# Review Score Prediction

**Krithika Ravishankar**
kr33277
krithravi@utexas.edu

**Francesco Leone**
frl263
frrleone@gmail.com

## 1 Introduction

The objective of this project was to develop a machine learning model that can accurately predict the score of a review based only on its text body. To do this we implemented a multinomial Naive Bayes model, a logistic regresion model and compared the results to a baseline model using `nltk`'s VADER (Bird et al., 2009), a pretrained sentiment analysis model. We also implemented a number of data normalization methods to refine our prediction results. On top of this, we investigated how the models performed with regard to reviews of specific products and product categories.

The motivation for this work stemmed from a keen interest in exploring the possible refinements that can be made to the models and their predictions, as well as an abundance of available data.

## 2 Related Work

The data we used was aggregated in the paper *The Multilingual Amazon Reviews Corpus*. The authors created the dataset to provide accessible and reliable training data for multilingual text classification, which was not as readily available at the time; other datasets, such as the `YELP open dataset`, were periodically updated and unreliable for repeatability of results. The reviewers used the cased multilingual BERT base model for classification and the mean absolute error (MAE) as their primary evaluation metric. The MAE was chosen, as opposed to using the prediction accuracy, in order to more accurately evaluate the system's performance. The evaluation penalized predictions farther away from the true score.

The authors performed two experiments. First, they performed training and classification for reviews of the same language. They distinguished between two classification tasks: fine grained, in which they predict the review score on a scale of 1-5, and binary, in which they predicted if the review was either 1-2 or 4-5 stars, omitting 3 star reviews. They then performed the same classification tasks across multiple languages.

```
{
  "review_id": "en_0163606",
  "product_id": "product_en_0898959",
  "reviewer_id": "reviewer_en_0859914",
  "stars": "1",
  "review_body": "Totally uncomfortable",
  "review_title": "One Star",
  "language": "en",
  "product_category": "drugstore"
}
```

Figure 1: Sample English-language entry

## 3 Data

We used The Multilingual Amazon Reviews Corpus, the `amazon_reviews_multi` dataset, for both training and testing. We only used the English language corpus within this dataset, which is comprised of reviews posted between November 1, 2015 and November 1, 2019 in the American Amazon marketplace.

Each entry contains an anonymized product ID, anonymized reviewer ID, review text, star rating, review title, and the product category as is given on Amazon. Additionally, each reviewer is restricted to 20 reviews and each specific product is restricted to 20 reviews in the dataset to ensure greater diversity in the datasets. All review bodies range between 20 and 2,000 characters in length. Each star rating accounts for 20% of the data points in each language (Keung et al., 2020), for a total of 210,000 entries in each language.

## 4 Methodology

We implemented three different predictive models to identify which performed the best. We used sci-kit learn's logistic regression and multinomial Naive Bayes and compared these results against a baseline model using VADER. We then trained and tested each of these models in two tasks. The first task was on our dataset as a whole, and the second on the reviews within individual product categories. Additionally, we attempted to create ordinal regression models, using mord, but were unable to do so successfully (Pedregosa-Izquierdo, 2015; Muhammad, 2019).

For the task using the entire dataset, we used the original train-test split provided with the data of 200,000-5,000. For the product-by-category analysis the given training, testing and development datasets were all combined together, for a total of 210,000 entries. This dataset was then split by category and shuffled, as the original datasets were ordered by star rating. The first 75% of the category subfiles were used as the train set, and the latter 25% were used as the test set.

In order to use our data we needed to perform a number of clean-up tasks on it, primarily in the form of normalization. In order to lemmatize our data we used the python package spaCy (Honnibal and Montani, 2017). We found that replacing contractions, such as "I'll" → "I will," improved the results of our lemmatization. We removed commas and periods, and kept other punctuation marks such as "!", since they could indicate sentiment. Additionally, as part of our lemmatization, we chose to remove stopwords from our data, eliminating tokens common across reviews that did little to improve predictability. In order to vectorize our word tokens we used sklearn's (Pedregosa et al., 2011) CountVectorizer and TfidfVectorizer, but found that there was no significant difference between the two. As such, the results presented in this paper are using the CountVectorizer. Matplotlib (Hunter, 2007) was used for graphing. pandas (pandas development team, 2020; Wes McKinney, 2010) and NumPy (Harris et al., 2020) were used in data analysis.

