CBR / AR / CF

데이터마이닝이론및응용
에이쁠원하조

2020147024 김우영
2020251009 김혜리
2020195053 노가원
2020147018 조윤영

목차

데이터 및 변수

CBR - categorical target

Banking Dataset - Marketing Targets | Kaggle

1 - age (numeric)
2 - job : type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student", "blue-collar","self-employed","retired","technician","services")
3 - marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)
4 - education (categorical: "unknown","secondary","primary","tertiary")
5 - default: has credit in default? (binary: "yes","no")
6 - balance: average yearly balance, in euros (numeric)
7 - housing: has housing loan? (binary: "yes","no")
8 - loan: has personal loan? (binary: "yes","no")
related with the last contact of the current campaign:
9 - contact: contact communication type (categorical: "unknown","telephone","cellular")
10 - day: last contact day of the month (numeric)
11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")
12 - duration: last contact duration, in seconds (numeric)
other attributes:
13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)
15 - previous: number of contacts performed before this campaign and for this client (numeric)
16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")
17 - y - has the client subscribed a term deposit? (binary:

CBR - continuous target

50 Startups Data | Kaggle

R&D spend: amount the startup spends on research and development

Administration: amount they spend on administration cost

Marketing spend:amount the startup spend on marketing

state :which state the startup is based in

profit: the profit made by the startup($)

AR

https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis

BillNo: 6-digit number assigned to each transaction. Nominal.

Itemname: Product name. Nominal.

Quantity: The quantities of each product per transaction. Numeric.

Date: The day and time when each transaction was generated. Numeric.

Price: Product price. Numeric.

CustomerID: 5-digit number assigned to each customer. Nominal.

Country: Name of the country where each customer resides. Nominal.

CF

https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis

user_id : 사용자 번호

book_id : 도서 번호

rating : 1~5 까지의 점수

books.xlsx : book_id에 따른 도서 제목(original_title)

Reference

KC, Manish. “Startup Success Prediction.” *Kaggle*, 16 Sept. 2020, https://www.kaggle.com/datasets/manishkc06/startup-success-prediction.
“Market Basket Analysis.” Kaggle, https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis.
Rathi, Prakhar. “Banking Dataset - Marketing Targets.” Kaggle, 19 Oct. 2020, https://www.kaggle.com/datasets/prakharrathi25/banking-dataset-marketing-targets.
Zygmuntz. “Zygmuntz/Goodbooks-10K: Ten Thousand Books, Six Million Ratings.” GitHub, https://github.com/zygmuntz/goodbooks-10k.

CBR - categorical target (classification)

data : banking dataset - marketing targets

x.describe()

	age	duration	campaign	pdays	previous
count	32950.000000	32950.000000	32950.000000	32950.000000	32950.000000
mean	40.014112	258.127466	2.560607	962.052413	0.174719
std	10.403636	258.975917	2.752326	187.951096	0.499025
min	17.000000	0.000000	1.000000	0.000000	0.000000
25%	32.000000	103.000000	1.000000	999.000000	0.000000
50%	38.000000	180.000000	2.000000	999.000000	0.000000
75%	47.000000	319.000000	3.000000	999.000000	0.000000
max	98.000000	4918.000000	56.000000	999.000000	7.000000

각 변수마다 범위 차이가 많이 나므로 scaling 진행

scaled_x.describe() # 표준화 이후, 독립변수(feature)의 기초통계량

	age	duration	campaign	pdays	previous
count	3.295000e+04	3.295000e+04	3.295000e+04	3.295000e+04	3.295000e+04
mean	2.510620e-16	-6.879003e-17	-5.692968e-17	-1.898734e-16	-2.266944e-17
std	1.000015e+00	1.000015e+00	1.000015e+00	1.000015e+00	1.000015e+00
min	-2.212156e+00	-9.967389e-01	-5.670225e-01	-5.118709e+00	-3.501269e-01
25%	-7.703301e-01	-5.990125e-01	-5.670225e-01	1.965838e-01	-3.501269e-01
50%	-1.935999e-01	-3.016831e-01	-2.036879e-01	1.965838e-01	-3.501269e-01
75%	6.714954e-01	2.350545e-01	1.596467e-01	1.965838e-01	-3.501269e-01
max	5.573702e+00	1.799373e+01	1.941638e+01	1.965838e-01	1.367745e+01

