**Ratings Prediction**

Submitted by:

Amit Kulkarni

**ACKNOWLEDGMENT**

# Yelp Review Rating Prediction: Machine Learning and Deep Learning Models

* [Zefang Liu]

## Blog by Yereya Berdugo on towards data science named **Review Rating Prediction: A Combined Approach**

# Automatic movie ratings prediction usingmachine learning [Mladen Marovi ́c, Marko Mihokovi ́c, Mladen Mikˇsa, Siniˇsa Pribil, and Alan TusUniversity of Zagreb, Faculty of Electrical Engineering and ComputingUnska 3, Zagreb, Croatia]

### **Multilingual Twitter Sentiment Classification: The Role of Human Annotators** [Igor Mozetič, Miha Grčar, and Jasmina Smailovič, from the Department of Knowledge Technologies at the Jožef Stefan Institute]

**INTRODUCTION**

* Business Problem Framing

Business requirement in this problem was to classify the given reviews into 5 classes based on the emotion/ sentiment in the text.

The implementation will help to predict the ratings for the test data set and further can be implemented in web app to automatically predict ratings for reviews.

* Conceptual Background of the Domain Problem

Classification and sentiment analysis will play a key role in obtaining a solution for the problem.

A study of frequently occurring words in the corpus will help to get an idea of the different rating classes.

* Motivation for the Problem Undertaken

The implementation of the project will help to successfully predict ratings for reviews without any human involvement and scrutiny

**Analytical Problem Framing**

* Data Sources and their formats

Data for the project is scraped majorly from two e-commerce sites viz. Amazon and Flipkart.

Scrapers were designed in selenium

* Data Preprocessing Done

Symbol cleaning was done.

The review text and star ratings were sliced in order to include only relevant data

Symbols cleaning was also done.

Irrelevant columns were dropped.

* Data Inputs- Logic- Output Relationships

Emotions from text can be extracted based on frequently occurring words.

Sentiment analysis is done in order to classify review text broadly in 3 classes (positive, negative and neutral)

The newly created sentiments column is highly correlated with the ratings thus giving a firm relationship idea about the data.

* Hardware and Software Requirements and Tools Used

The software development was done on jupyter notebook

Hardware specs used were:

1. 8 GB memory
2. I7 processor
3. Os: windows 10 home 2019
4. Selenium for automating data scraping.

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

Two approaches that were followed are as follows:

1. Sentiment Analysis

Frequently occurring words from the classes were extracted and test data was searched for those words to classify/ attach that particular emotion to text

1. Classification

The data is classified into 5 classes according to feature in the text.

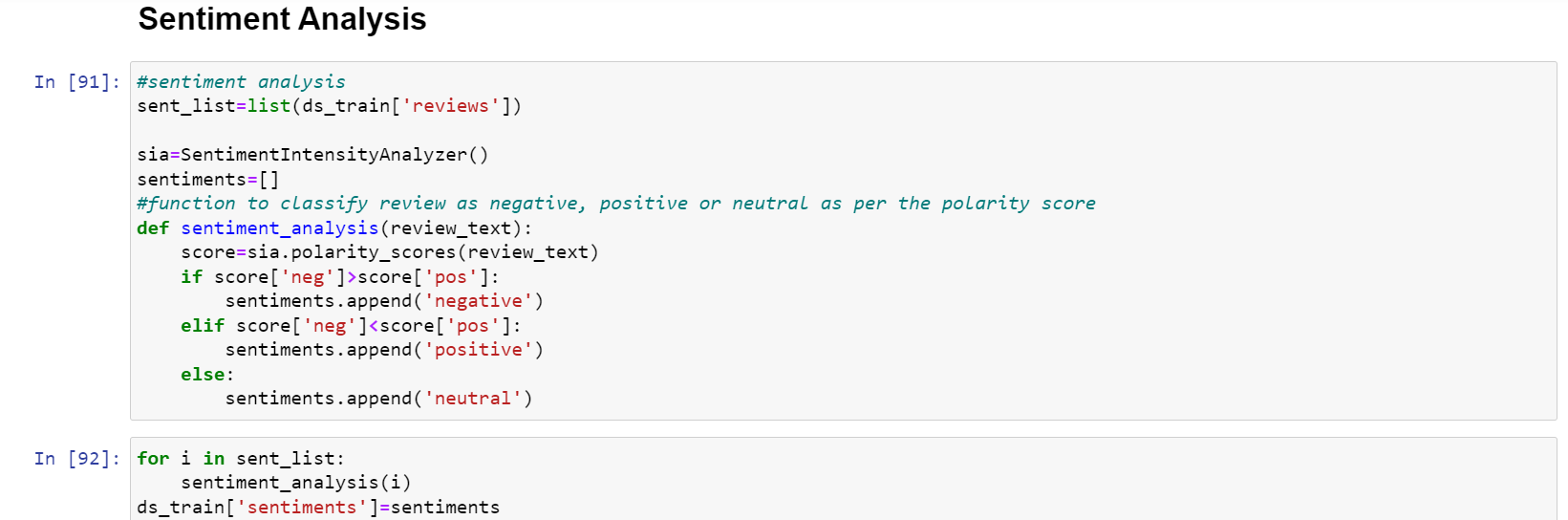
* Testing of Identified Approaches (Algorithms)

