Ratings Prediction Project

Submitted by:

Akriti Kakkar

ACKNOWLEDGMENT

I have referred to data trained study material and python documentation for this project.

INTRODUCTION

- Business Problem Framing

  We have a client who has a website where people write different reviews for technical products. Now they are adding a new feature to their website i.e. The reviewer will have to add stars(rating) as well with the review. The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars. Now they want to predict ratings for the reviews which were written in the past and they don't have a rating. So, we have to build an application which can predict the rating by seeing the review.

- Conceptual Background of the Domain Problem

  Data Description:

  1. Comment: Review of the product.
  2. Rate: Star Rating by reviewer. The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars.

- Review of Literature

Follow the links to the dashboards:

https://docs.google.com/presentation/d/1sQ3S8YE8Y8oSzELe0gSy5jjXoH5IOLQQwU8Su1kub0M/edit?usp=sharing

https://docs.google.com/presentation/d/1wmvgQVB50HUQeZqG-Kqb0bPwPzX5dLt_wshQCDbW5WM/edit?usp=sharing

https://docs.google.com/presentation/d/1b0TEUZI2cHsC4spBMJUfnnVaYzZY4AhbdlFDQPU0ojc/edit?usp=sharing

https://docs.google.com/presentation/d/1dJccwP6hPnyJyUTV7SpbVyJz9vAgKxCrACm3tIiZicg/edit?usp=sharing

- Motivation for the Problem Undertaken

Client wants to predict ratings for the reviews which were written in the past and they don’t have a rating. So, we have to build an application which can predict the rating of electronic products by seeing the review.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem
- Read Dataset And Make It In Proper Format
- Encode Labels
- Convert All Cases To Lower
- Remove Punctuations
- Remove Stopwords
- Check stats of comments
- Convert all texts into vectors
- Import classifier
- Train And Test
- Check the accuracy/confusion matrix

- Data Sources and their formats

The data is scrapped from e commerce websites (unstructured data) and has been converted into structured format (csv data).

- Data Preprocessing Done
- Read Dataset And Make It In Proper Format
- Encode Labels
- Convert All Cases To Lower
- Remove Punctuations
- Remove Stopwords
- Check stats of comments
- Convert all texts into vectors

- Data Inputs- Logic- Output Relationships

  Data input is comment and output is stars rating. Good review leads to higher rating and vice versa. Data inputs are comments and are converted into tfid vectors.

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

  No assumptions are made in this project.

- Hardware and Software Requirements and Tools Used

Installation of Anaconda Community version (Open Source).

Required Installations:

Pandas (Within Environment)

Numpy (Within Environment)

Seaborn (Within Environment)

Matplotlib (Within Environment)

TFID Vector

Word Cloud

Sklearn

Pickle

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

2. Read Dataset And Make It In Proper Format

3. Encode Labels
4. Convert All Cases To Lower
5. Remove Punctuations
6. Remove Stopwords
7. Check stats of comments
8. Convert all texts into vectors
9. Import classifier
10. Train And Test
11. Check the accuracy/confusion matrix

- Testing of Identified Approaches (Algorithms)

Model: Multinomial NB

- Run and Evaluate selected models https://docs.google.com/presentation/d/1dJccwP6hPnyJyUTV7SpbVyJz9vAgKxCrACm3tIiZicg/edit?usp=sharing`
- Key Metrics for success in solving problem under consideration

  TFID Vectorizer played a crucial role in success of the model. Scikit-learn is a free software machine learning library for the Python programming language. It supports Python numerical and scientific libraries, in which TfidfVectorizer is one of them. It converts a collection of raw documents to a matrix of TF-IDF features. As tf–idf is very often used for text features, the class TfidfVectorizer combines all the options of CountVectorizer and TfidfTransformer into a single model. The TfidfVectorizer uses an in-memory vocabulary (a python dict) to map the most frequent words to feature indices and hence compute a word occurrence frequency (sparse) matrix.

- Visualizations

Follow the links to the dashboards:

https://docs.google.com/presentation/d/1sQ3S8YE8Y8oSzELe0gSy5jjXoH5IOLQQwU8Su1kub0M/edit?usp=sharing

https://docs.google.com/presentation/d/1wmvgQVB50HUQeZqG-Kqb0bPwPzX5dLt_wshQCDbW5WM/edit?usp=sharing

https://docs.google.com/presentation/d/1b0TEUZI2cHsC4spBMJUfnnVaYzZY4AhbdlFDQPU0ojc/edit?usp=sharing

- Interpretation of the Results

  Follow the link to the dashboard:

https://docs.google.com/presentation/d/1dJccwP6hPnyJyUTV7SpbVyJz9vAgKxCrACm3tIiZicg/edit?usp=sharing

CONCLUSION

- Key Findings and Conclusions of the Study

Follow the links to the dashboards:

https://docs.google.com/presentation/d/1sQ3S8YE8Y8oSzELe0gSy5jjXoH5IOLQQwU8Su1kub0M/edit?usp=sharing

https://docs.google.com/presentation/d/1wmvgQVB50HUQeZqG-Kqb0bPwPzX5dLt_wshQCDbW5WM/edit?usp=sharing

https://docs.google.com/presentation/d/1b0TEUZI2cHsC4spBMJUfnnVaYzZY4AhbdlFDQPU0ojc/edit?usp=sharing

https://docs.google.com/presentation/d/1dJccwP6hPnyJyUTV7SpbVyJz9vAgKxCrACm3tIiZicg/edit?usp=sharing

- Learning Outcomes of the Study in respect of Data Science

Visualizations and data cleaning convert a whole complex and messy dataset into insightful and interesting representation, which make it easier to reach the core of the problem and solve it.

Multinomial NB has provided good accuracy (above hit ratio) with right fit.

In training mode, it has provided accuracy of 68% and in testing mode it has provided accuracy of 66%, hence, I have saved this model for production.

TFID Vectorizer solved dimensionality and dealing with text data issues.

- Limitations of this work and Scope for Future Work

  Further optimization can be obtained by applying deep learning solutions. Since, it requires very high RAM capacity, it could not be displayed in jupyter notebook... I would like to update Google Colab Notebook for future projects, if acceptable... That can help me to submit a completely optimized model