura
cy: 0.8360
```

모델을 실행할 때마다 결과값이 바뀜.

Main Subject – 합성곱 신경망

핸드폰 구매내역
(Cell)

```
Train on 155472 samples
155472/155472 [==============================] - 145s 934us/sample - loss: 0.3959 -
accuracy: 0.8255
```

악기 구매내역
(musical)

```
Train on 8203 samples
8203/8203 [==============================] - 4s 500us/sample - loss: 0.3915 - accura
cy: 0.8721
```

악기 구매내역의 accuracy가 더 높음.

Main Subject – BERT 분석

성능높이기 위한 시도

핸드폰 구매내역 (Cell)

Out [12]: 0.7722772277227723

악기 구매내역 (musical)

Out [12]: 0.8118811881188119

→ 악기 구매내역의 accuracy가 더 높음.

학습 data 개수 500개로 증가

Out [14]: 0.780439121756487

→ 학습 data 개수 증가하여 성능이 향상됨.

Main Subject - Result

Cell – tensorflow : 0.8544304
다층신경망 : 0.867526 -> 0.85268086 (Hyperopt 실행)
신경망 언어모형 : 0.7641
FastText : 0.6340 -> 0.6820 (모델 여러 번 실행)
합성곱 신경망 : 0.8255
BERT : 0.77227722 -> 0.78043912 (학습데이터 증가)

Musical – tensorflow : 0.8902974 -> 0.890785 (Optimizer = nadam 변경)
-> 0.887372 (Optimizer = RMSprop 변경)
다층신경망 : 0.887372 -> 0.8620185 (Hyperopt 실행)
신경망 언어모형 : 0.8792 -> 0.8783 (Dense 추가로 쌓기)
FastText : 0.8130 -> 0.8360 (모델 여러 번 실행)
합성곱 신경망 : 0.8721
BERT : 0.81188118

결과적으로 성능 향상은 밑줄 친 부분에서 이뤄내었다.

Main Subject - Result

Cell – tensorflow : 0.8544304
- 다층신경망 : 0.867526 -> 0.85268086 (Hyperopt 실행)
- 신경망 언어모형 : 0.7641
- FastText : 0.6340 -> 0.6820 (모델 여러 번 실행)
- 합성곱 신경망 : 0.8255
- BERT : 0.77227722 -> 0.78043912 (학습데이터 증가)

Musical – tensorflow : 0.8902974 -> 0.890785 (Optimizer = nadam 변경)
-> 0.887372 (Optimizer = RMSprop 변경)
- 다층신경망 : 0.887372 -> 0.8620185 (Hyperopt 실행)
- 신경망 언어모형 : 0.8792 -> 0.8783 (Dense 추가로 쌓기)
- FastText : 0.8130 -> 0.8360 (모델 여러 번 실행)
- 합성곱 신경망 : 0.8721
- BERT : 0.81188118

전체적으로 data개수가 많은 Cell data보다 data 개수가 적은 Musical data에서 성능이 더 높았다.

Cell에서는 다층신경망을 사용한 모델이 가장 좋은 성능, Musical 에서는 tensorflow를 사용한 모델이 가장 좋은 성능을 가졌다.

Conclusion

- 분석 결과를 바탕으로 핸드폰 데이터는 다층신경망을 이용해 모델링을 하고, 악기 데이터는 tensorflow를 이용해 모델링을 하여 더 추가 되는 데이터들을 분석하면 가장 좋은 결과를 얻을 수 있을 것이다. 추가되는 리뷰들에 대해 추천/비추천 으로 분류시킨다면 상품을 구매하려는 소비자들이 구매 결정을 하는데 도움을 줄 수 있다고 생각한다.

Conclusion

- 이 분석을 통해 Data에 따라 성능이 잘 나오는 모델들이 다른 것을 알 수 있었다. 그리고 성능 향상을 위해 노력해보았는데, 학습이 잘 안되는 문제점을 보완하기 위해 될 때까지 해보는 방법을 사용했다. 또한 학습시키는 데이터를 증가시켜 overfitting 문제를 조금이나마 제거했고, Optimizer같은 parameter를 조정해보았다.
- 아쉬운 점은, 조금 더 성능 향상을 위해 시도해 볼만한 activation parameter 조정과, dropout이나 batchnormalization을 추가하여 overfitting 문제를 더 막을 수 있었다고 생각한다. 또한 accuracy_score를 이용해 성능을 측정하였는데 accuracy보다는 roc_auc_score를 사용하는 것이 더 좋은 모델을 평가하는데에 도움이 되었을 것이다.

References

- About confusion matrix : https://towardsdatascience.com/understanding-data-science-classification-metrics-in-scikit-learn-in-python-3bc336865019
- About FastText : https://lovit.github.io/nlp/representation/2018/10/22/fasttext_subword/

Thank You