Service2

Review  **Rating**

Required Services

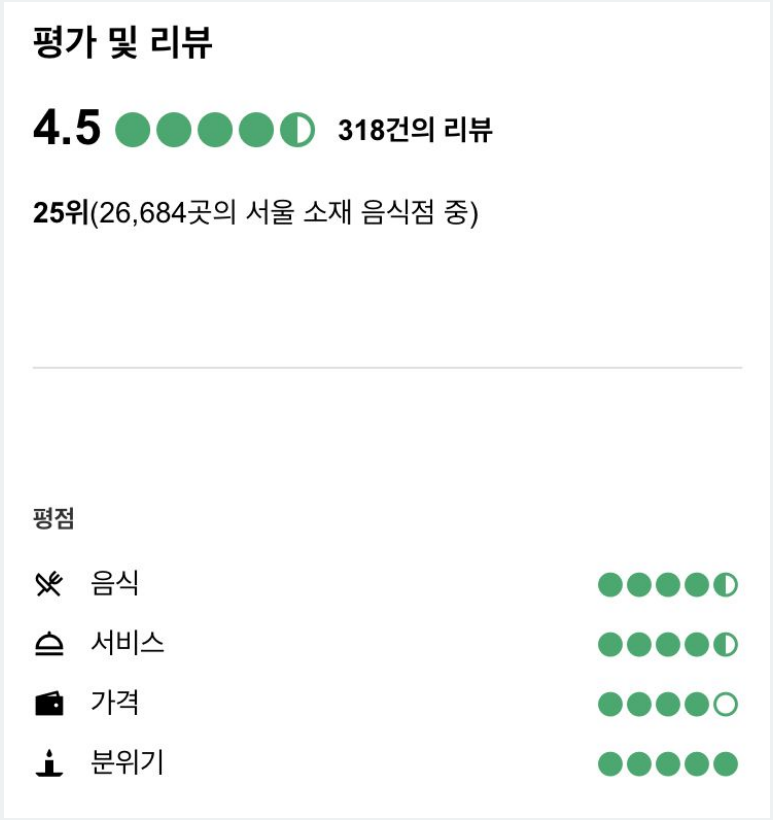
1. Crawling Data from Web
   => (Implemented)
1. Split KNU Sentiment Dictionary
   => (Canceled)
1. Sentiment analysis using KNU Sentiment Dictionary
   => (Implemented)
1. Data Preprocessing
   => (Implemented)
1. AI Modeling
   => (Implemented)
1. Comparison with paper
   => (Implemented)
1. Improvement Direction
   => (2nd semester)

Required Services

1. Crawling Data from Web
   => (Implemented)
1. Split KNU Sentiment Dictionary
   => (Canceled)
1. Sentiment analysis using KNU Sentiment Dictionary
   => (Implemented)
1. Data Preprocessing
   => (Implemented)
1. AI Modeling
   => (Implemented)
1. Comparison with paper
   => (Implemented)
1. Improvement Direction
   => (2nd semester)

1. Crawling Data from Web
   - Crawling reviews from restaurants with about 100 reviews

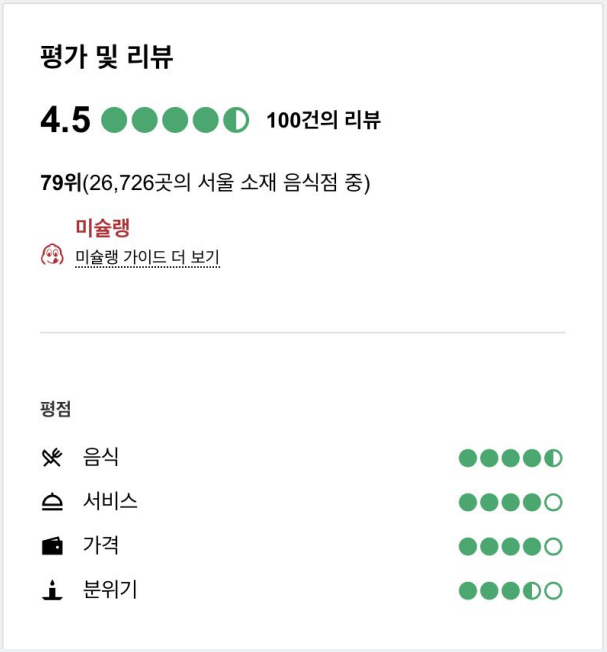
1. Crawling Data from Web



Using Library: Selenium4
Using Tool: Chrome Driver

Data was imported from a site called Tripadvisor.
The data is based on the restaurant's overall rating, the number of reviews, and each criterion (food, service, price, atmosphere). The tools used are selenium and chrome driver.

