Seng 550 Project Report

Group 9

Topic: Amazon Reviews Sentiment Analysis

Braden Thompson - 30039343
Hatem Muhamad - 30038416
Md Rashik Hassan - 30048022
Quinn Ceplis - 30027148
Youstina Attia - 30038658

Abstract

Customers buying products on amazon often leave reviews of the products they have purchased as well as rating the product on a five star scale. This paper analyzes Amazon review data to predict 1) the sentiment and 2) the rating of future reviews. For both predictions, four machine learning algorithms are trained and compared using quantitative metrics to determine the best performing model and hyperparameters. For sentiment analysis, it is determined that the best model is Logistic Regression with SGD, while the best model for star prediction is Multi-class Logistic Regression. These findings can be used to help small businesses compare reviews on their own ecommerce sites, to reviews on Amazon and prioritize product lines.

Introduction

Customer analytics metrics have been monopolized by big tech. During the COVID-19 pandemic, online shopping use grew massively due to lockdowns and consumers feeling safer ordering products from their own home. This led to Amazon's profits soaring to all time highs. Individual vendors have slowly been forced to switch from selling on independent websites to selling on Amazon. This means a portion of their potential revenue is going to Amazon. To help recapture some of this revenue, vendors would benefit from being able to gauge customer feelings to prioritize their product lines.

The algorithms this paper provides would play an important part in vendors regaining their agency. Namely, it would allow them to prioritize their product line both on and off Amazon. This could give them data they need to improve and attract customers to their own retail services. In turn, this would keep more currency in circulation instead of in Jeff Bezos' investment portfolio–a much healthier scenario for local economies.

Amazon has introduced review rating systems where a customer who has bought the product can put a rating out of 5. EBay has star rating systems as well but it also has the functionality to place bids and asks for a certain product and the ratings only exist for the sellers on the platform. Comments are more related to the seller rather than the customer.

This paper aims to answer the following questions:

1) For sentiment analysis, what is the best classification model based on the available data?
2) For star prediction, what is the best multi-class classification model based on the available data?

For context, what is meant by "best" in terms of classification? This paper defines best as the model with the highest precision, then accuracy, F1 score, and recall. The elimination of false positives is paramount for finding sentiment and star rating consistently.

This paper demonstrates that Logistic Regression with SGD performs best for sentiment analysis of reviews, and that Multi-class Logistic Regression performs best for star prediction based on review text.

Background and related work

To understand this report, one should have a solid understanding of binary classification and multi classification concepts. In basic terms, binary classification algorithms group data into one of two categories. In our case, binary classification is used for sentiment analysis where the two categories are positive and negative sentiment. Multi Classification algorithms are machine learning algorithms that classify data into multiple categories. For this report, multi classification algorithms are used for star prediction, where each star is a category.

It would also be helpful to have an understanding of the tools used in the experiment. Few of the notable mentions were Jupyter Notebooks, Apache Spark, DataBricks, Natural Language Toolkit, and MLFlow. In order to run our notebook, a Spark cluster is required.

Methodology

To set up the experiment, the Amazon reviews dataset is read into a PySpark DataFrame in order to be preprocessed. A sample review is provided to give insight into the data being worked with:


```
{
    "overall": 5.0,
    "verified": true,
    "reviewTime": "08 22, 2013",
    "reviewerID": "A34A1UP40713F8",
    "asin": "B00009W3I4",
    "style": {
        "Style:": " Dryer Vent"
    },
    "reviewerName": "James. Backus",
    "reviewText": "I like this as a vent as well as something that will keep house
                   warmer in winter. I sanded it and then painted it the same color
                   as the house. Looks great.",
    "summary": "Great product",
    "unixReviewTime": 1377129600
}
```


Figure 1: Raw json review data.

All columns besides “overall” (the 5-star rating) and “reviewText” are dropped from the dataframe. Other columns don’t need to be maintained, as measurement isn’t focused on predicting sentiment on certain types of products. A new column is added to the dataframe to provide simple representations of the sentiment for classifiers. In this column, “1” denotes a positive review, and “0” denotes a negative review. A positive review is a review with a score

greater than or equal to 4 while a negative review is a review with a score less than 4. We decided to use a higher cutoff for positive scores to account for the bias in reviews that leads reviewers to tend to only leave very high or very low scores. The review text is then converted to lowercase, punctuation and stopwords are removed, and the text is stemmed and tokenized.

