Rating Prediction project

Submitted by:

Pushpender Kumar

ACKNOWLEDGMENT

Predicting rating by using review will be difficult these days in this project i have to work on text data most. And my last project helps me a lot for doing this same text type project but this project is different because i have to scrap data my self .

INTRODUCTION

- Business Problem Framing

  In real world with this prediction u can predict customers rating by using this review.

- Conceptual Background of the Domain Problem

  Concept behind this project is simple command input data and result will be customers ratings there are many site who wants to know customers rating by his reviews .

- Review of Literature

  In short in this text data I applied five model and random forest work good according to other model in fact this model aur data is not get high accuracy but random forest word well.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

This is classification types of problem so apply five model libraries and random forest work good than others

Decision tree:

accuracy: 68.02%

	precision	recall	f1-score	support
1	0.73	0.78	0.76	1364
2	0.39	0.31	0.34	301
3	0.47	0.42	0.44	440
4	0.54	0.47	0.50	904
5	0.75	0.80	0.77	2416
accuracy			0.68	5425
macro avg	0.58	0.55	0.56	5425
weighted avg	0.67	0.68	0.67	5425

Random Forest:

```
accuracy: 74.65%
              precision    recall  f1-score   support

           1       0.76      0.88      0.81      1364
           2       0.96      0.26      0.41       301
           3       0.96      0.33      0.49       440
           4       0.94      0.33      0.49       904
           5       0.71      0.97      0.82      2416

    accuracy                           0.75      5425
   macro avg       0.87      0.55      0.60      5425
weighted avg       0.79      0.75      0.71      5425
```

Adaboost classifier:

accuracy: 61.0%

	precision	recall	f1-score	support
1	0.67	0.71	0.69	1364
2	0.41	0.03	0.06	301
3	0.28	0.13	0.18	440
4	0.34	0.10	0.16	904
5	0.63	0.90	0.74	2416
accuracy			0.61	5425
macro avg	0.46	0.37	0.36	5425
weighted avg	0.55	0.61	0.55	5425

Gradient boosting classifier:

accuracy: 66.62%

	precision	recall	f1-score	support
1	0.71	0.77	0.74	1364
2	0.77	0.10	0.18	301
3	0.58	0.17	0.27	440
4	0.62	0.19	0.29	904
5	0.65	0.94	0.77	2416
accuracy			0.67	5425
macro avg	0.67	0.44	0.45	5425
weighted avg	0.66	0.67	0.61	5425

Bagging classifier:

```
accuracy: 71.85%
              precision    recall  f1-score   support

           1       0.70      0.86      0.77      1364
           2       0.58      0.27      0.36       301
           3       0.62      0.39      0.48       440
           4       0.64      0.42      0.51       904
           5       0.76      0.87      0.81      2416

    accuracy                           0.72      5425
   macro avg       0.66      0.56      0.59      5425
weighted avg       0.71      0.72      0.70      5425
```

- Data Sources and their formats

  This data i got from amazon site with defferent types of products like laptop,phone, camera and home treater etc. i scrap all data by using selenium and save in csv format.

- Data Preprocessing Done

  For preprocesing I have work on text data.preprocessing steps:

  1. I convert all text from review column into Lowercase by using simple code-

     ```
     df['Review']=df['Review'].apply(lambda x: x.lower())
     df['Review'].head()
     ```

  2. Removing all punctuation-

```
import string

def punctuation_removal(text):
    all_list = [char for char in text if char not in string.punctuation]
    clean_str = ''.join(all_list)
    return clean_str

df['Review'] = df['Review'].apply(punctuation_removal)
df['Review'].head()
```

3. Removing stopwords by NLTK library

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('english')

df['Review'] = df['Review'].apply(lambda x: ' '.join([word for word in x.split()
                                                      if word not in (stop)]))
df['Review'].head()
```

4. Removing commonly use words

```
freq = pd.Series(' '.join(df['Review']).split()).value_counts()[:10]
print(freq)

# common word removing

freq = list(freq.index)
df['Review']=df['Review'].apply(lambda x: " ".join(x for x in x.split()
                                                   if x not in freq))
```

CONCLUSION

- Key Findings and Conclusions of the Study

  Describe the key findings, inferences, observations from the whole problem.

- Learning Outcomes of the Study in respect of Data Science

  List down your learnings obtained about the power of visualization, data cleaning and various algorithms used. You can describe which algorithm works best in which situation and what challenges you

faced while working on this project and how did you overcome that.

- Limitations of this work and Scope for Future Work

  What are the limitations of this solution provided, the future scope? What all steps/techniques can be followed to further extend this study and improve the results.