1. Crawling Data from Web



This code is part of the code that gets ratings. The output was saved as follows.
[Overall, food, service, price, mood]

```
driver.get(url)
...
results['restaurant_ratings'].append(rating)

    # 기준 별 평점 구하기
    rating_elements = driver.find_elements(By.XPATH, "//span[@class='vzATR']")

    # 각 기준별 평점 출력하기
    for rating_element in rating_elements:
        rating = rating_element.find_element(By.CSS_SELECTOR,
'.ui_bubble_rating').get_attribute('class')
        rating = getIntRating(rating)
        results['restaurant_ratings'].append(rating)
```

```
Output:
"restaurant_ratings": [
        40, 40, 35, 40, 35
    ]
```

1. Crawling Data from Web

With output, the review was saved as follows.
{“Rating”: 30, “Text”: “...”}, …

```
# 각 리뷰의 정보 추출
for review in reviews:
    rating = review.find_element(By.CSS_SELECTOR,
        '.ui_bubble_rating').get_attribute('class')
    rating = getIntRating(rating)
    review_text = review.find_element(By.CLASS_NAME,
        'partial_entry').text
        …
```

```
Output:
{
          "Rating": 30,
          "Text": "부산에서 가장 잘 알려진 초기 수제 맥주 브랜드 중 하나. 그만큼 대체로 양식화된 프랜차이즈 같은 느낌을 준다. 그래도 나쁘지 않음."
       }, …
```

Required Services

1. Crawling Data from Web
   => (Implemented)
1. Split KNU Sentiment Dictionary
   => (Canceled)
1. Sentiment analysis using KNU Sentiment Dictionary
   => (Implemented)
1. Data Preprocessing
   => (Implemented)
1. AI Modeling
   => (Implemented)
1. Comparison with paper
   => (Implemented)
1. Improvement Direction
   => (2nd semester)

2. Split KNU Sentiment Dictionary
   - Separate the emotional dictionary into four criteria -> Cancel

2. Split KNU Sentiment Dictionary

Table 4. Number of sentiment terms

food	price	service	atmosphere	total
438	142	682	581	1,843

Food(N=438)

Terms	Polarity
개운하다	2
감칠맛	2
깊은 맛이	2
맛없다	-1
...	

Price(N=142)

Terms	Polarity
비싸다	-2
저렴하다	1
최저가	1
비합리적	-2
...	

Service(N=682)

Terms	Polarity
친절하게	2
예의 바른	2
신경질적	-1
상냥하다	2
...	

Atmosphere(N=581)

Terms	Polarity
시끄럽다	-2
쾌적한	2
고급스러운	1
침침한	-1
...	

Initially, I tried to distinguish all 14000 words.

But the existing KNU method used at least three evaluators to determine the positive, neutral, and negative of each word and to reach an agreement through discussion (voting method).

Therefore, Data can have my subjectivity

So I made a model with the basic Sentiment dictionary.

2. Split KNU Sentiment Dictionary

The corresponding 'word' : converted the file into python dictionary data type with polarity and format. This made it easy to find the polarity of the word.

```
{
...
    "맛있는": 2,
    "맛있는 것을": 2,
    "맛있는 음식": 1,
...
}
```

Required Services

1. Crawling Data from Web
   => (Implemented)
1. Split KNU Sentiment Dictionary
   => (Canceled)
1. Sentiment analysis using KNU Sentiment Dictionary
   => (Implemented)
1. Data Preprocessing
   => (Implemented)
1. AI Modeling
   => (Implemented)
1. Comparison with paper
   => (Implemented)
1. Improvement Direction
   => (2nd semester)

3. Sentiment analysis using KNU Sentiment Dictionary
   - Analysis of reviews based on Sentiment Dictionary

3. Sentiment analysis using KNU Sentiment Dictionary

The following characteristics were extracted from the reviews.

