FLIP ROBO

Reviews Rating Predictor

Submitted by:

Moncy Kurien

ACKNOWLEDGMENT

1. The data is scraped from Amazon.in, Flipkart.com and snapdeal.com.
2. Up-Sampling reference: https://github.com/kothiyayogesh/medium-article-code/blob/master/How%20I%20dealt%20with%20Imbalanced%20text%20dataset/data_augmentation_using_language_translation.ipynb

INTRODUCTION

- Business Problem Framing

  It is quite common for eCommerce companies to introduce new features into their existing e-commerce site. Our client has a website where people write different reviews for technical products. Now they are adding a new feature to their website. The reviewer will have to add stars(rating) as well with the review. The reviewers will be able to add star ratings (1 to 5) with their review. However, the reviews written in past will not have any stars. Hence, our client wants a solution that can predict the star rating by seeing the reviews written in the past.

- Conceptual Background of the Domain Problem

  The star rating will help new buyers to understand if a product will be useful for them or not. Hence, the star rating features is a good value addition. Being able to predict the star rating for all the past reviews add advantage to the customer in understanding the product.

- Motivation for the Problem Undertaken

  The star rating is a very useful feature for the customers to quickly understand about a product and decide if to buy it or not. By being able to predict the rating from past reviews, the client can benefit from implementing the 'add star' feature in their website for future reviews.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  The data gave a good result with RandomForestClassifier. RandomForest is built from an ensemble of Tree based estimators. This model uses Bagging technique.

- Data Sources and their formats

  The data required for the Rating predictor model is different reviews and ratings from different eCommerce websites. The data used in this project contains Reviews and the corresponding Ratings. The data is scraped from Amazon, Flipkart and Snapdeal sites. The dataset contains 2 columns:

  1. ‘Reviews’, an object data type – contains the reviews of different products.
  2. ‘Ratings’, numeric data type – Contains the ratings of each review. Ratings values are from 1 to 5.

  Screen-Shot:

	Reviews	Ratings
0	Super sounds.Super Bass quality and sound adhurs	5
1	Nice and good product.Very nice product	5
2	This headphone is awesome. Bass is very excell...	5
3	Service is awesome.Service is awesome	5
4	quality.superb sound...😍😍lovely product	5

- Data Preprocessing Done
  1. Converted all the characters in the ‘Reviews’ column to lowercase.
  2. Replaced all email addresses, website links, currencies, phone numbers and any numbers with constant texts link emailaddress, webaddress, currencyamount, phonenumber and numbr respectively.

3. Removed all non-alphabetic characters.
4. Replaced all multiple blank spaces with single blank space.
5. Changed the datatype of ‘Reviews’ as ‘str’ and stored as ‘reviews’.
6. Removed the stopwords from the ‘reviews’ column.
7. Used ‘reviews’ and ‘Ratings’ fields for further steps.
8. Used the cleaned ‘reviews’ field to create features using TFIDF with monograms, bigrams and trigrams. Max features restricted to 10,000 to compensate the computation power available.

- ## Data Inputs- Logic- Output Relationships

The input data consists of float values which are derived using TFIDF method from the ‘reviews’(cleaned ‘Reviews’) column in the dataset.

The TFIDF method uses monograms, bigrams and trigrams to create the independent features for the model and the output contains numerical classes.

The model approximates the function between the input and the output.

- ## State the set of assumptions (if any) related to the problem under consideration

No restrictions on where the data is scraped.

- ## Hardware and Software Requirements and Tools Used

1. Google Colab
2. SKLEARN
3. MATPLOTLIB
4. PANDAS
5. NUMPY
6. Google Translator

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
    1. Scrape the required dataset from different eCommerce websites.
    2. Clean the dataset using NLP approaches.
    3. Compare different models and identify the suitable model.
- Testing of Identified Approaches (Algorithms)
    1. SGDClassifier
    2. DecisionTreeClassifier
    3. MultinomialNB
    4. RandomForestClassifier
    5. AdaBoostClassifier
    6. GradientBoostingClassifier
    7. XGBClassifier

- Run and Evaluate selected models

<u>Comparing basic models using Accuracy metric:</u>

Code:

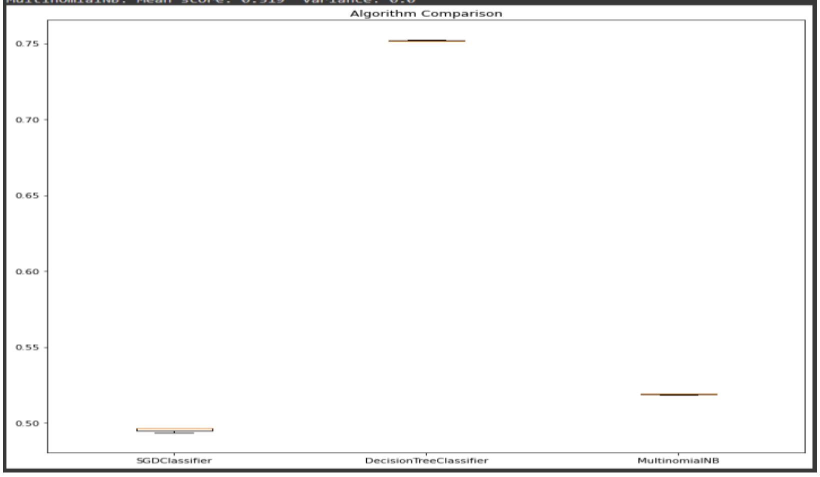
```
models = [SGDClassifier(),DecisionTreeClassifier(),MultinomialNB()]
results = []
m_names = []
for model in models:
    name = model.__class__.__name__
    kfold = KFold(n_splits=3, random_state=0, shuffle=True)
    cv_result = cross_val_score(model, x_train, y_train, cv= kfold, scoring = 'accuracy')
    results.append(cv_result)
    m_names.append(name)
    print(f"{name}: Mean score: {round(cv_result.mean(),3)}  Variance: {round(cv_result.var(),3)}")



# Compare Algorithms
plt.figure(figsize = (12,12))
plt.title('Algorithm Comparison')
plt.boxplot(results)
plt.xticks(np.arange(1,len(m_names)+1),labels=m_names)
plt.show()
```

Results:

```
SGDClassifier: Mean score: 0.495  Variance: 0.0
DecisionTreeClassifier: Mean score: 0.752  Variance: 0.0
MultinomialNB: Mean score: 0.519  Variance: 0.0
```



<u>Comparing Ensemble techniques using Accuracy metric:</u>

Code:

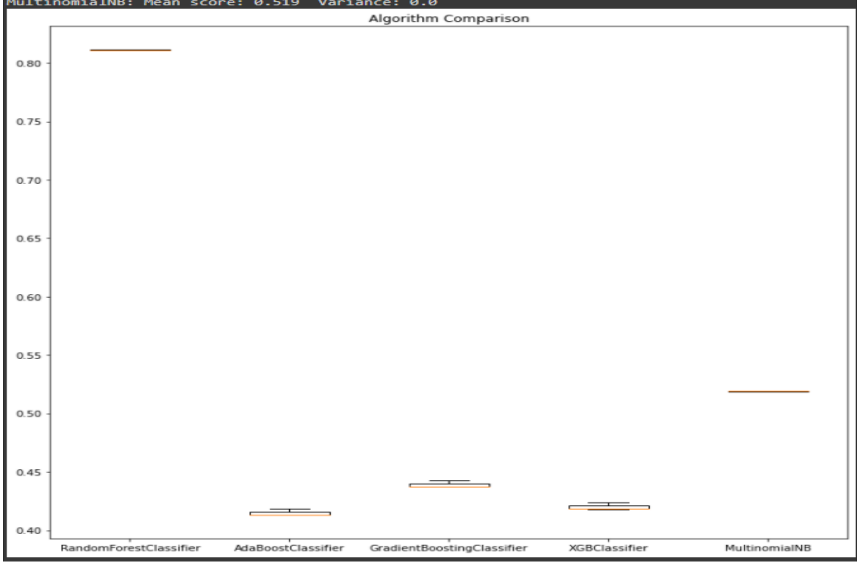
```
models = [RandomForestClassifier(),AdaBoostClassifier(),GradientBoostingClassifier(),XGBClassifier(),MultinomialNB()]
results = []
m_names = []


for model in models:
    name = model.__class__.__name__
    kfold = KFold(n_splits=3, random_state=0, shuffle=True)
    cv_result = cross_val_score(model, x_train, y_train, cv= kfold, scoring = 'accuracy')
    results.append(cv_result)
    m_names.append(name)
    print(f"{name}: Mean score: {round(cv_result.mean(),3)}  Variance: {round(cv_result.var(),3)}")


# Compare Algorithms
plt.figure(figsize = (12,12))
plt.title('Algorithm Comparison')
plt.boxplot(results)
plt.xticks(np.arange(1,len(m_names)+1),labels=m_names)
plt.show()
```