scaling이 진행된후 변수간 범위 차이가 줄어든것을 알 수 있다

```
print(y['target_y'].value_counts())

0    29238
1     3712
Name: target_y, dtype: int64
```

y값의 분포가 unbalanced하다

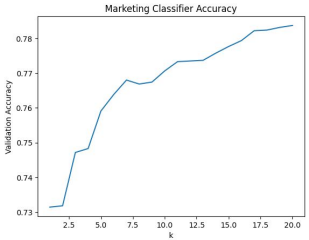
```
print(y_resampled['target_y'].value_counts())

0    2376
1    2376
Name: target_y, dtype: int64
```

data set을 split한 후 y값의 분포를 balanced하게 맞춰줌

KNN classification

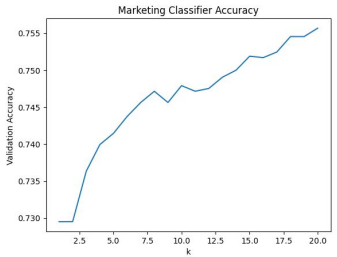
1. distance = euclidean



accurcy

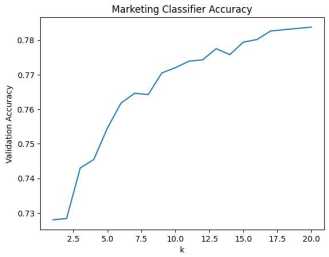
0.7916540212443096

2. distance = cosine



0.7596358118361153

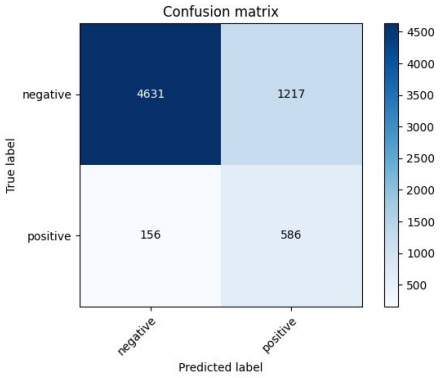
3. distance = mahalanobis



0.791350531107739

=> 유클리드 거리 계산 방식 채택

Evaluation




```
Accuracy: 0.7916540212443096
Precision: 0.3250138657792568
Recall: 0.7897574123989218
F1 score: 0.4605108055009823
```

- **accuracy** (정확도): 전체 중 모델이 바르게 분류한 비율로 0.792 로 꽤 높다다
- **precision** (정밀도): 모델이 Positive라 분류한 것 중 실제값이 Positive인 비율로 0.325로 다소 낮다
- **recall** (재현도): 실제값이 Positive인 것 중 모델이 Positive라 분류한 비율로 0.790로 꽤 높다
- **f-1**: Precision과 Recall의 조화평균으로 0.461로 크게 높지 않다

f-1값에 따라 모델의 성능이 크게 좋다고 할 수 없다

CBR - continuous target (regression)

data설명 (data: start-up data)

```
df.describe()
```

	R&D Spend	Administration	Marketing Spend	Profit
count	50.000000	50.000000	50.000000	50.000000
mean	73721.615600	121344.639600	211025.097800	112012.639200
std	45902.256482	28017.802755	122290.310726	40306.180338
min	0.000000	51283.140000	0.000000	14681.400000
25%	39936.370000	103730.875000	129300.132500	90138.902500
50%	73051.080000	122699.795000	212716.240000	107978.190000
75%	101602.800000	144842.180000	299469.085000	139765.977500
max	165349.200000	182645.560000	471784.100000	192261.830000