For sentiment analysis, the text vector alongside a binary sentiment label was used to train the models. Due to this simple distinction, binary classifiers were used. These include SVM with Stochastic Gradient Descent (SGD), Naive Bayes, Logistic Regression with the limited-memory Broyden–Fletcher–Goldfarb–Shanno optimization algorithm (LBFGS), and Logistic Regression with SGD.

For star reviews prediction, we used the text vector alongside the overall star rating to train our models. Due to the ability to rate a product with an integer between 1 and 5, inclusive, we decided to use multi-class classification models. We decided to use Multi-class Logistic Regression, Decision Tree, Random Forest, and Naive Bayes.

Further information on models used and hyperparameters tuned can be seen below:

Table 1: Star prediction model with tuned hyperparameters.

Model	Hyperparameters Tuned
Multi-Class Logistic Regression	Regularization Parameter, Elastic Net Parameter
Decision Tree	Max Depth, Max Bins
Random Forest	Max Depth, Max Bins, Number of Trees
Naive Bayes	Smoothing

We were able to use cross-validation methods for hyper parameter tuning of each method. The data was initially split into training and test data with an 80% and 20% split respectively. The training data was further split for cross validation with an 80% used for training. The remaining 20% was used for hyperparameter tuning. Once each model was built and the hyperparameters were tuned, we used our testing data to evaluate each model.

In order to develop our experiment, we began by setting up on DataBricks and used a small sample of the data to quickly build and test our models. Evaluators were created to compare results for each of the four models in star prediction and sentiment analysis. The following performance metrics were used to evaluate the models: Accuracy, Precision, Recall and F1 score.

Results

The study consisted of two separate predictions; the sentiment of a review and the reviews star rating. The results section compares the Accuracy, F1 score, Recall and Precision of four different models for each type of prediction.

Sentiment Analysis Results

Table 2: Performance metrics for sentiment analysis models.

Model	Accuracy	F1	Recall	Precision
SVM with SGD	0.86	0.84	0.86	0.85
Naive Bayes	0.83	0.84	0.83	0.85
Logistic Regression with LBFGS	0.83	0.83	0.83	0.84
Logistic Regression with SGD	0.87	0.86	0.87	0.86

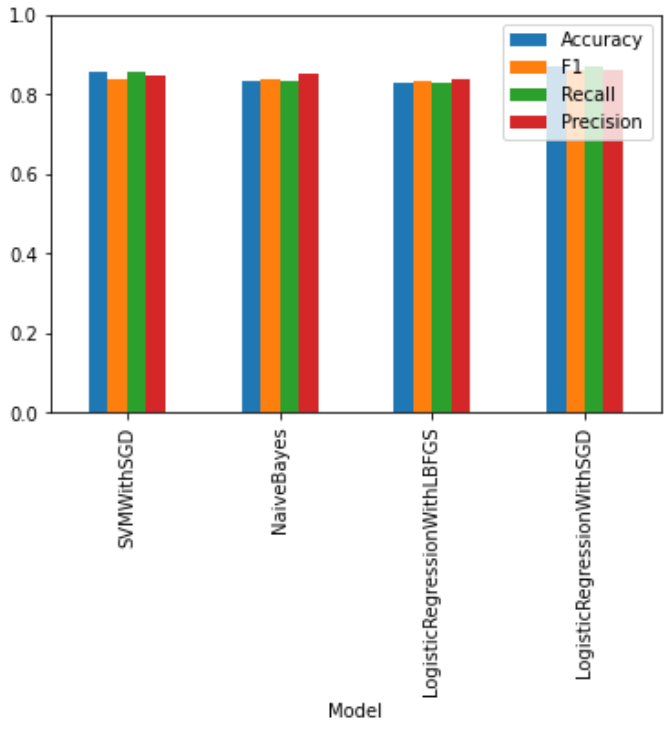

Figure 2: Performance metrics for sentiment analysis models.