## 5 Results and Discussion

We compared our three models to examine their predictive ability in the two tasks using this dataset. In order to do so, we considered the accuracy, F1 scores and $R^2$ scores of the logistic regression, multinomial Naive Bayes and VADER models. Since VADER returns a sentiment score between -1 and 1, we mapped its prediction to integers in the range [1, 5].

### 5.1 Accuracy Across Models

Accuracy was the first metric we assessed when analyzing the results of our models. From figure 2 A, across the entire dataset the VADER model achieved an accuracy of 0.33, while logistic regression reached 0.47 and Naive Bayes reached 0.46. This discrepancy was consistent for the entire dataset (marked entire on the plot, farthest to the left) as well as the individual product categories. The exception to this is the personal care appliances category. This can be explained by the small number of reviews present in this category (75) as compared to others (apparel: 15,971, home: 17,679, etc.). This small number is the reason the results for this product category are outliers in each of the metrics we use to assess our models, and as such should be disregarded.

Our logistic regression and Naive Bayes models were very close in accuracy for both tasks. For the dataset as a whole, the logistic regression model had a slightly higher accuracy (0.47 vs. 0.46). Across product categories, Naive Bayes tended to have slightly higher accuracy. The difference is insignificant in the majority of individual categories except digital ebook purchase (LR: 0.41 vs. NB: 0.46), watch (0.43 vs. 0.32) and jewelry (0.44 vs. 0.41).

### 5.2 F1 Score Across Models

Looking at the F1 scores of our models in figure 2 B, we can see that the results follow the same trend as the results for accuracy. Namely, the VADER model severely underperformed compared to the logistic regression and Naive Bayes models, which performed very similarly. VADER achieved an F1 score of 0.33, while logistic regression achieved 0.47 and Naive Bayes achieved 0.46. Again, the Naive Bayes model slightly outperformed the logistic regression model in the majority of individual categories, but performed slightly worse on the dataset as a whole.

### 5.3 $R^2$ Across Models

From figure 2 C, our models all displayed very weak $R^2$ values, indicating that the fit of each of the models was far from perfect. On the dataset as a whole, our logistic regression and Naive Bayes models had R$^2$ values of 0.28 and 0.23 respectively.

This shows that the logistic regression model had a better overall fit than Naive Bayes. However when looking at individual product categories we can see that the results are more complicated than that. Naive Bayes had a positive R$^2$ value for all categories except personal care appliances and video games, however logistic regression had negative values for the digital ebook purchase and digital video download categories as well. Our VADER model had a negative R$^2$ value in most categories, as well as the dataset as a whole. The negative R$^2$ values indicate that the model performed worse than a horizontal line.

### 5.4 Distance between prediction and true value

An issue we encountered with our evaluation is that for a 5-star review predictions of 1 and 4 stars would be scored equally, when the latter prediction is far more accurate. To account for that inequality in scoring, we used the distance between the prediction and the true value as a metric. Ideally, this line of analysis can be performed using an ordinal regression model. After trying to implement this, we found that the results were very poor and decided it was outside the scope of this project to properly implement and explore.

Figure 2 D shows the average proportion of predictions off by $N$ stars across the product categories. From the graph, we can see that the logistic regression and Naive Bayes models performed similarly, with a majority of predictions being within 1 stars of the true value. In comparison, the VADER model had a significantly smaller proportion of correct predictions (distance of 0) and a significantly larger proportion of distance of 2 predictions.

### 5.5 Evaluation and Problems

The results we were able to achieve with our models were underwhelming in each of our evaluation metrics. Our logistic regression and Naive Bayes models were able to correctly predict less than 50% of the testing data, which was far lower than we had initially hoped for. Especially surprising were the results from VADER, which severely underperformed relative to our expectations, with an accuracy rating of 33% on the overall task.

We found that VADER performed better using lemmatized text, however its documentation states that it should perform better without. One potential explanation for this and its poor performance in predicting review scores is that it was trained on social media posts, which hindered its performance analyzing text from other sources. Social media posts and product reviews are quite distinct in a number of ways, including tone, language content and general structure.

## 6 Team Structure

Both team members were interested in working on all aspects of the project, so the labour was left without clear divisions.

