Comparison of Text Mined Customer Review Rating Prediction Models

CONTRIBUTORS: ADAM CLEAVER BENG, ING. LUKAS TOPINKA, M.SC.
DATASCIENTEST SUPERVISOR: MAËLYS BASSILEKIN BLANC

Abstract

This report compares methods of datamining the text from Customer Reviews and the Accuracy of Rating predictions from 1 to 5 stars. This report satisfies the business need to identify the best model to accurately classify Customer reviews into a rating. The report aims to save future studies time in computation comparisons by finding the best preprocessing methods and the best models to classify reports by rating.

The findings of this report could be implemented in several use cases:

- Generate an automated rating system, which offers customers a pre-generated star rating based on the content of their review.
- Identify Reviews which are incorrectly rated, to remove them from further analyses or submit them to further analyses.
- Classify reviews which are no longer associated with their original rating, or reviews which are not part of a rating system.
    - Sort reviews for customer service customer response Management to organize reviews by priority.
    - Classify reviews for automated CRM Tools enabling automated responses to reviews based on predicted rating.
    - Identify and handle reviews which are given in bad faith, by identifying text that often features in bad faith reviews or that does not match typical review text associated with ratings.

Introduction

This Project takes review text from Amazon reviews and uses various machine learning models to explore how accurately the rating can be predicted by machine learning models. This serves as an introductory investigation into neuro linguistic processing (NLP) to create sentiment analyses. The project looks at the specific strengths of regression models and classification models against review rating accuracy.

While the project ultimately did not generate a highly precise tool for predicting amazon review sentiments, a lot of learnings were taken from the various stages of preprocessing and machine learning models utilized. Enabling scrutinization of the classification and regression models and their benefits in potential use cases.

Furthermore, the project builds a solid foundation for further investigation and suggests content to support a potential roadmap for further preprocessing techniques and more complex machine learning or deep learning models for future investigations.

Exploration

Data Quality

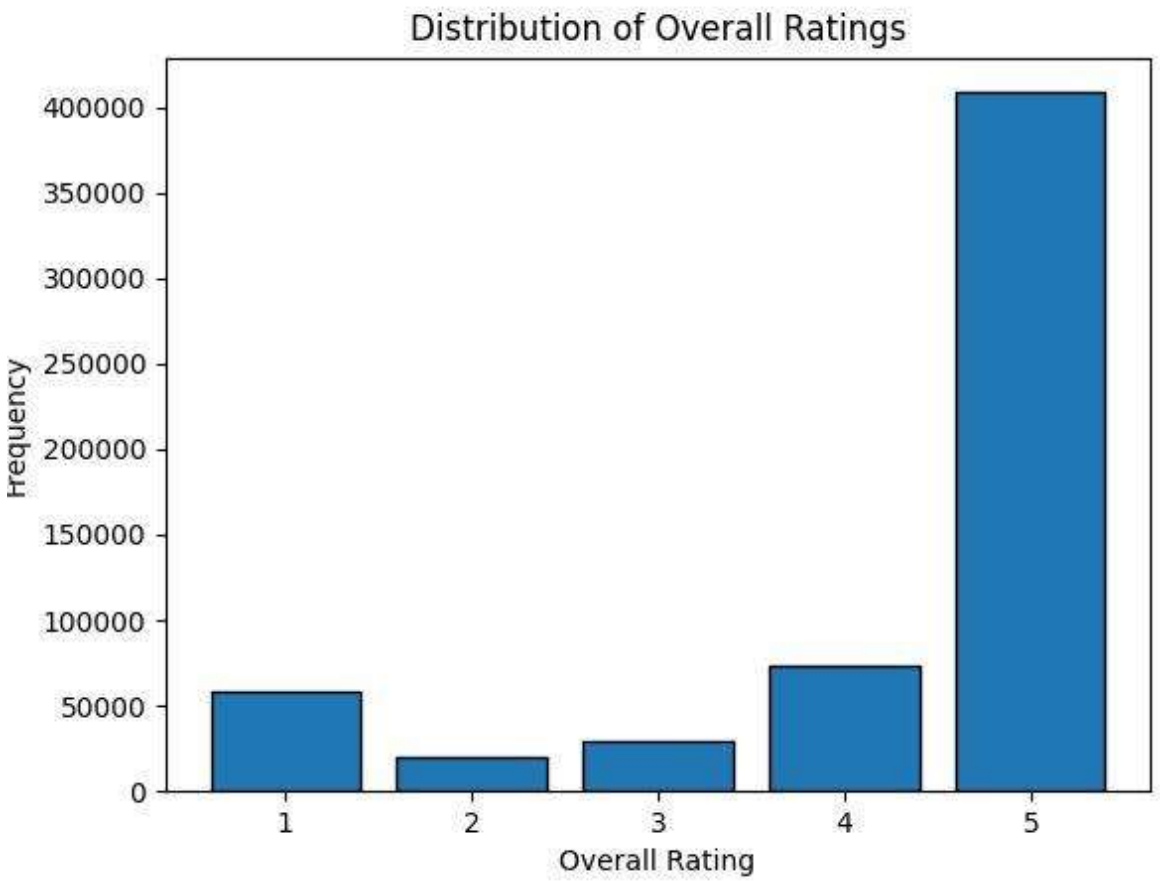
This report analyses Amazon review data from the Appliances Category, the data was originally collected in 2014 and most recently updated in 2018. Though the data has been parsed for NLP usage, extra Data Cleaning and preprocessing is required to enable the variety of modelling techniques that will be tested in this report.