Star Prediction Results

Table 3: Performance metrics for star prediction models.

Model	Accuracy	F1	Recall	Precision
Logistic Regression	0.69	0.61	0.69	0.59
Decision Tree	0.69	0.57	0.69	0.56
Random Forest	0.68	0.56	0.68	0.55
Naive Bayes	0.08	0.04	0.08	0.03

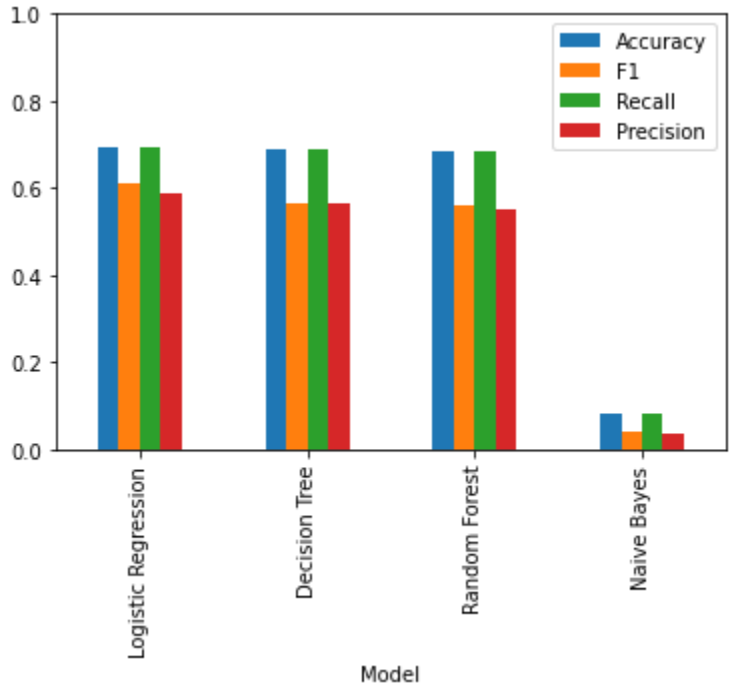

Figure 3: Performance metrics for star prediction models.

Conclusions and future work

For modeling the overall sentiment of a review, the best performing model was Logistic Regression with SGD with an accuracy of 0.86, F1 score of 0.84, recall of 0.86 and precision of 0.85. The other models were not far behind so any of the four models could be used depending on the situation.

Star prediction had a larger difference in results between the models, with Multi-class Logistic Regression performing slightly better than Decision Tree and Random Forest. Naive

Bayes performed significantly worse and would not be a good model to use for predicting results.

For future work we would like to scale our solution to scrape and process reviews directly from Amazon in real time. These models could also be used on other e-commerce sites with reviews that might not include a star rating to help prioritize products and gauge customer satisfaction.

Individual contributions

Table 4: Member contributions.

Name	Contribution
Braden Thompson	Data cleaning, sentiment analysis models and writeup
Hatem Muhamad	Star rating cross validation implementation
Md Rashik Hassan	Writeup
Quinn Ceplis	Results analysis implementation and writeup
Youstina Attia	Star rating models, cross validation and writeup

References

[1] Ni, J. (2021). Amazon review data. Retrieved 10 December 2021, from http://deepyeti.ucsd.edu/jianmo/amazon/index.html

[2] Classification and regression - Spark 3.2.0 Documentation. (2021). Retrieved 12 December 2021, from https://spark.apache.org/docs/latest/ml-classification-regression.html

[3] ML Tuning - Spark 3.2.0 Documentation. (2021). Retrieved 12 December 2021, from https://spark.apache.org/docs/latest/ml-tuning.html

[4] Multiclass Text Classification. (2021). Retrieved 9 December 2021, from https://www.ateam-oracle.com/post/multiclass-text-classification-crossvalidation-with-pyspark-pipelines

[5] Extracting, transforming and selecting features - Spark 3.2.0 Documentation. (2021). Retrieved 5 December 2021, from https://spark.apache.org/docs/latest/ml-features

[6] Data Preparation. (2021). Retrieved 5 December 2021, from https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/3923635548890252/1357850364289680/4930913221861820/latest.html