review_score: the sum of the polarities of each review

sentimental_score: Total polarity sum of stores

pos_freq: morpheme frequency with good polarity

pos_val: Value of good polarity

neg_freq: Morphological frequency of bad polarity

neg_val: Value of bad polarity

```
# 모든 분석된 리뷰들에 대해
for review in analyzed_reviews:
    # 각 리뷰의 모든 형태소들에 대해
    for morph in review:
        review_score = 0
        # 만약 형태소가 사전에 있다면
        if morph in sentiment_dictionary:
            tmp_score = sentiment_dictionary[morph]
            # 나쁜 형태소가 나온 빈도, 그 극성의 합
            if tmp_score < 0:
                neg_freq += 1
                neg_val += tmp_score
            # 좋은 형태소가 나온 빈도, 그 극성의 합
            elif tmp_score > 0:
                pos_freq += 1
                pos_val += tmp_score
            review_score += tmp_score
            sentiment_score += tmp_score
```

3. Sentiment analysis using KNU Sentiment Dictionary

The data was saved as follows.
[Store review, good morpheme frequency, good morpheme value, bad morpheme frequency, bad morpheme value]

```
results.append((sentiment_score/review_Num, pos_freq/review_Num,
pos_val/review_Num, neg_freq/review_Num, neg_val/review_Num))
```

```
Output:
sentiment_score,pos_freq,pos_val,neg_freq,neg_val
1.6363636363636365,1.0454545454545454,1.6363636363636365,0.0,0.0
```

Required Services

1. Crawling Data from Web
   => (Implemented)
1. Split KNU Sentiment Dictionary
   => (Canceled)
1. Sentiment analysis using KNU Sentiment Dictionary
   => (Implemented)
1. Data Preprocessing
   => (Implemented)
1. AI Modeling
   => (Implemented)
1. Comparison with paper
   => (Implemented)
1. Improvement Direction
   => (2nd semester)

4. Data Preprocessing
   - Data Type, Scaling

4. Data Preprocessing - Data view

The data was saved as follows.

[Store review sentiment, good morpheme frequency, good morpheme value, bad morpheme value, rating]

pandas Dataframe info

```
sentiment_score,pos_freq,pos_val,neg_freq,neg_val,rating
1.6363636363636365,1.0454545454545454,1.6363636363636365,0.0,0.0,45
1.0,0.7170731707317073,1.1902439024390243,0.1317073170731707,-0.1902439024390244,45
0.5714285714285714,0.4778325123152709,0.6995073891625616,0.0689655172413793,-0.1280788177339901,40
```

```
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   sentiment_score  72 non-null     float64
 1   pos_freq         72 non-null     float64
 2   pos_val          72 non-null     float64
 3   neg_freq         72 non-null     float64
 4   neg_val          72 non-null     float64
 5   rating           72 non-null     int64
```

4. Data Preprocessing - Scaling

StandardScaler, not MinMaxScaler, for its data characteristics.

```
from sklearn.preprocessing import StandardScaler
# 스케일링
std = StandardScaler()
std.fit(df_x)
df_x_scaled = std.transform(df_x)
x_train, x_test, y_train, y_test = train_test_split(df_x_scaled, df_y,
test_size=0.2, random_state=37)
```

Required Services

1. Crawling Data from Web
   => (Implemented)
1. Split KNU Sentiment Dictionary
   => (Canceled)
1. Sentiment analysis using KNU Sentiment Dictionary
   => (Implemented)
1. Data Preprocessing
   => (Implemented)
1. AI Modeling
   => (Implemented)
1. Comparison with paper
   => (Implemented)
1. Improvement Direction
   => (2nd semester)

