Ratings Predictions Project

Submitted by:

K. Rahul Ramanujam

ACKNOWLEDEGMENT

1) The very basic idea behind the gathering the data (Product reviews from e-commerce sites) has been covered to us, we've been made to work on the projects where we had to scrape the data from websites and then work on it. Thus, experience of earlier web-scraping projects has come in handy in gathering the data for this project.

2) The professional help that I've received was from our reporting manager Mr. Shubham Yadav, he has helped me understand on how important the data scraping is. I'd like to credit him for helping me on data scraping of nearly the length of around 34K units.

3) The rest of the project needed text pre-processing and model building; those concepts have already been covered in my academy (DataTrained). I have used NLP methods to filter off the odd/extra data from the review texts and then the ML modelling is done.

INTRODUCTION

- The business problem involved in this project is that we have a client who runs a website where people write different technical reviews. Now they want to add a new feature to rate the products. The new feature is for the reviews from now, but the already written ones will be left out. Thus, we'll have to come up with a model which will predict the ratings based on the review.

- The appropriate domain problems like customer retention projects, sentiment analysis projects are some of the best ones to refer and work. Those will give the basic idea of how the companies go through the concepts of retaining their position in the market by keeping the constant surveillance on their products and its reception from people. Customer sentiments are something that the firms value the most, they do whatever they can to please them. Thus, the current project will give a gist of how the sentiment analysis works and how the companies go through all of them.

- The only objective was to find the right model in the end which will predict the ratings for the reviews.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  There aren't much mathematical calculations involved as the data is of review texts. The analysis is done during the text conversion to numerical with the help of Tfidf and CountVector.

- Data Sources and their formats

  The data is basically the reviews from e-retailors websites. The customers express their opinions on their site, we as a data scientists scrape such data to do the sentiment analysis on it and model it eventually. The format of the data is raw and object datatype in nature.

	reviews	ratings
0	Camera is not that good but the performance wi...	4
1	Best for high end graphic games. PUBG is avail...	4
2	Best quality products	5
3	Value for money..... but heavy	5
4	Camera quality & performance is good.. feature...	4
...	...	...
35709	Nice phone	3
35710	Superb quality ❤️👍	5
35711	Value for money 💰	5
35712	Worst battery pickup	1
35713	Very nice product	5

[35714 rows x 2 columns]

- Data Preprocessing Done

  Data preprocessing is an integral part of the projects like these. The basic idea behind this is to filter the raw data first while applying the NLP methods to it.

Text Preprocessing

The preprocessing involves following changes:

1) URLs, as the people tend to spam the section with irrelevant urls, we need to get rid of those for better modelling.

2) Numbers and Punctuation: We don't need those as they hardly add up any info in the end.

3) Converting the text into lowercase as it will bring the uniform nature to it.

4) Tokenize the data to:

```
i) Remove stopwords (The words which don't add up anymore than the rest.)

ii) Stemming and Lammatization (Methods to reduce the noise and extra info from the data)

iii) Remove the words having length <= 2
```

5) Convert the list of tokens into back to the string

The above screenshot is from the notebook where, I've followed the steps to pre-process the data before sending it the modelling stage.

- Data Inputs- Logic- Output Relationships

  The input was nothing but a raw data scraped from the e-commerce site which has reviews (texts) and ratings count (1 star, 2star etc.). Initially the reviews column was of object datatype, later it was converted to string so that the changes can be made accordingly. The relationship between the input and output has been made to be more clear post text-processing as it had many outcasts which the model couldn't read, thus, getting rid of those made the job easy in modelling stage. The final model is now able to predict the ratings if the input are given as reviews.

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

  The assumptions had to be taken while the text pre-processing stage since I wasn't sure how well the scraped data is from the outcast's presence. Thus, I had to take an assumption that there might be presence of outcasts like Unnecessary single letter texts, numbers, punctuations etc. I can't go through each and every piece of the data since it was around 35K in length.