1. Decision Tree Classifier
2. Random Forest Classifier
3. Switched Vector Classifier

* Run and evaluate selected models

**Approach 1:**

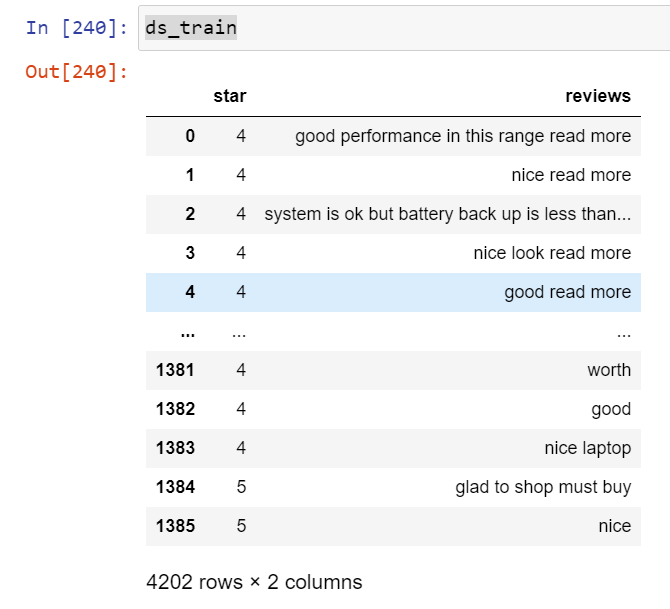
1. Frequently occurring words from the classes were extracted
2. Sentiment intensity analyser was used to classify the review as positive negative or neutral



**Approach 2:**

1. The reviews in train classified on the basis of star ratings and a model was trained on these features

Final dataframe:



Further classification algorithms were trained according to the train data set and predictions on test data set were done

* Key Metrics for success in solving problem under consideration

Metrics that were used were as follows:

1. accuracy\_score
2. classification\_report
3. confusion\_matrix

* **Visualizations**

First visualization that was used here was a word cloud



To get the word clouds list of reviews with different classes were created.

So the created lists were as follows:

1. List\_1:1 star
2. List\_2:2 star
3. List\_3:3 star
4. List\_4:4 star
5. List\_5:5 star

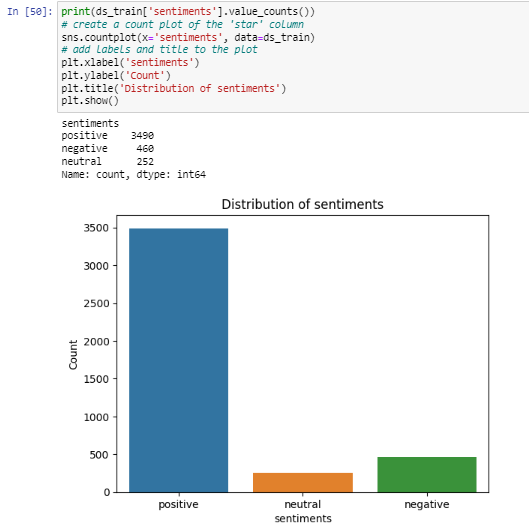
the word clouds for these lists were then plotted accordingly.

Visualization of counts of all the classes



A column of the sentiments was added to the dataframe and the count was plotted.

Observing the plots a relation between the reviews and ratings can be established.