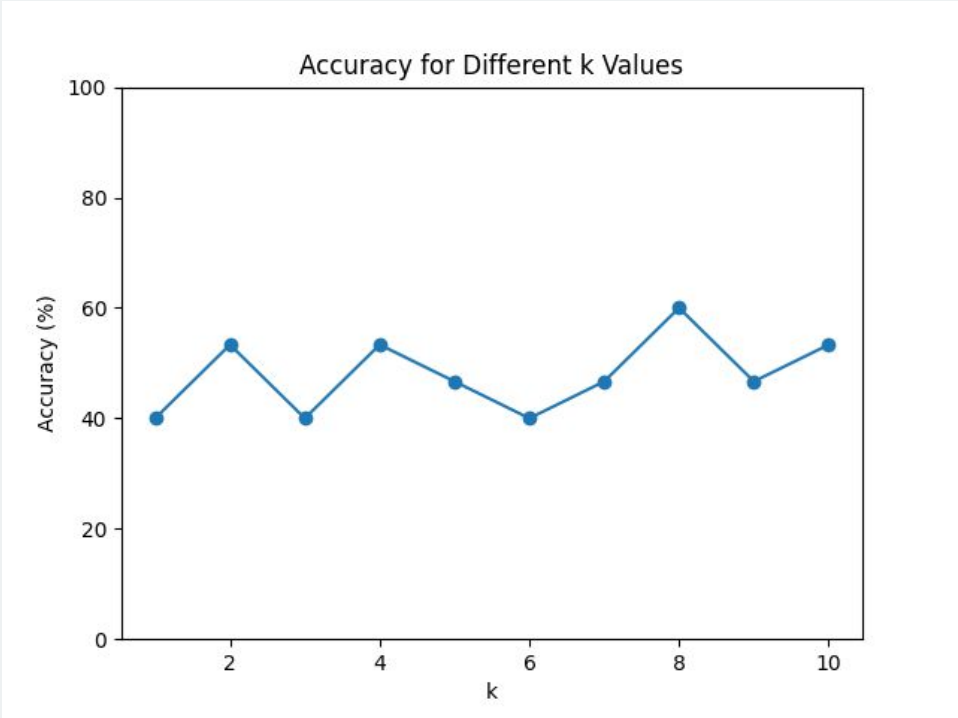
5. AI Modeling
   - KNN, Linear Regression, RandomForestRegressor, XGB Regressor

5. AI Modeling - KNN

accuracies while adjusting K to 1 to 10.

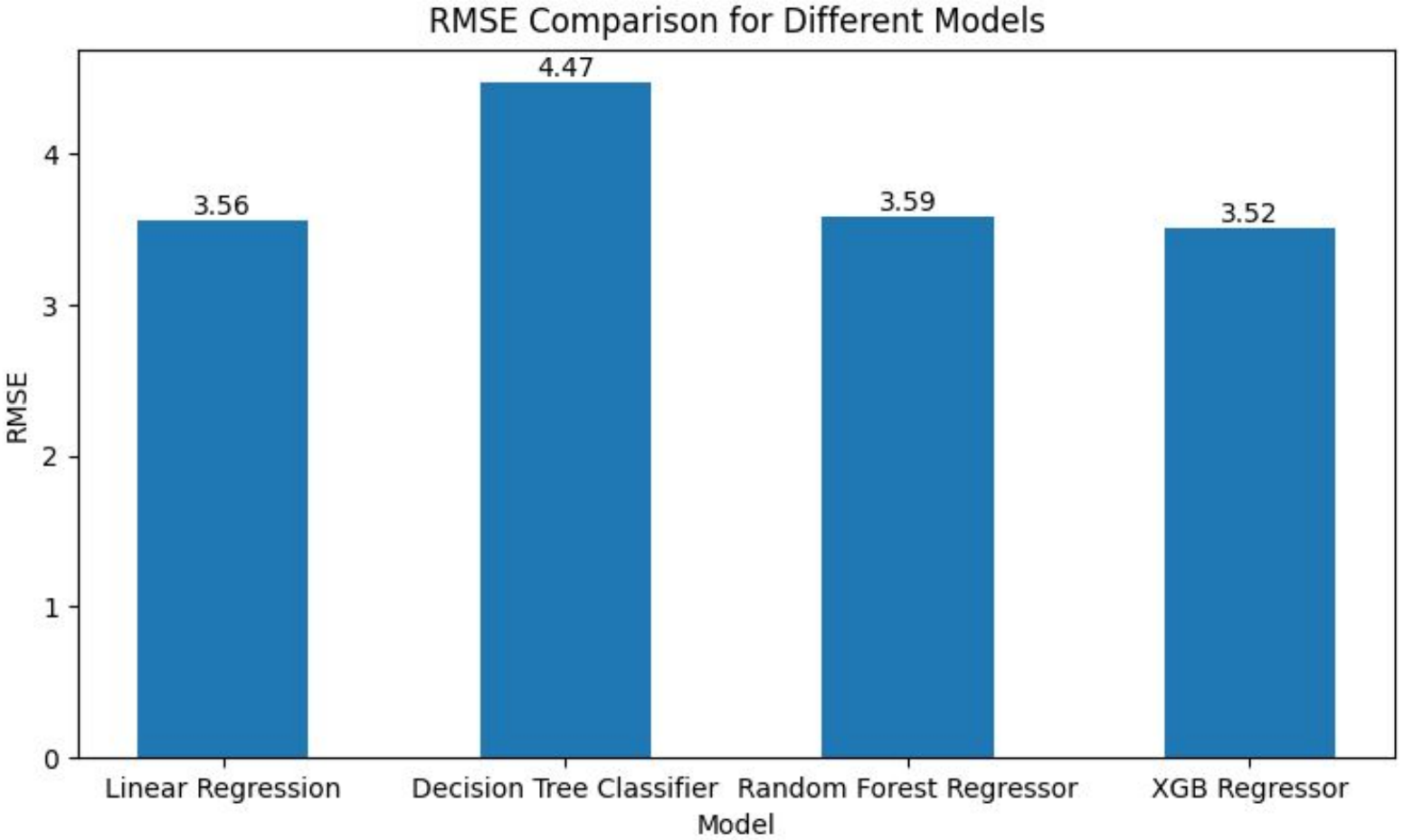
```
from sklearn.neighbors import KNeighborsClassifier
for k in range(1, 11):
    knn = KNeighborsClassifier(n_neighbors=k, n_jobs=-1)
    knn.fit(x_train, y_train)
    score = knn.score(x_test, y_test)
    print('k: %d, accuracy: %.2f' % (k, score*100))
```

