Review Predict with Text Analysis

고객관계관리

서예지 장예훈 조용걸

1. Introduction of Project

* Amazon의 Book Review Data 사용

- DATA

{ "reviewerID": "A2SUAM1J3GNN3B", "asin": "0000013714", "reviewerName": "J. McDonald", "helpful": [2, 3], **"reviewText"**: "I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think the book was published for singing from more than playing from. Great purchase though!", **"overall"**: 5.0, "summary": "Heavenly Highway Hymns", "unixReviewTime": 1252800000, "reviewTime": "09 13, 2009" }

Review Text를 입력하면 Overall을 예측

1. Introduction of Project

- DATA의 개수

Overall	TotalCount	Percentage of Data
1.0	323,833	3.6 %
2.0	415,110	4.7 %
3.0	955,189	10.7 %
4.0	2,223,094	25.0 %
5.0	4,980,915	56.0 %

Data set is Imbalanced

2. Preprocessing and Feature Engineering

<형태소 분석 후 TF-IDF>

① reviewText를 nltk로 품사 Tagging

② Tagging한 품사 중 Overall에 미치는 영향이 클 것이라 예상되는 품사 select

→ JJ(adv)/RB,VB(verb)

③ 특정 품사로만 이루어진 Data로 TD-IDF

④ Logistic Regression과 SVM모델을 사용하여 Training

⑤ Accuracy 측정

Accuracy Logistic Regression : 0.609 / SVM : 0.602

2. Preprocessing and Feature Engineering

<Data 전처리 후 TF-IDF>

① reviewText를 nltk와 BeautifulSoup으로 특수문자 제거, stop words 제거 후 Data Shuffle

② Data의 Imbalance를 맞추기 위해 제일 적은 overall(1.0) 개수에 맞춰 Data 자른 후 Data Shuffle

③ TF-IDF

④ Logistic Regression과 SVM모델을 사용하여 Training

⑤ Accuracy 측정

Accuracy Logistic Regression : 0.685 / SVM : 0.616

2. Preprocessing and Feature Engineering

<Data Oversampling&Undersampling>

① 1.0, 2.0, 3.0 Data를 Bootstrapping하여 개수를 2배 Oversampling

② 4.0, 5.0 Data를 Undersampling하기 위해 1.0 개수에 맞추기

③ nltk와 BeautifulSoup으로 특수문자 제거, stop words 제거 후 Data Shuffle

④ TD-IDF

⑤ Logistic Regression 모델을 사용하여 Training

⑥ Accuracy 측정

Accuracy Logistic Regression : 0.742

3. Hyperparameter

- Bootstrapping

```
from random import random, randrange
def subsample(dataset, ratio=2.5):
    sample = list()
    n_sample = round(len(dataset) * ratio)
    while len(sample) < n_sample:
        index = randrange(len(dataset))
        sample.append(dataset[index])
    return sample
```

Ratio : 2.5 로 조정

3. Hyperparameter

- TF-IDF Parameter

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(analyzer='word', sublinear_tf=True,lowercase=True,
                             stop_words='english', ngram_range=(1,2),
                             max_df = 0.2, min_df =2)

tfidf = vectorizer.fit_transform(text_data)
```

N-gram : (1,2), Max_df : 0.2, Min_df : 2 로 조정

3. Hyperparameter

- Logistic Regression

```
from sklearn import linear_model

logreg = linear_model.LogisticRegression(C=10.0, random_state=42,
                                          multi_class='multinomial’,
                                         warm_start=True, solver='sag',)


logreg.fit(x_train, y_train)
```

C : 10.0, multi_class : multinomial 로 조정

THANK

YOU

☺