Project Report

Flipkart Customer Sentiment Detection

Submitted by :

Pranav Tiwari
Utkarsh Farkya
Anju Mankar

Mentors:

Poulami Bakshi
Subhajit Mondal

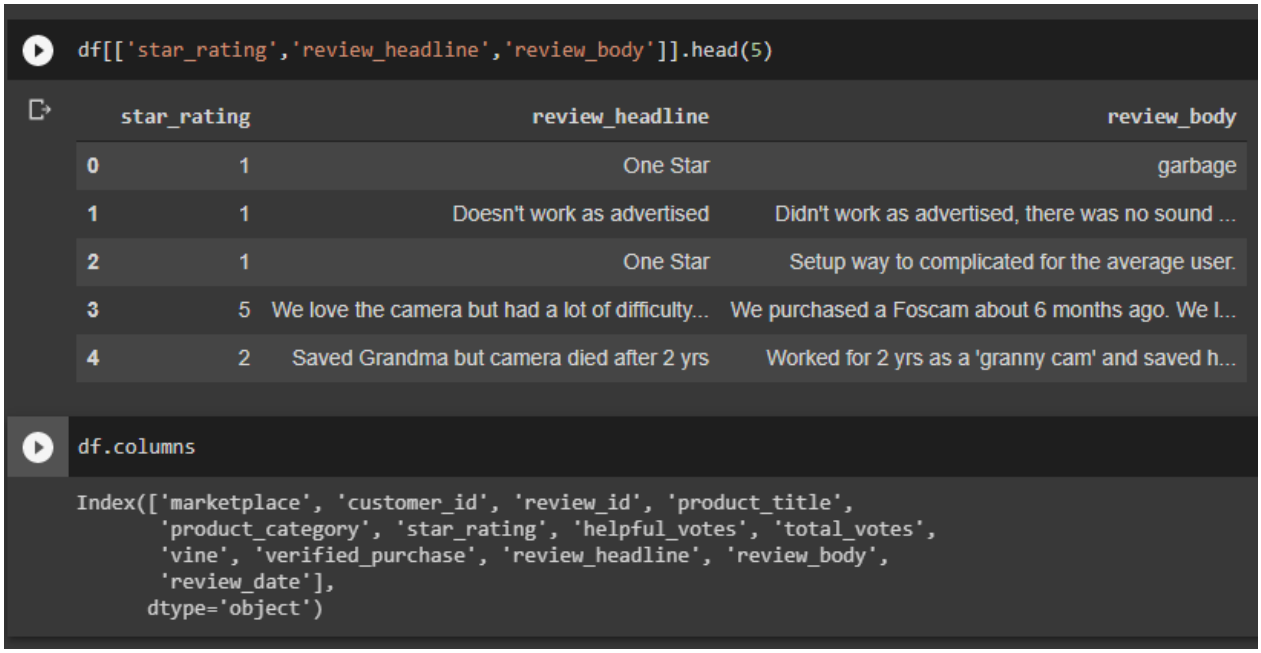
Sentiment analysis, also referred to as opinion mining, is a sub machine learning task where we want to determine what is the general sentiment of a given piece of text. Using machine learning techniques and natural language processing we can extract the subjective information of a given piece of text and try to classify it according to its polarity such as positive, neutral or negative. It is a really useful analysis since we could possibly determine the overall opinion about a selling object, or predict stock markets for a given company like, if most people think positive about it, possibly if its stock markets will increase, and so on. Sentiment analysis is actually far from being solved since the language is very complex (objectivity/subjectivity, negation, vocabulary, grammar,and so on) but it is also why it is very interesting to work on.

In this project we choose to try to classify reviews from Flipkart into “positive” or “negative” sentiment by building various models based on Machine Learning and Deep Learning. Flipkart is an online e-commerce website to buy various items ranging from groceries to electronics. People buy stuff and give their reviews based on how they feel about the product, whether they like it or dislike it, if setting up the product is easy or hard, if product is useful or not and much more.

Data

To gather data many options are available. We can collect dataset by web scraping or data mining from the website itself or can use the already available dataset. For this project we have used

Flipkart dataset i.e. reviews given on products bought from Flipkart.
Given dataset can be found here:
https://drive.google.com/drive/folders/1s4EQIUQxGYn4SMaXAvj2nmRPUrzrMJrO



```
df[['star_rating','review_headline','review_body']].head(5)
```

	star_rating	review_headline	review_body
0	1	One Star	garbage
1	1	Doesn't work as advertised	Didn't work as advertised, there was no sound ...
2	1	One Star	Setup way to complicated for the average user.
3	5	We love the camera but had a lot of difficulty...	We purchased a Foscam about 6 months ago. We l...
4	2	Saved Grandma but camera died after 2 yrs	Worked for 2 yrs as a 'granny cam' and saved h...

```
df.columns
```

```
Index(['marketplace', 'customer_id', 'review_id', 'product_title',
       'product_category', 'star_rating', 'helpful_votes', 'total_votes',
       'vine', 'verified_purchase', 'review_headline', 'review_body',
       'review_date'],
      dtype='object')
```