5. AI Modeling - KNN Result




```
Output:
k: 1, accuracy: 40.00
k: 2, accuracy: 53.33
k: 3, accuracy: 40.00
k: 4, accuracy: 53.33
k: 5, accuracy: 46.67
k: 6, accuracy: 40.00
k: 7, accuracy: 46.67
k: 8, accuracy: 60.00
k: 9, accuracy: 46.67
k: 10, accuracy: 53.33
```

5. AI Modeling - RMSE Comparison for Different Models




```
LinearRegression
RMSE:  3.5616
--------------
DecisionTreeClas
sifier
Rmse:  4.4721
--------------
RandomForestRegr
essor
Rmse:  3.5915
--------------
XGBRegressor
RMSE :  3.5152
```

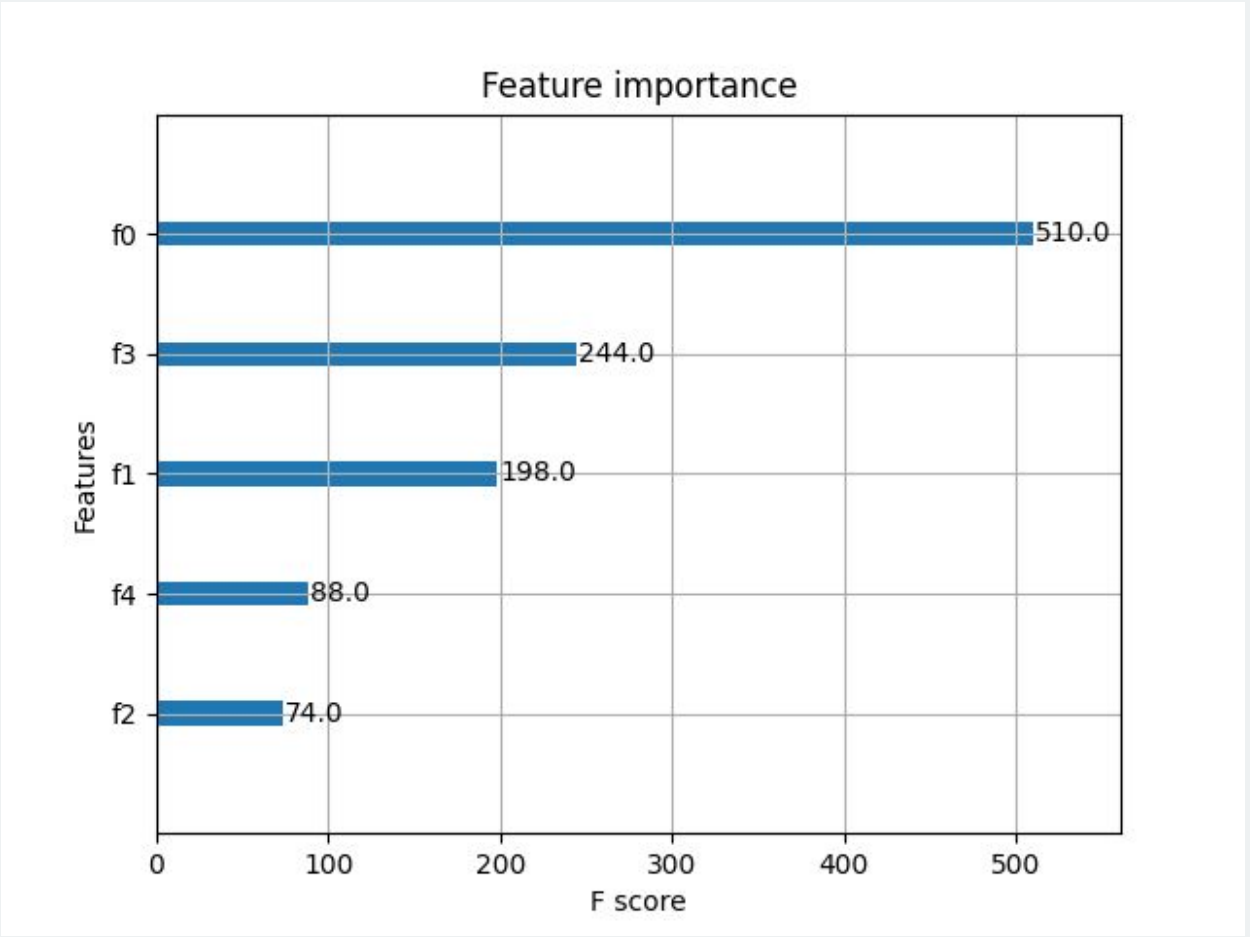
5. AI Modeling - XGBRegressor Predict Result

Examples predicted 5 with the lowest XGB Regressor.

```
Output:
XGBRegressor
Predict:  43.346458 Answer:  45
Predict:  43.153618 Answer:  45
Predict:  45.185047 Answer:  50
Predict:  42.336891 Answer:  40
Predict:  40.042076 Answer:  40
```

5. AI Modeling - XGB Regressor Feature Importance

The total Sentiment analysis ranks first,
neg_freq is second,
pos_freq is the third most influential.



Required Services

1. Crawling Data from Web
   => (Implemented)
1. Split KNU Sentiment Dictionary
   => (Canceled)
1. Sentiment analysis using KNU Sentiment Dictionary
   => (Implemented)
1. Data Preprocessing
   => (Implemented)
1. AI Modeling
   => (Implemented)
1. Comparison with paper
   => (Implemented)
1. Improvement Direction
   => (2nd semester)

6. Comparison with paper and
   - Compare with paper RMSE

6. Comparison with paper

It's very different from the paper

The difference from the paper was that the emotional dictionary was divided into four criteria.

Additionally, I used the y-value rating as int to use classifier. So, we had a value of almost 10 times 3.x, and we re-calculated the Regression RMSE by changing the rating to 0-5 instead of 0-50.