KNN regression

K=1~20까지 나타낸 그래프이다
MSE가 가장 작은 지점인 K=9가 적절한 선택으로 보여진다

```
regressor.fit(x_train, y_train)
```

```
KNeighborsRegressor(n_neighbors=9, weights='distance')
```

neighbor=9이다

Evaluation

```
MSE : 296474517.8419266
RMSE: 17218.43540632907
MAE : 13312.95866968669
MAPE: 0.21775031279172735
```

MSE(Mean Squared Error)는 예측값과 실제값의 차이를 제곱하여 평균을 구한 값으로, 값이 작을수록 모델의 성능이 좋다고 평가됨. 이 모델에서 MSE는 296474517.8419266로, 오차의 크기가 크기 때문에 예측력이 낮다고 볼 수 있음

RMSE(Root Mean Squared Error)는 MSE에 루트를 씌운 값으로, 값이 작을수록 모델의 성능이 좋다고 평가됨. 이 모델에서 RMSE가 17218.43540632907으로, MSE와 비슷하게 예측력이 낮다고 평가할 수 있음

MAE(Mean Absolute Error)는 예측값과 실제값의 차이를 절대값으로 구한 후, 평균을 구한 값으로, 값이 작을수록 모델의 성능이 좋다고 평가됨. 이 모델에서 MAE가 13312.95866968669으로, MSE나 RMSE보다는 예측력이 나은 편이지만, 여전히 아주 좋은 것은 아님

MAPE(Mean Absolute Percentage Error)는 MAE를 실제값으로 나눈 비율을 백분율로 나타낸 값으로, 값이 작을수록 모델의 성능이 좋다고 평가됨. 이 모델에서 MAPE가 0.21775031279172735으로, 상대적으로 예측력이 좋은 편임

따라서, 이 모델의 평가 결과를 종합적으로 판단하면 예측력이 낮으며, 모델을 개선할 필요가 있음

AR

Data/data.describe

```
data
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01.12.2010 08:26	2,55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	01.12.2010 08:26	2,75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01.12.2010 08:26	3,39	17850.0	United Kingdom
...	...	...	...	...	...	...	...
522059	581587	PACK OF 20 SPACEBOY NAPKINS	12	09.12.2011 12:50	0,85	12680.0	France
522060	581587	CHILDREN'S APRON DOLLY GIRL	6	09.12.2011 12:50	2,1	12680.0	France
522061	581587	CHILDRENS CUTLERY DOLLY GIRL	4	09.12.2011 12:50	4,15	12680.0	France
522062	581587	CHILDRENS CUTLERY CIRCUS PARADE	4	09.12.2011 12:50	4,15	12680.0	France
522063	581587	BAKING SET 9 PIECE RETROSPOT	3	09.12.2011 12:50	4,95	12680.0	France

522064 rows × 7 columns

```
data.describe()
```

	Quantity	CustomerID
count	522064.000000	388023.000000
mean	10.090435	15316.931710
std	161.110525	1721.846964
min	-9600.000000	12346.000000
25%	1.000000	13950.000000
50%	3.000000	15265.000000
75%	10.000000	16837.000000
max	80995.000000	18287.000000

Preprocess

```
data_tmp = data.groupby('CustomerID')['Itemname'].apply(set).apply(list).to_list()
print(data_tmp[:10])
```

```
[['MEDIUM CERAMIC TOP STORAGE JAR'], ['MINI PAINT SET VINTAGE', 'CHRISTMAS METAL TAGS
```

```
te = TransactionEncoder()
te_ary = te.fit_transform(data_tmp) # df_tmp를 장바구니 형태로 변경 (fit과 transform을 동시에 진행)
print(te_ary)
```

```
[[False False False ... False False False]
 [False False False ... False False False]
 [False False False ... False False False]
 ...
 [False False False ... False False False]
 [ True False False ... False False False]
 [False False False ... False False False]]
```

1) DataFrame을 nested list(장바구니) 형태로 변경

- CustomerID가 동일한 Itemname을 하나의 리스트로 묶음