The feature variables will be derived from the review text of each review and the target variable will be the rating from 1-5 stars. The data quality is verifired by checking for duplicates and NaN values, various tokenization and vectorization techniques are implemented to enable the machine learning models to accurately parse the data.

Other columns can also be processed to enable further investigations and project expansions. Though these extra deliverables will depend on favorable project scope and timeline.

Th full data Quality report can be viewed in the supporting report: Text Review Sentiment Analysis using Classification Modelling vs Regression Modelling - Data Quality Report

Fig 1. Bar chart displaying target Variable imbalanced distribution



Machine Learning Modelling

In this report several Classification and Regression Models are compared using data from amazon reviews as detailed in the Data Quality Report. The models are detailed in the following sections. The goal of the modelling is to accurately analyze the sentiment of the text review and predict the rating that the user submitted alongside their review.

Regression Models

The Models being compared include:

- Linear Regression
- Lasso
- Ridge
- ElasticNet
- Histogram Gradient Boosting Regressor

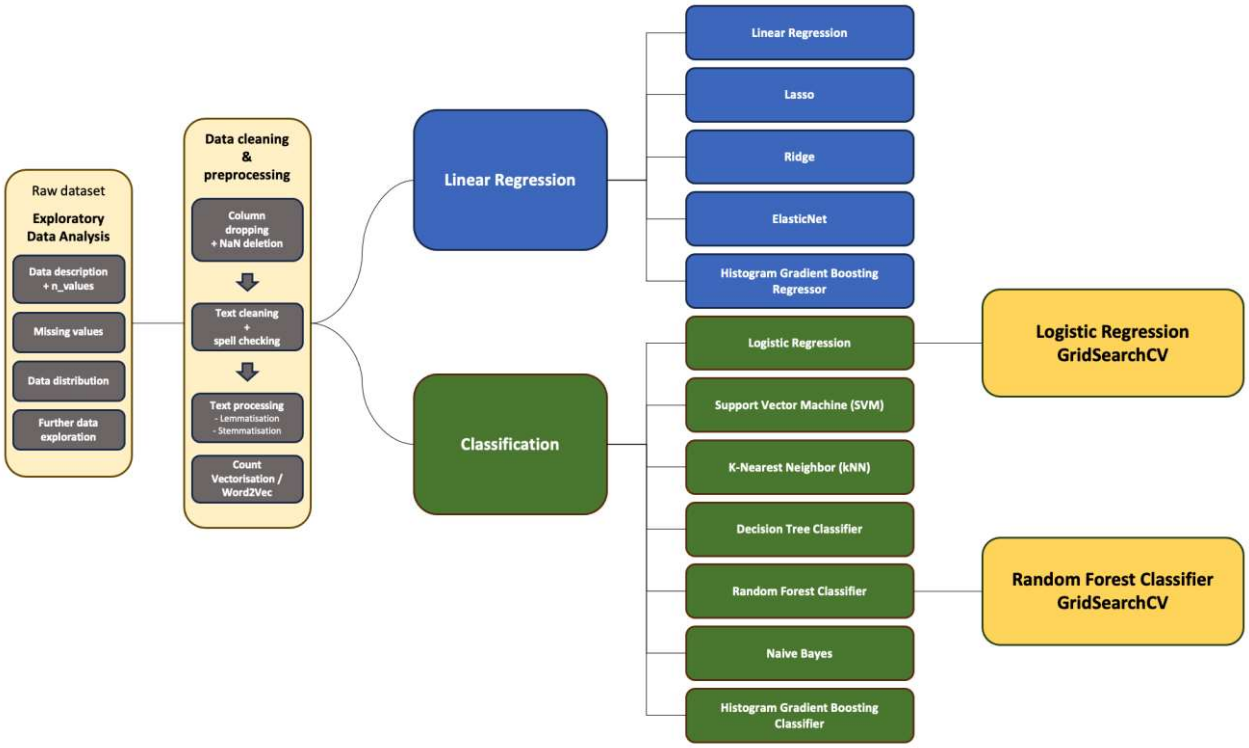
Classification Models

The Models being compared include:

- LogisticRegression
- Support Vector Machine Classification
- K Nearest Neighbors
- Decision Tree Classifier
- Random Forest Classifier
- Naive Bayes
- Histogram Gradient Boosting Classifier

All Regression models are run through grid search cross validation to find the best arguments and sampling methods. Due to time constraints, only the best performing classification models were selected for cross validation. This created time at the end of the project to explore further improvements to the modelling process.

Fig 2. Project Overview – Modelling Schema



Hypothesis

The ratings target variable consists of integers that follow a linear related scale. Theoretically these reatings should be able to return a reliable score in regression models as well as classification models. This project aims to compare the accuracy of this statement using a comprehensive number of preprocessing and machine learning modelling techniques.

The project compares the best accuracies of the machine learning models detailed above using multiple preprocessing methods. The expected result is a similarity between the accuracy and error between the classification models and the regression models.

To measure this variance between models, the following scores will be taken from the model predictions:

- Accuracy: The number of correct predictions as a percentage.