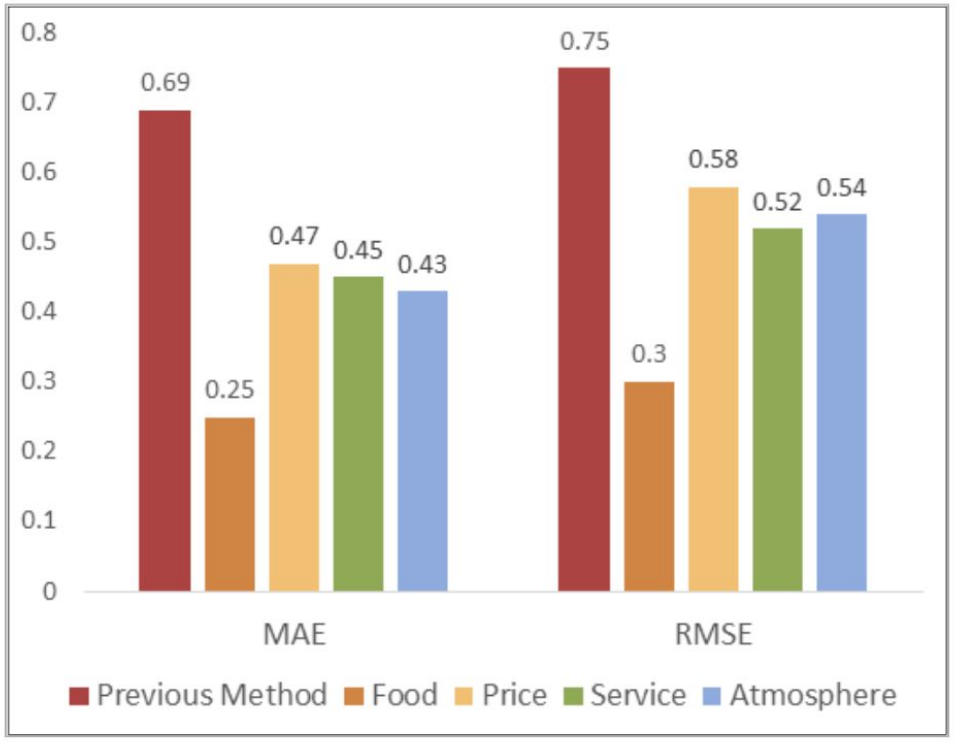
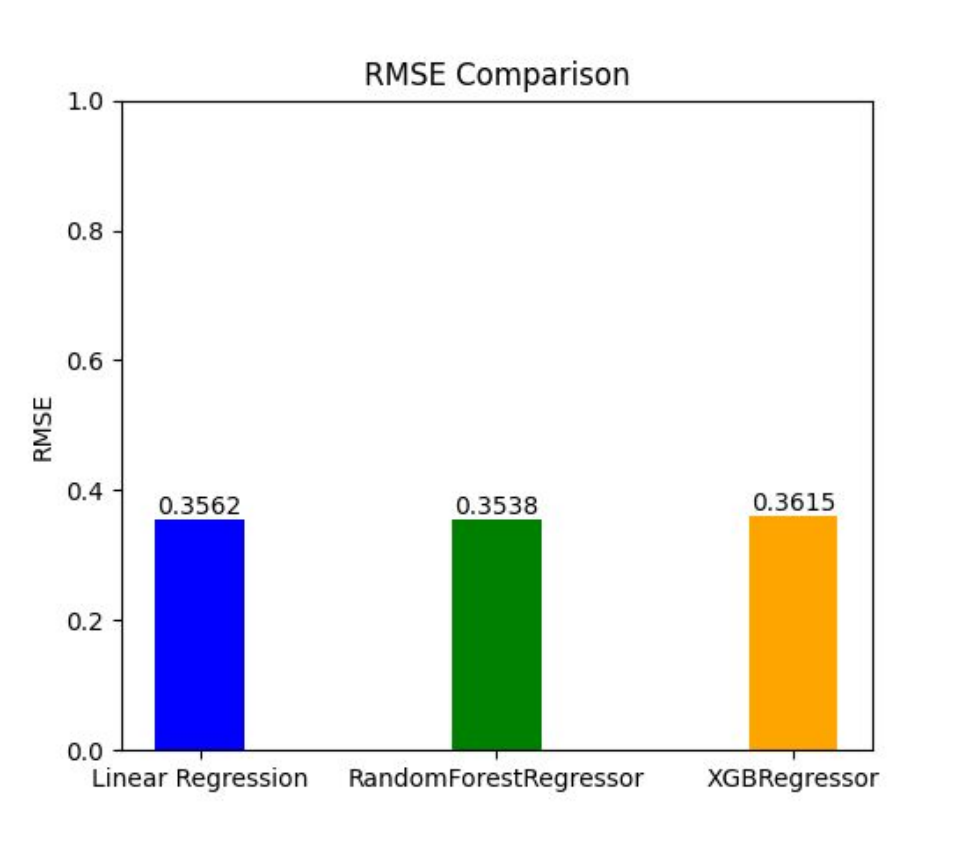