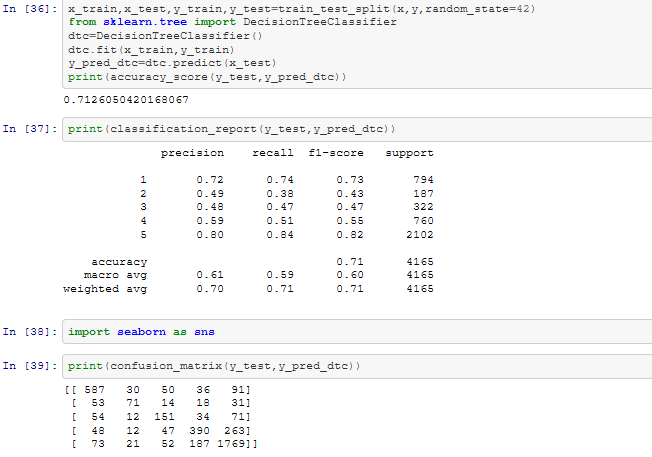


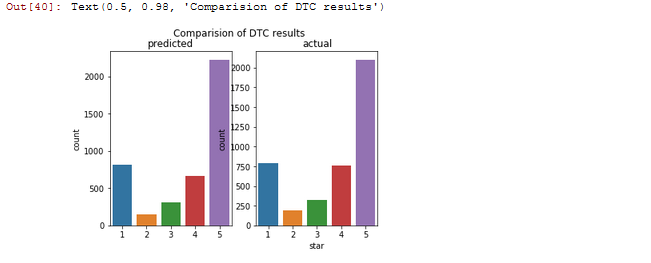
Observations:

1. post seeing the results of sentiment analysis we can clearly devise that positive reviews are in abundance
2. neutral reviews or reviews with scores 2 and 3 are less, so model finalization must depend on accuracy and precision to predict these values/ ratings
3. if we see the sentiments that are predicted we can see some comments even if are having 1 rating are predicted/ classified as neutral
4. we can see some words which suggest bad reviews in first 2 wordclouds
   1. bad
   2. issue
   3. problem
   4. waste
   5. worst
   6. replacement
   7. don't
   8. don't buy
   9. doesn't
   10. poor
5. Looking at the words we get in wordclouds of list4(rating4) and list5(rating 5) we can clearly see some positive words as follows
   1. good
   2. best
   3. great
   4. nice
   5. awesome
   6. better
   7. excellent
   8. amazing