Preprocessing

The given dataset is already clean, not much effort is required to preprocess the data. No null values and duplicate values are present in the dataset that means we can directly proceed to prepare the dataset for modelling. The important features to us are reviews (review_headline + review_body) and the star_rating which gives us a sense of what emotion or sentiment a user is trying to convey through the review. star_rating of 1,2 is considered as negative and 4,5 is considered as positive and since we are not considering a neutral sentiment we will omit rows

with star_rating as 3. More importantly we are going to classify sentiments as positive and negative i.e. convert star_ratings of 1,2 to 0 (negative class or False) and 4,5 to 1 (Positive class or True).

```
[ ]  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4652 entries, 0 to 4651
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   marketplace        4652 non-null   object
 1   customer_id        4652 non-null   int64 
 2   review_id          4652 non-null   object
 3   product_title      4652 non-null   object
 4   product_category   4652 non-null   object
 5   star_rating        4652 non-null   int64 
 6   helpful_votes      4652 non-null   int64 
 7   total_votes        4652 non-null   int64 
 8   vine               4652 non-null   object
 9   verified_purchase  4652 non-null   object
 10  review_headline    4652 non-null   object
 11  review_body        4652 non-null   object
 12  review_date        4652 non-null   object
dtypes: int64(4), object(9)
memory usage: 472.6+ KB
```

```
[ ]  df.isnull().sum()
```

```
marketplace          0
customer_id          0
review_id            0
product_title        0
product_category     0
star_rating          0
helpful_votes        0
total_votes          0
vine                 0
verified_purchase    0
review_headline      0
review_body          0
review_date          0
dtype: int64
```

To prepare the dataset for modelling we extract new ratings and combined reviews column from dataframe to process it.

- Convert everything to lowercase.
- Remove HTML, XML tags from review data.
- Remove URLs from the data.
- Decontract words i.e. expand the short forms like won't, can't to will not and can not.
- Removing any kind of emoticons from the text. Emoticons can be a useful criteria to understand the sentiment of any text but in this model we are removing it keeping in mind the complexity of the model.
- Removing alphanumeric words, punctuations and stopwords from the text
- Finally we perform lemmatization on the text to chop down words to their root word, which initially might be in plural form or superlative degree.

Modelling

After preprocessing and preparing the data for modelling we start the modelling. Since we have to predict discrete values 1, 0 or true, false we will use Classification models in this case. This is a case of binary classification because we have only two classes to predict which are good and bad. There are a lot of classification models out there but we are going to use some specific models. The models used in this project is:

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Gradient Boosting Classifier

- Bagging Classifier
- Gaussian Naive Bayes
- K Neighbors Classifier
- Neural Network (CNN)

Accuracy and F1 score for these models are:

Logistic Regression

Accuracy score (training set): 0.99221902017291066
F1 score (training set): 0.9326874895607149
Accuracy score (test set): 0.8951612903225806
F1-score (test set): 0.9136786188579017

Decision Tree Classifier

Accuracy score (training set): 0.8838616714697406
F1-score (training set): 0.9326874895607149
Accuracy score (test set): 0.8502304147465438
F1-score (test set): 0.9136786188579017

Random Forest Classifier

Accuracy score (training set): 0.9982708933717579
F1-score (training set): 0.9989266547406083
Accuracy score (test set): 0.8709677419354839
F1-score (test set): 0.9224376731301939

Gradient Boosting Classifier

Accuracy score (training set): 0.8769452449567723
F1-score (training set): 0.928845192467922
Accuracy score (test set): 0.8502304147465438
F1-score (test set): 0.9153645833333333

Bagging Classifier

Accuracy score (training set): 0.9982708933717579
F1-score (training set): 0.9989266547406083
Accuracy score (test set): 0.8421658986175116
F1-score (test set): 0.9029057406094969

Gaussian Naive Bayes

Accuracy score (training set): 0.9662824207492795
F1-score (training set): 0.9785988659228095
Accuracy score (test set): 0.8271889400921659
F1-score (test set): 0.8995983935742972

K Neighbors Classifier

Accuracy score (training set): 0.8919308357348703
F1-score (training set): 0.9364083432253688
Accuracy score (test set): 0.8122119815668203
F1-score (test set): 0.8908238446081714

Neural Network (CNN)

Accuracy score (training set): 0.985878962536023
F1-score (training set): 0.9912327786723921
Accuracy score (test set): 0.9297235023041475
F1-score (test set): 0.957491289198606 3

After training the dataset we can see that the Deep Learning based neural network model has the best performance over all other models that we have used in the whole project. All other models other than neural networks have an accuracy score over

the test set not more than 89% while the neural network gives an accuracy score of over 92% on the test set which of course is better than all others. Our major focus was to increase the accuracy score of our test set which was achieved using neural networks. Therefore we can conclude that neural network is the best model out of all in this case.