Fig. 5. Performance evaluation result (MAE, RMSE)

5. Comparison with paper - Float Rating(0 ~ 5)



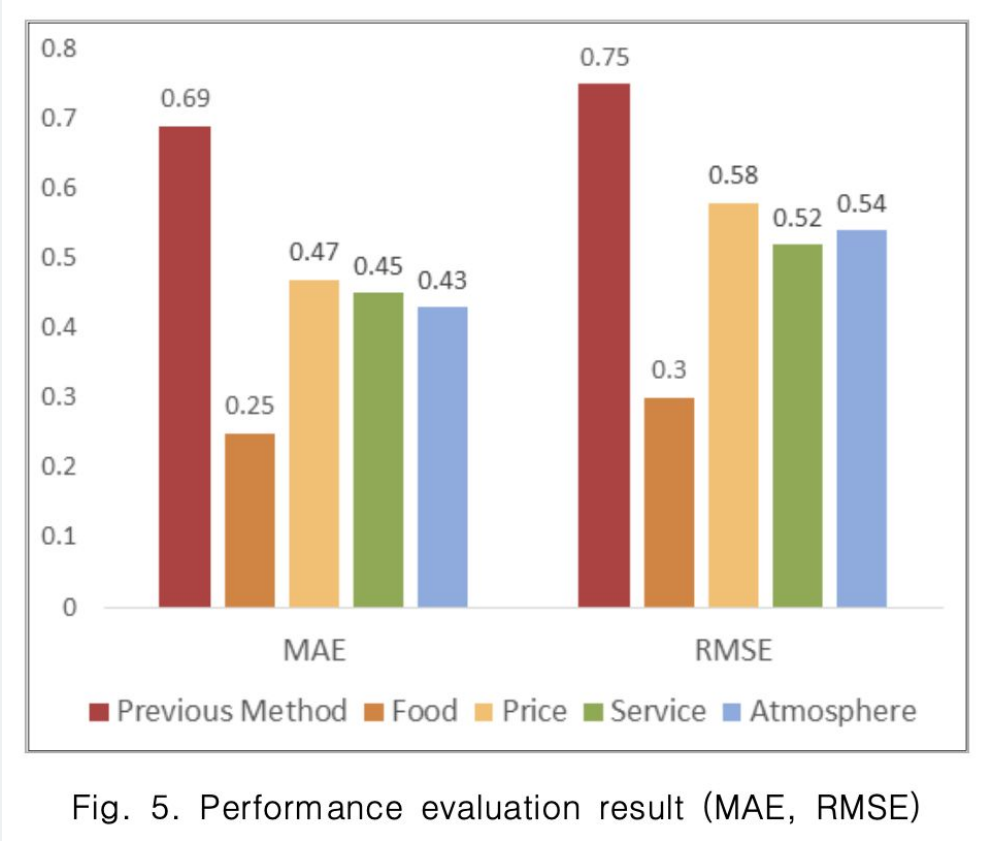


Fig. 5. Performance evaluation result (MAE, RMSE)

Required Services

1. Crawling Data from Web
   => (Implemented)
1. Split KNU Sentiment Dictionary
   => (Canceled)
1. Sentiment analysis using KNU Sentiment Dictionary
   => (Implemented)
1. Data Preprocessing
   => (Implemented)
1. AI Modeling
   => (Implemented)
1. Comparison with paper
   => (Implemented)
1. Improvement Direction
   => (2nd semester)

7. Improvement Direction
   - Tuning Hyper Parameter, New Model, More Data, Improve Model

7. Improvement Direction

- Get more data
- Try More Models
- Try NLP Model Not Sentiment analysis
- Try Various Model Hyperparameters
- Increase accuracy

Sources

- Rating Prediction by Evaluation Item through Sentiment Analysis of Restaurant Review
- 트립어드바이저
- GitHub - park1200656/KnuSentiLex: KNU(케이앤유) 한국어 감성사전