## References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*. O'Reilly Media Inc.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, et al. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Muhammad. 2019. Simple trick to train an ordinal regression with any classifier.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and others. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Fabian Pedregosa-Izquierdo. 2015. *Feature extraction and supervised learning on fMRI : from practice to theory*. Theses, Université Pierre et Marie Curie - Paris VI.

The pandas development team. 2020. pandas-dev/pandas: Pandas.

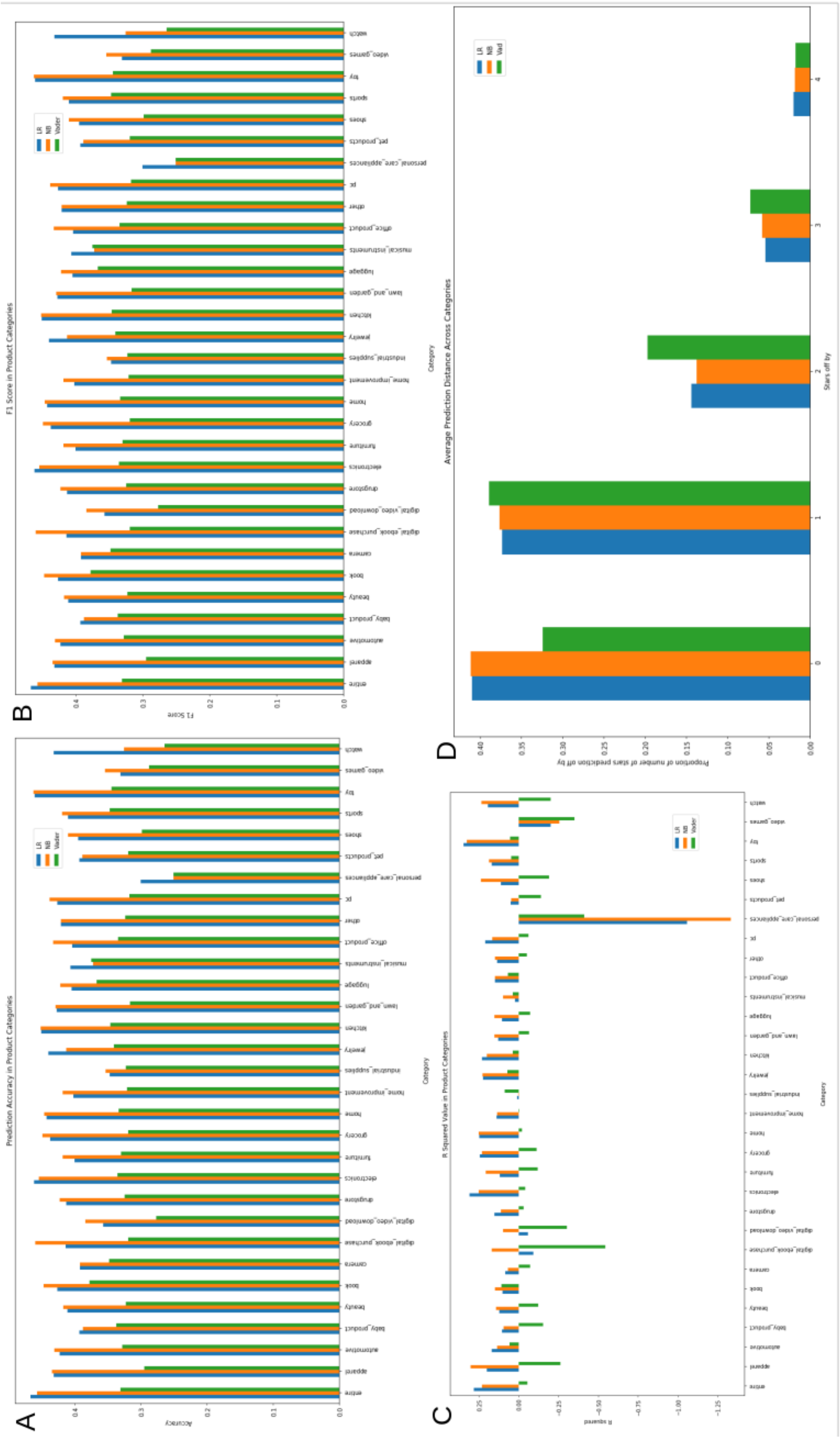

Figure 2: Metrics and data analysis