- Precision: The number of correct positive predictions as a percentage. *
- Recall: The number of correct positives as a percentage with respect to false negative predictions. *
- F1 Score: A metric that combines Precision and Recall providing a balanced measure*
- R Squared: The goodness of fit indicator where values between 0.75 and 1 indicate a strong regression. **
- Mean Squared Error: Error indicator which penalizes large errors.

*Precision, Recall and F1 Score are all calculated using a weighted average, the support for each class impacts the score.

**When R Squared returns a negative value, this indicates an incredibly poor regression, this is a known bug of the scorer.

Considerations

The classification models return exact categories 1-5, the regression models will only return a continuous series of results. This has its benefits; the spread of the data can be analyzed in greater detail than the classification reports. The visibility of the data can be used to identify edge cases and see how noise in the categories behaves, which is not possible in the classification reports.

To maintain a fair measure between the two types of models the regression models are dichotomized before scores are taken. This dichotomization takes the form of a simple Pandas Series cut between categories at values 1.5, 2.5, etc.

Since the categories are related by a linear scale it is possible to take a nominal mean squared error from both the classification and dichotomized regression results. It is also possible to take a more accurate mean squared error from the continuous regression results, but comparisons were drawn from the dichotomized results to maintain fairness.

The HistGradientBoostingRegressor (HGBR) and the HistGradientBoostingClassifier (HGBC) were chosen over the GradientBoostingRegressor (GBR) and the GradientBoostingClassifier (GBC) to reduce runtimes whilst operating on reasonably sized datasets.

Results

Summary

The results show that Classification techniques are more suited to the data predictions, however none of the results are strong enough to clearly define the models as better than a basic model. The best model produces only 77% accurate results with a large mean squared error of 0.996 on the test data. The average std deviation covers 1 class to either side of the correct class and confusion matrices analysis reveals that this accuracy largely relies on assigning a large percentage of the data to class 5.

77% Accuracy is only 8% better than a basic model that returns class 5 for every prediction and so this model cannot be considered very strong. The strength of the model accuracies is not sufficient to support the potential business cases outlined in the introduction reliably.