```
df = pd.DataFrame(te_ary, columns=te.columns_)
```

	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 DAISY PEGS IN WOOD BOX	12 EGG HOUSE PAINTED WOOD	12 HANGING EGGS HAND PAINTED	12 IVORY ROSE PEG PLACE SETTINGS	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	...	ZINC STAR T-LIGHT HOLDER	ZINC SWEETHEART SOAP DISH	SWEI
0	False	False	False	False	False	False	False	False	False	False	...	False	False	
1	False	False	False	False	False	False	False	False	False	False	...	False	False	
2	False	False	False	False	False	False	False	False	False	False	...	False	False	
3	False	False	False	False	False	False	False	False	False	False	...	False	False	
4	False	False	False	False	False	False	False	False	False	False	...	False	False	

5 rows × 3846 columns

2)fit과 transform을 동시에 진행하여

nested list (장바구니)를 association rule을 찾기 위한 dataframe의 형태로 변경

1. Apriori Algorithm

```
frequent_itemsets = apriori(df, min_support=0.07, use_colnames=True) # 여기서는 0.07를 기준으로 선정
frequent_itemsets.sort_values(by='support', ascending=False)
```

	support	itemsets
85	0.202234	(REGENCY CAKESTAND 3 TIER)
109	0.196416	(WHITE HANGING HEART T-LIGHT HOLDER)
72	0.162439	(PARTY BUNTING)
7	0.156621	(ASSORTED COLOUR BIRD ORNAMENT)
94	0.147079	(SET OF 3 CAKE TINS PANTRY DESIGN)
...	...	...
73	0.070980	(PINK CREAM FELT CRAFT TRINKET BOX)
44	0.070514	(JUMBO BAG PINK VINTAGE PAISLEY)
9	0.070282	(BLUE HARMONICA IN BOX)
115	0.070049	(WORLD WAR 2 GLIDERS ASSTD DESIGNS)
29	0.070049	(HANGING HEART JAR T-LIGHT HOLDER)

123 rows × 2 columns

구매량이 큰 품목 중 다른 품목과 같이 구매한 경우가 많은 품목

- REGENCY CAKESTAND 3 TIER (support = 0.202234)
- GREEN REGENCY TEACUP AND SAUCER (support = 0.088667)
- ROSES REGENCY TEACUP AND SAUCER (support=0.0970444)

- GREEN REGENCY TEACUP AND SAUCER', 'ROSES REGENCY TEACUP AND SAUCER'를 같이 구매한 경우가 많다. (support = 0.073307)
- REGENCY CAKESTAND 3 TIER와 ROSES REGENCY TEACUP AND SAUCER를 같이 구매한 경우가 많다. (support = 0.073307)

2. Confidence