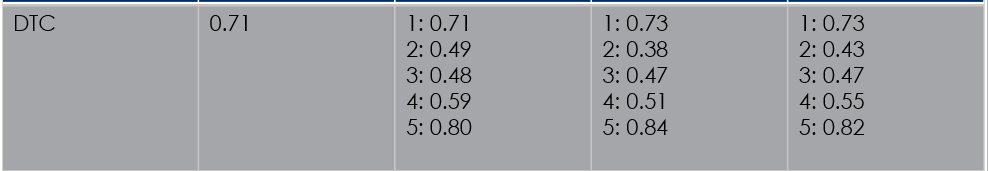
**RESULTS**

* **Decision tree classifier:**

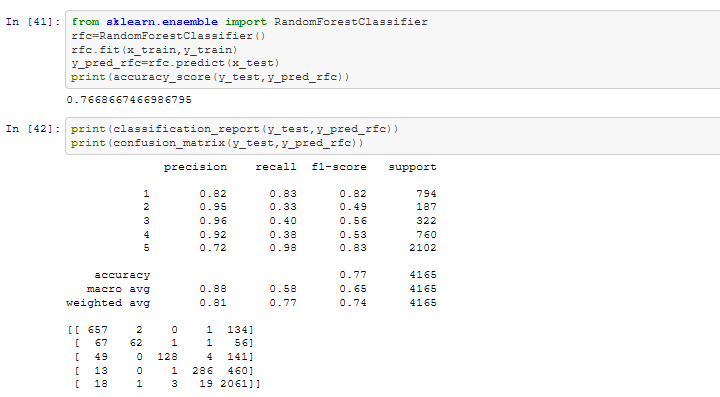


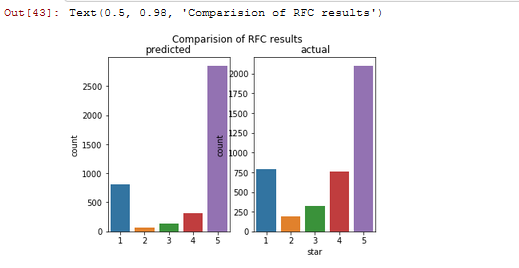




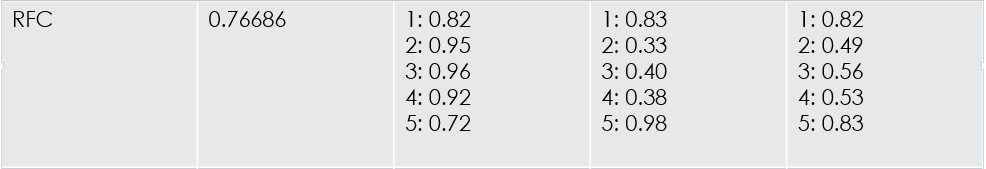


* **Random Forest Classifier:**

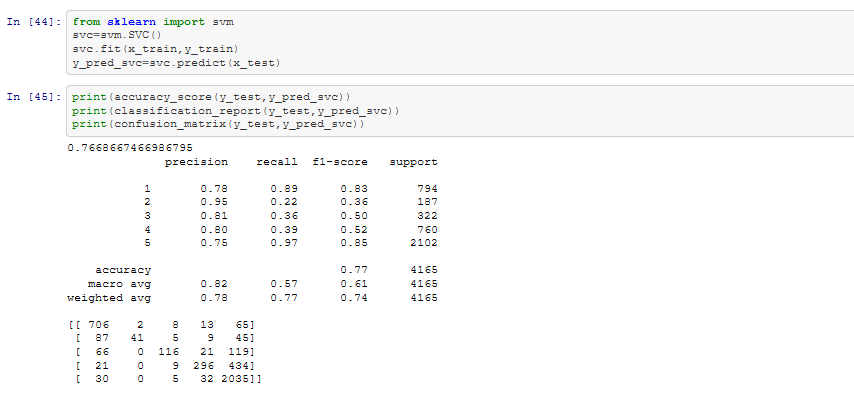


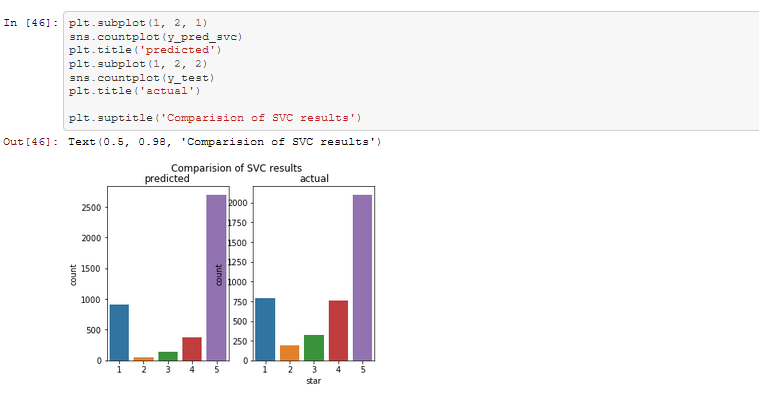




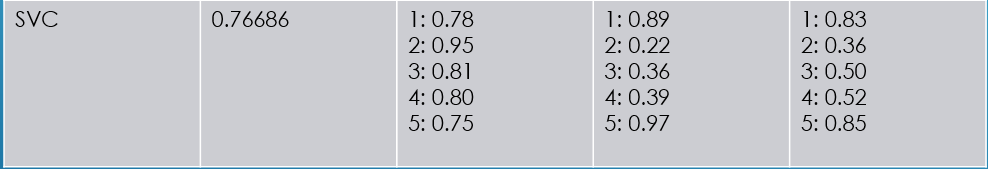


* **SVC:**









Looking at the ability of classifying less abundant data SVC model was classified and pickled.

**CONCLUSION**

* Key Findings and Conclusions of the Study

Sentiment analysis proves to be a better approach when dealing with multiple classes in text.

Comparison of predicted and actual results help greatly to get an idea of model performance.

* Learning Outcomes of the Study in respect of Data Science

Visualization of the frequently occurring words in text data greatly helps to get an idea of how a particular class might look and helps the model to better recognize the provided data.

Data cleaning helps reduce the processing time in case of text data.

Visualization of results must be done to get an idea of model performance.

* Limitations of this work and Scope for Future Work

More data can be extracted to better train the model, but a better model must learn from minimal amount of data which must be looked upon in this project