The full results are documented in the supporting document: Text Review Sentiment Analysis using Classification Modelling vs Regression Modelling – Modelling details and Results Report

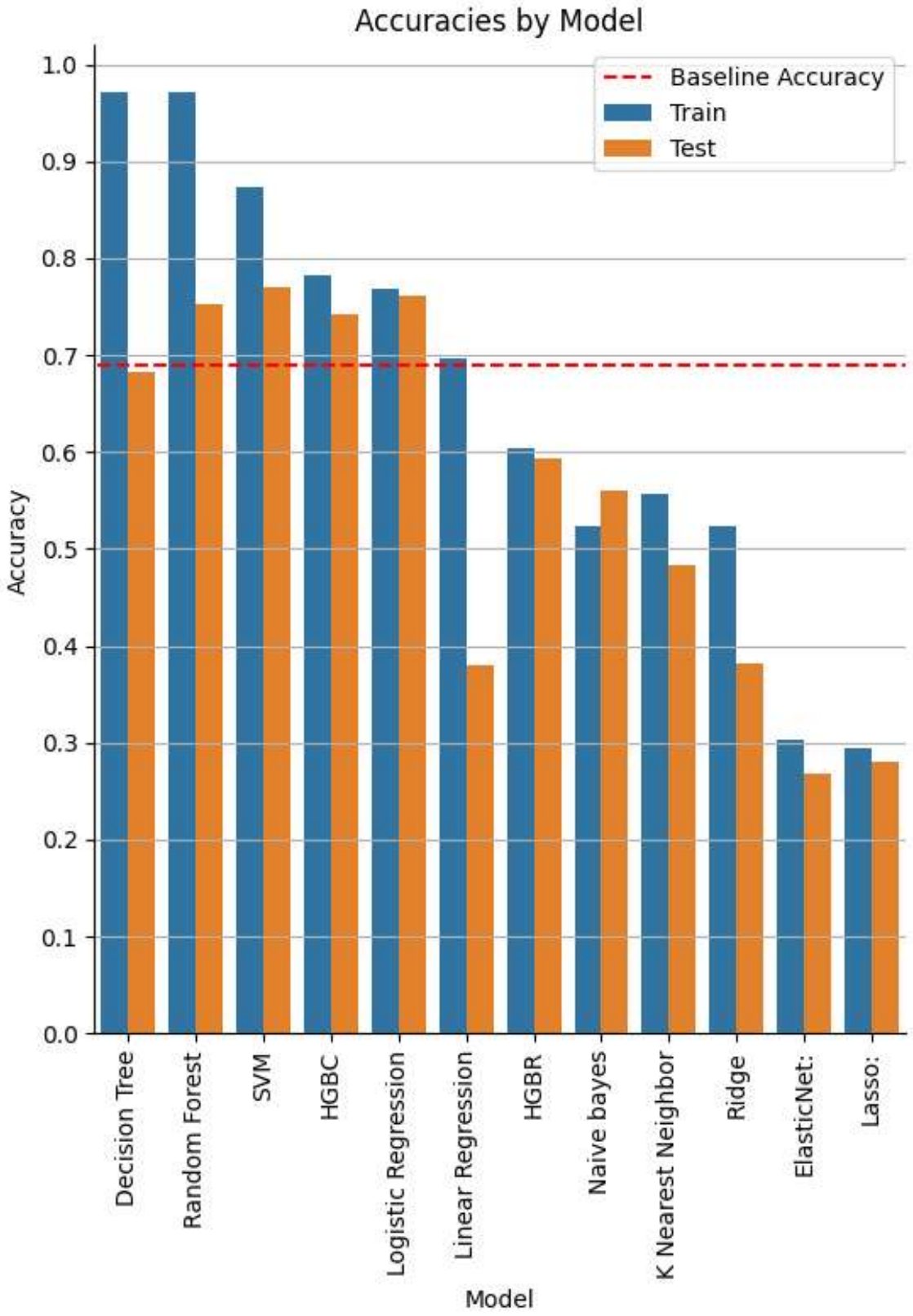
Table 3. Best Regression Results

Model	Token Method	Vector Method	Sampler	Mean test accuracy	Mean test precision	Mean test recall	Mean test f1 score	Mean test r2 score	Mean Test mse
HGBR	lemmatized Train Data	TFIDF	None	0.604	0.740	0.604	0.642	0.563	0.747
HGBR	lemmatized Test Data	TFIDF	None	0.593	0.728	0.593	0.632	0.509	0.829

Table 4. Best Classification Results

Model	Token Method	Vector Method	Sampler	Mean test accuracy	Mean test precision	Mean test recall	Mean test f1 score	Mean test r2 score	Mean Test mse
SVM	lemmatized train data	TFIDF	None	0.874	0.882	0.814	0.856	0.762	0.403
SVM	lemmatized test data	TFIDF	None	0.770	0.725	0.77	0.709	0.410	0.996
Logistic Regression	lemmatized	TFIDF	None	0.769	0.720	0.769	0.715	0.408	1.004
Logistic Regression	lemmatized Test Data	TFIDF	None	0.762	0.703	0.763	0.707	0.376	1.053
Random Forest	Lemmatized train data	Count Vector	None	0.971	0.971	0.971	0.970	0.955	0.076
Random Forest	Lemmatized test data	Count Vector	None	0.753	0.708	0.753	0.684	0.267	1.24

Fig 5. Bar chart displaying all results in order of best accuracies (train or test)



Many models show indications of overfitting, with higher train scores posted compared to test scores. The classification model Logistic Regression shows the least amount of variance between the train and test scores, indicating that the model has found its best accuracy with the dataset in its current form.