- Hardware and Software Requirements and Tools Used

  I used Google Colab instead of jupyter notebook for this particular project. There is a data of 35K in length, it is tough for a local machine to do the modelling with such huge data thus, I had to use colab since it has GPU support.

  Since NLP and neural network concepts were used in the project, there are some packages and libraries called:

  From NLTK we've imported libraries for stemming and lamenting. Also to get rid of Stopwords, NLTK package us used.

  To convert the text data into numerical, we used *sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer*

  For data modelling we used Random Forest, XGBoost, NaiveBayes and LSTM of ANN.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  For the problem-solving, I had to first look at the data properly in order to see what are the tools or libraries required for it. Then, I had to used NLP methods like getting the data off the stopwords, punctuations, single letter words, URL links and so on. Text pre-processing method has done a half job, the other half of the job is to find the right models for it predict the data correctly.

- Testing of Identified Approaches (Algorithms)

i) First, the ratings column of the data is Imbalanced thus, we've reduced to sentiments.
ii) Second, the reviews column of the data is converted to string datatype and then X and Y data split is done using traintestsplit library.
iii) Further training and testing is done in the model selection phase. Every model is made to train the data and test it simultaneously.
iv) The tools like: Accuracy score, f1 score, Classification report, Confusion Matrix are used to find the score of the test done.

- Run and Evaluate selected models

XGBoost ensemble model is used to find the right prediction score while it undergoes Instantiation, fitting and predictions.

XGBoost

```
# Instantiation, fitting and predictions

xgb_ = xgb.XGBClassifier(
 learning_rate =0.1,
 n_estimators=1000,
 max_depth=5,
 min_child_weight=1,
 gamma=0,
 subsample=0.8,
 colsample_bytree=0.8,
 objective= 'multi:softmax',
 nthread=4,
 scale_pos_weight=1,
 seed= seed)

xgb_.fit(X_train, y_train)
predictions = xgb_.predict(X_test)
```

The parameters are assigned according to the data that is given to it.

a) n_estimators = no.of trees. (1000 is chosen as the data is huge)
b) max_depth = Depth of each tree. (5 is given to increase each tree's grasping capacity)
c) objective = the aim of the model. (softmax is used since it's a multi-classification problem)

d) nthread = Number of cores being utilized. (4 is given, 6 or 8 can also be given according to the machine capacity)

- Interpretation of the Results

Post text pre-processing, the data looked much cleaner and on point. It had no extra baggage of info except the ones that mattered. NLP methods have come in handy in dealing with the texts.

Post data vectorization with the help of countvector and TfIdf, the data split was one. Y data had the data of ratings which were then reduced to sentiments as it was imbalanced and there was a risk of model's prediction error. Thus, reducing it to mere sentiments made things easy for the model processing.

CONCLUSION

- Key Findings and Conclusions of the Study

1) I've learned that the type of data can play a major part in the final prediction. More refined the data is, better will be the prediction by the best-chosen model.
2) The role of ensemble models such as XGBoost plays a crucial part in the projects like these, the hyperparameters that were used in gradient boosting have played a major part in better prediction.

- Learning Outcomes of the Study in respect of Data Science

According to what I've gathered from doing this project is that, the role NLP – Natural Language Processing is superior than the rest. It not only helps in data cleaning but also helps in finding the sentiments out of it.

Neural Networks methods can come in handy here, the ability to work on the large datasets is something that almost no other models can do it in minimal time as ANN models do.

I've learned that the projects which has huge data like this one, must be dealt with utmost planning, since, the training time can take away all the time we have, thus, we need to chose those parameters which will work efficiently while taking less time to finish it.

- Limitations of this work and Scope for Future Work

  I wish there was more time given as the data is huge, the processing time will take longer time to process the data.

  For future work, I expect the data requirement for the work should be accounted for the time being given to finish the project.