```
association_rules(frequent_itemsets, metric="confidence", min_threshold=0.1).sort_values(by = ['confidence', 'lift', 'support'], ascending=False)
# 최소 confidence를 0.1로 지정
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
6	(GREEN REGENCY TEACUP AND SAUCER)	(ROSES REGENCY TEACUP AND SAUCER)	0.088667	0.097044	0.073307	0.826772	8.519515	0.064702	5.212516
8	(RED HANGING HEART T-LIGHT HOLDER)	(WHITE HANGING HEART T-LIGHT HOLDER)	0.088201	0.196416	0.071212	0.807388	4.110590	0.053888	4.173031
1	(ROSES REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.097044	0.088667	0.073307	0.755396	8.519515	0.064702	3.735748
11	(ROSES REGENCY TEACUP AND SAUCER)	(REGENCY CAKESTAND 3 TIER)	0.097044	0.202234	0.073307	0.755396	3.735253	0.053661	3.261455
7	(PAPER CHAIN KIT VINTAGE CHRISTMAS)	(PAPER CHAIN KIT 50'S CHRISTMAS)	0.108660	0.141727	0.075401	0.693756	4.895265	0.059999	2.802852
5	(LUNCH BAG CARS BLUE)	(LUNCH BAG RED RETROSPOT)	0.110542	0.120411	0.070980	0.642105	5.245487	0.057448	2.452087
2	(HEART OF WICKER LARGE)	(HEART OF WICKER SMALL)	0.112171	0.132185	0.070980	0.632760	4.787070	0.056152	2.363202
4	(LUNCH BAG RED RETROSPOT)	(LUNCH BAG CARS BLUE)	0.120411	0.110542	0.070980	0.589548	5.245487	0.057448	2.116990
3	(HEART OF WICKER SMALL)	(HEART OF WICKER LARGE)	0.132185	0.112171	0.070980	0.536972	4.787070	0.056152	1.917440
6	(PAPER CHAIN KIT 50'S CHRISTMAS)	(PAPER CHAIN KIT VINTAGE CHRISTMAS)	0.141727	0.108660	0.075401	0.532020	4.895265	0.059999	1.904609
9	(WHITE HANGING HEART T-LIGHT HOLDER)	(RED HANGING HEART T-LIGHT HOLDER)	0.196416	0.088201	0.071212	0.362559	4.110590	0.053888	1.429406
10	(REGENCY CAKESTAND 3 TIER)	(ROSES REGENCY TEACUP AND SAUCER)	0.202234	0.097044	0.073307	0.362486	3.735253	0.053661	1.416369

- GREEN REGENCY TEACUP AND SAUCER을 산 사람이 ROSES REGENCY TEACUP AND SAUCER을 살 확률 (confidence = 0.826772) > ROSES REGENCY TEACUP AND SAUCER을 산 사람이 GREEN REGENCY TEACUP AND SAUCER을 살 확률 (confidence = 0.755396)
- ROSES REGENCY TEACUP AND SAUCER을 산 사람이 REGENCY CAKESTAND 3 TIER을 살 확률 (confidence = 0.755396) > REGENCY CAKESTAND 3 TIER을 산 사람이 ROSES REGENCY TEACUP AND SAUCER을 살 확률 (confidence = 0.362486)

3. Lift

```
association_rules(frequent_itemsets, metric="lift", min_threshold=1).sort_values(by = ['lift', 'confidence', 'support'], ascending=False)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
6	(GREEN REGENCY TEACUP AND SAUCER)	(ROSES REGENCY TEACUP AND SAUCER)	0.088667	0.097044	0.073307	0.826772	8.519515	0.064702	5.212516
1	(ROSES REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.097044	0.088667	0.073307	0.755396	8.519515	0.064702	3.735748
5	(LUNCH BAG CARS BLUE)	(LUNCH BAG RED RETROSPOT)	0.110542	0.120411	0.070980	0.642105	5.245487	0.057448	2.452087
4	(LUNCH BAG RED RETROSPOT)	(LUNCH BAG CARS BLUE)	0.120411	0.110542	0.070980	0.589548	5.245487	0.057448	2.116990
7	(PAPER CHAIN KIT VINTAGE CHRISTMAS)	(PAPER CHAIN KIT 50'S CHRISTMAS)	0.108660	0.141727	0.075401	0.693756	4.895265	0.059999	2.802852
6	(PAPER CHAIN KIT 50'S CHRISTMAS)	(PAPER CHAIN KIT VINTAGE CHRISTMAS)	0.141727	0.108660	0.075401	0.532020	4.895265	0.059999	1.904609
2	(HEART OF WICKER LARGE)	(HEART OF WICKER SMALL)	0.112171	0.132185	0.070980	0.632760	4.787070	0.056152	2.363202
3	(HEART OF WICKER SMALL)	(HEART OF WICKER LARGE)	0.132185	0.112171	0.070980	0.536972	4.787070	0.056152	1.917440
8	(RED HANGING HEART T-LIGHT HOLDER)	(WHITE HANGING HEART T-LIGHT HOLDER)	0.088201	0.196416	0.071212	0.807388	4.110590	0.053888	4.173031
9	(WHITE HANGING HEART T-LIGHT HOLDER)	(RED HANGING HEART T-LIGHT HOLDER)	0.196416	0.088201	0.071212	0.362559	4.110590	0.053888	1.429406
11	(ROSES REGENCY TEACUP AND SAUCER)	(REGENCY CAKESTAND 3 TIER)	0.097044	0.202234	0.073307	0.755396	3.735253	0.053661	3.261455
10	(REGENCY CAKESTAND 3 TIER)	(ROSES REGENCY TEACUP AND SAUCER)	0.202234	0.097044	0.073307	0.362486	3.735253	0.053661	1.416369

- GREEN REGENCY TEACUP AND SAUCER을 산 사람이 ROSES REGENCY TEACUP AND SAUCER을 살 확률에는 매우 큰 양의 상관관계가 있음 (lift = 8.519515)
- REGENCY CAKESTAND 3 TIER을 산 사람이 ROSES REGENCY TEACUP AND SAUCER을 살 확률에도 큰 양의 상관관계가 있음 (lift = 3.735253)

CF