Sampling methods to reduce the impact of target imbalance were ineffective, so too were the attempts to alternately preprocess the data with the Google Word2Vec model.

Specific Review Analysis

A demonstration model was produced that returned the class of example reviews, including new reviews from user input. The model predicted the review class with the percentage confidence in the chosen class in the case of classification models and returning the continuous float class prediction in the case of regression models.

In demonstration the model performed well when predicting reviews which matched class 5 ratings and performed reasonably well when predicting class 1 ratings. However, middling ratings between 2 and 3 were not well predicted.

Testing the model in this way allowed for closer inspection of individual reviews and testing specific review formats that are harder to predict, such as sarcasm.

Table 6a. Model Demonstration Results

ID	Review text	Note	Actual Rating	Prediction Model	Model Rating	Rating Confidence
1	Bought it in August 2017, it croaked in December 2017. RIP.	Random review sampled from the dataset	2	Classification: Logistic Regression	1	100%
				Classification: Random Forest	1	75.7%
				Regression: HGBR	2.48	-
2	Working great as a replacement for my Sharp original	Random review sampled from the dataset	4	Classification: Logistic Regression	5	85.3%
				Classification: Random Forest	5	79.6%
				Regression: HGBR	4.95	-

Table 6b. Model Demonstration Results continued

ID	Review text	Note	Actual Ratin	Prediction Model	Model Ratin	Rating Conf
3	FROM SELLER- I was able to find parts listings, but no other literature. It should be pretty self explanatory though. It is pretty plug and play.... YOU NEED TO TRY YOUR PLUG AND PLAY- AND YOU WILL DISCOVER IT IS NOT SELF A EXPLAINER THERE ARE MANY STEPS. THAT NEED TO BE DONE. IF YOU DO NOT KNOW WHAT YOU ARE DOING – DON'T BELITTLE YOUR CUSTOMER	Random review sampled from the dataset	1	Classification: Logistic Regression	5	29.4%
				Classification: Random Forest	1	65.8%
				Regression: HGBR	2.91	-
4	Closest to the one I had originally used.	Random review sampled from the dataset	5	Classification: Logistic Regression	4	45.2%
				Classification: Random Forest	5	85.1%
				Regression: HGBR	4.22	-
5	It doesn't stay cold enough to safely store food. I've tried every trick, actually considering returning it.	Random review sampled from the dataset	3	Classification: Logistic Regression	1	27.9%
				Classification: Random Forest	3	64.8%
				Regression: HGBR	2.19	-
6	Oh yeah, I love this product, it's GREAT when things break on first use.	Review created to test sarcasm	1	Classification: Logistic Regression	5	91.0%
				Classification: Random Forest	5	73.7%
				Regression: HGBR	4.82	-

Specific Review Prediction Observations

There are several key observations which shed light on areas where the models could be improved and how the models compare.

In this random selection the Random Forest classification model outperforms the other models, however the Logistic Regression and the Random Forest and Logistic Regression had similar total accuracy, precision and recall, so this can be dismissed as a coincidence specific to the datapoints selected for demonstration.

All models performed equally badly on the sarcastic review example, the nuance of sarcasm was lost in predicting the review and all models labelled the sarcastic 1-star review as a 5-star fantastic review. The sarcastic review will be excluded from variance analysis in this report, but this point highlights the need for a more complex model than simple word representation as only a nuanced study of word relationships will have a chance to predict when a text is written sarcastically. This is classically quite difficult for most humans to pick up on in pure text format unless specific emojis or significant context are present.

Review ID 3 is interesting for two reasons, it is the only review to include a quote, which was not factored into the preprocessing methods, it is also the only review to be written in all caps. Caps were removed from the text at the initial tokenization stage; however, this review highlights the importance of an all-caps message. All-caps is a famous format in online culture for being synonymous with shouting and features regularly in meme culture. An all-caps message is more likely to indicate a negative review, and this is a point where the preprocessing could draw inspiration for improvement. Having the text in lower case format is important for matching like words, but as example, a count of capitalized characters could be performed between the removal of html tag text and the lower-case tokenization of the text and returned as an additional feature column. This feature would not be on the same scale as the other count of words features so this would likely introduce a need to scale the features before performing predictions.