Results:

```
RandomForestClassifier: Mean score: 0.811  Variance: 0.0
AdaBoostClassifier: Mean score: 0.415  Variance: 0.0
GradientBoostingClassifier: Mean score: 0.439  Variance: 0.0
XGBClassifier: Mean score: 0.42  Variance: 0.0
MultinomialNB: Mean score: 0.519  Variance: 0.0
```

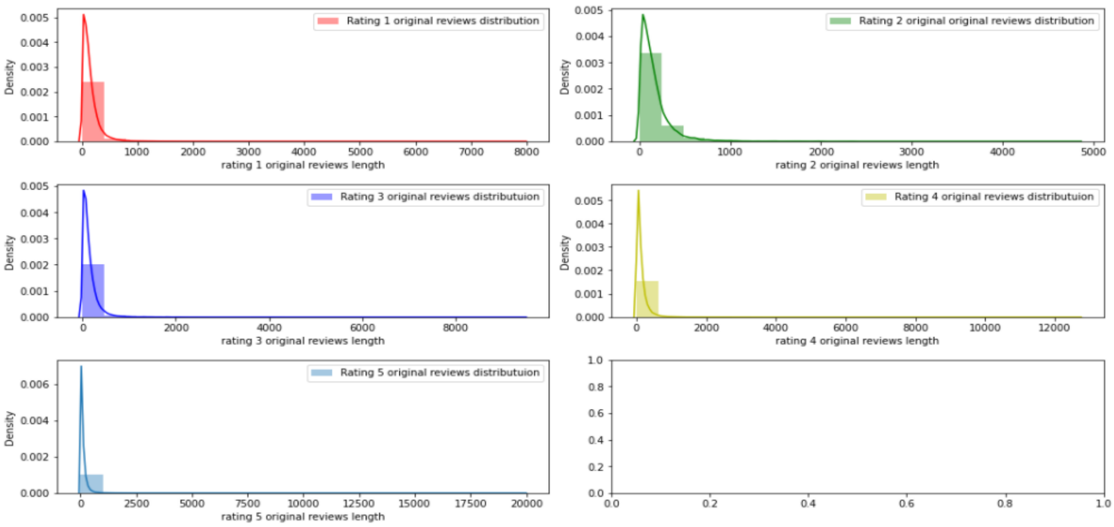


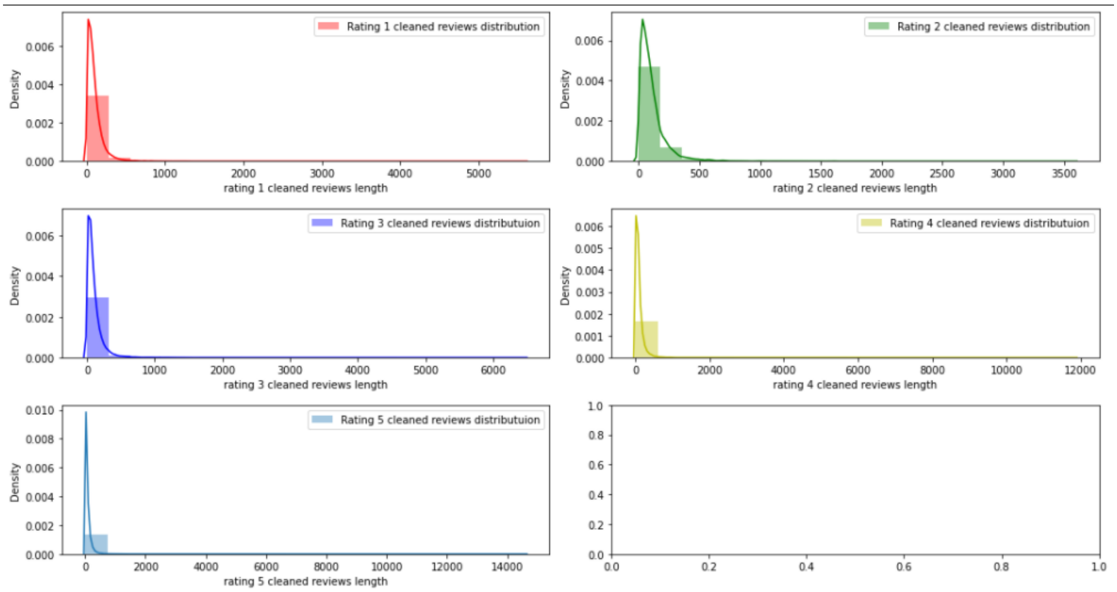
Observations:

1. The MultinomialNB, DecisionTreeClassifier and the RandomForestClassifier algorithms gave good results.
2. Among the above algorithms, the RandomForestClassifier is chosen for Hyperparameter tuning.

- Key Metrics for success in solving problem under consideration
    1. Accuracy is used as the key metric for evaluation. Because the dataset used is balanced, Accuracy is a good metric.
    2. Classification Report was used so that the Precision, Recall and F1 scores could also be used to evaluate the model.
- Visualizations

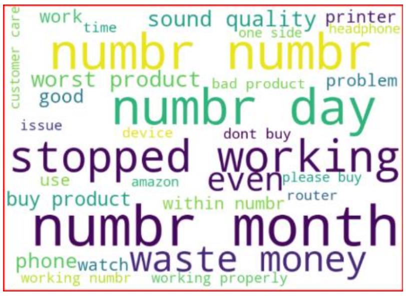
1. Original Reviews(top) vs cleaned reviews(bottom) lengths distributions:



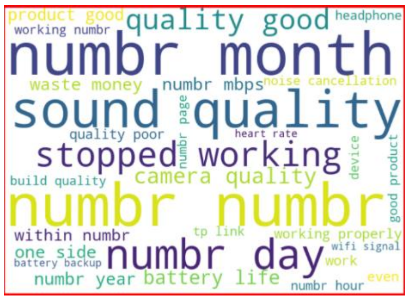


Observations:

1. The lengths of the cleaned reviews have decreased compared to the lengths of the original reviews.
2. Comparing Ratings 1, 2, 3 4 and 5 word-clouds:



Rating 1



Rating 2

Rating 3

Rating 4

Rating 5

Observations:

1. We can see that there is a clear distinction between the words used in the reviews of Rating 1 and Rating 5.
2. The Rating 1 reviews have many negative words for eg: 'Stopped working', 'waste money', 'bad product', etc,.
3. The Rating 5 reviews have many positive words for eg: 'value money', 'nice product', 'better', 'good quality'.
4. Some of these positive words in Rating 5 reviews are also available in Ratings 3 and 4. Also, there are some other words in the word cloud that are unique to Ratings 3, 4 and 5 respectively.
5. It would make sense to use monogram, bigram and trigram while using TFIDF.

- Interpretation of the Results
    1. Ratings 3,4 and 5 have many common words. The model could get confused between classifying the records as class 3, 4 and 5. Hence Ensemble techniques will be more effective than basic algorithms.
    2. Due to computational limits, only 10,000 features were used from TFIDF. Using more features may increase the model performances.

CONCLUSION

- Key Findings and Conclusions of the Study
    1. The reviews of the higher ratings contain more positive words than the least ratings.
    2. More the data used for training the better the model performed.
    3. RandomForestClassifier is able to perform well for the data used.
- Learning Outcomes of the Study in respect of Data Science

    Learned about the up-sampling techniques. Up-sampling is very useful in text/NLP based problems since in these problems, the more text combinations we are able to find the better the model performs.

- Limitations of this work and Scope for Future Work

    Computational power is the limitation faced in this project. The RAM memory in Google Colab was not enough for certain computations with the whole features.

    If there is enough computation power, using more data for training will give better results.