```
[8] from sklearn.metrics.pairwise import cosine_similarity
    book_sim = cosine_similarity(df_scaled, df_scaled)
    # item by item 으로 구성된 행렬
    print(book_sim.shape)

    (7774, 7774)

[9] book_sim = pd.DataFrame(book_sim, index = df_scaled.index, columns = df_scaled.index)

    book_sim.head()
```

book_id	1	2	3	4	5	6	7	8	9	10	...	9990	9991	9993	9994	9995	9996	9997	9998	9999	10000
book_id																					
1	1.000000	0.141213	-0.029434	0.074984	-0.031170	0.024693	-0.001886	-0.038442	-0.002494	0.086905	...	-0.003820	0.001899	0.006276	0.016715	0.000849	0.011151	0.000000	0.000000	0.000164	0.001537
2	0.141213	1.000000	-0.074004	0.102229	-0.038082	0.018374	0.064416	-0.059612	0.010517	0.113843	...	0.006319	-0.003999	0.000000	0.000060	0.001631	0.003054	-0.006155	0.006663	-0.010687	-0.002263
3	-0.029434	-0.074004	1.000000	-0.110276	0.000193	-0.017613	-0.050501	-0.004163	0.068082	-0.018559	...	0.002032	-0.003466	0.000000	0.004407	0.002926	-0.002422	-0.007791	-0.007283	-0.003395	0.000000
4	0.074984	0.102229	-0.110276	1.000000	0.081551	-0.010110	0.052522	0.075303	-0.084975	0.155764	...	-0.000407	0.013118	-0.003396	-0.002474	-0.024819	0.004485	0.009231	0.008669	-0.007729	0.013907
5	-0.031170	-0.038082	0.000193	0.081551	1.000000	0.000000	-0.047001	0.173033	-0.014163	0.027118	...	-0.005163	-0.016530	0.000000	0.000071	0.018988	-0.006368	-0.009829	0.009459	-0.001555	0.012372

5 rows × 7774 columns

유사도 평가 결과는 위와 같다. 각 도서가 유사한 정도를 표로 확인할 수 있다. 수치가 1에 가까울 수록 유사도가 높다.

```
book_sim[2].sort_values(ascending=False)[:10]

book_id
2      1.000000
18     0.523581
23     0.523137
24     0.504721
27     0.452843
25     0.415714
21     0.409587
1      0.141213
10     0.113843
422    0.111169
Name: 2, dtype: float64
```

평가의 성능을 간략히 알아보기 위해 <Harry Potter and the Philosopher's Stone>와 유사한 도서들을 출력해 보았다. 출력된 결과는 왼쪽과 같다. 결과를 보면 출력된 도서의 Id는 18, 23, 24, 27, 25, 21에 해당하고 모두 0.4 이상의 유사값이 나왔는데, 이 도서들은 모두 Harry Potter 시리즈에 속하는 도서로 결과의 성능을 간략히 확인해볼 수 있었다.