Model Confidence

Studying the confidence and variance of the models reveals further insights. The Logistic Regression model failed to accurately predict a review, but it also has the most reactive confidence. When the variance is larger than 1 the confidence is less than 30%, reviews with variance 1 range from 45 – 100% confidence. This differs from the Random Forest model which has similar reviews for all variances, correct or not. The reviews with variance of 1 have confidences around 75-80%, whereas correct reviews range from 65-85%. This could be a representation of the overfitting present in the Random Forest model compared to the Logistic Regression model, which had the least amount of overfitting.

Table 7. Logistic Regression Variance Confidence.

Review ID	Absolute Error	Confidence
1	1	100%
2	1	85.3%
3	4	29.4%
4	1	45.2%
5	2	27.9%
Average	**1.8**	**57.56%**

Table 8. Random Forest Variance vs Confidence

Review ID	Absolute Error	Confidence
1	1	75.7%
2	1	79.6%
3	0	65.8%
4	0	85.1%
5	0	64.8%
Average	**0.4**	**74.2%**

The Regression models return an entirely different confidence as they return a continuous result. The models do not have a built-in confidence function to determine the model confidence, however a confidence can be inferred provided assumptions are followed.

Assumption 1: The model's confidence is linearly related to the magnitude of the variance from the predicted class.

Assumption 2: If the model returns a prediction that matches a class value to 2dp, this is considered 100% confidence. Rounding error is not considered in this inferred confidence.

Assumption 3: If the model returns a prediction that matches a class border to 2dp, this is considered 40% confidence. This represents the linear scale of the prediction, the model cannot choose between the two classes. For example, a prediction of 3.50 is either class 3, or class 4. It could not be class 1, hence 2 out of 5, rather than 1 out of 5.

With these assumptions the confidence can be inferred from the predicted class using this formula:

Formula 9. Regression Inferred Confidence

$$Confidence = Abs(var) \cdot (-120) + 100$$

Where var is the variance from the class.

Using this formula, the confidence from the true class can also be predicted. The equation is built around the closest class only, when applied to other classes the formula will return negative numbers for larger variances rather than a balanced confidence as the classification models would.

Table 10. HGBR Variance Analysis

Review ID	Absolute Error from True class	Inferred confidence of True Class	Absolute error from predicted class	Inferred Class Confidence of Prediction
1	0.48	42.4%	0.48	42.4%
2	0.95	-14%	0.05	94.0%
3	1.91	-129.2%	0.09	89.2%
4	0.78	6.4%	0.22	73.6%
5	0.81	2.8%	0.19	77.2%
Average	**0.986**			**75.28%**

Confidence can be extracted from the regression model for comparison, but it is important to remember that the classification models confidence and the regression models confidence differ in nature. The regression model confidence is a linearly defined distance from class indicator, whereas the classification model's confidence will be derived from the supporting features that match the class. These will generate a combination of confidences that add up to 100% and the highest percentage will be the chosen class prediction.

For example, in Random Forest for ID no 2 the class confidences are:

- Class 1 Confidence: 0.0%
- Class 2 Confidence: 0.0%
- Class 3 Confidence: 0.0%
- Class 4 Confidence: 20.4%
- Class 5 Confidence: 79.6%

Whereas for Logistic Regression for the same review the class confidences are:

- Class 1 Confidence: 0.0%
- Class 2 Confidence: 0.0%
- Class 3 Confidence: 0.3%
- Class 4 Confidence: 14.4%
- Class 5 Confidence: 85.3%

In contrast the confidence derived from HGBR for the same review are:

- Class 1 Confidence: -374.0%
- Class 2 Confidence: -254.0%
- Class 3 Confidence: -134.0%
- Class 4 Confidence: -14.0%
- Class 5 Confidence: 94.0%

The Regression model variance allows for much more scrutiny over the accuracy of the model. The average overall confidence is similar to the Random Forest model's confidence, but the HGBR confidence has a greater spread.

For more detailed analysis of the regression model output, see the Regression Cutting Analysis Section of the supporting document: Text Review Sentiment Analysis using Classification Modelling vs Regression Modelling – Modelling details and Results Report where a larger set of results are inspected, and class distributions are closely scrutinized.

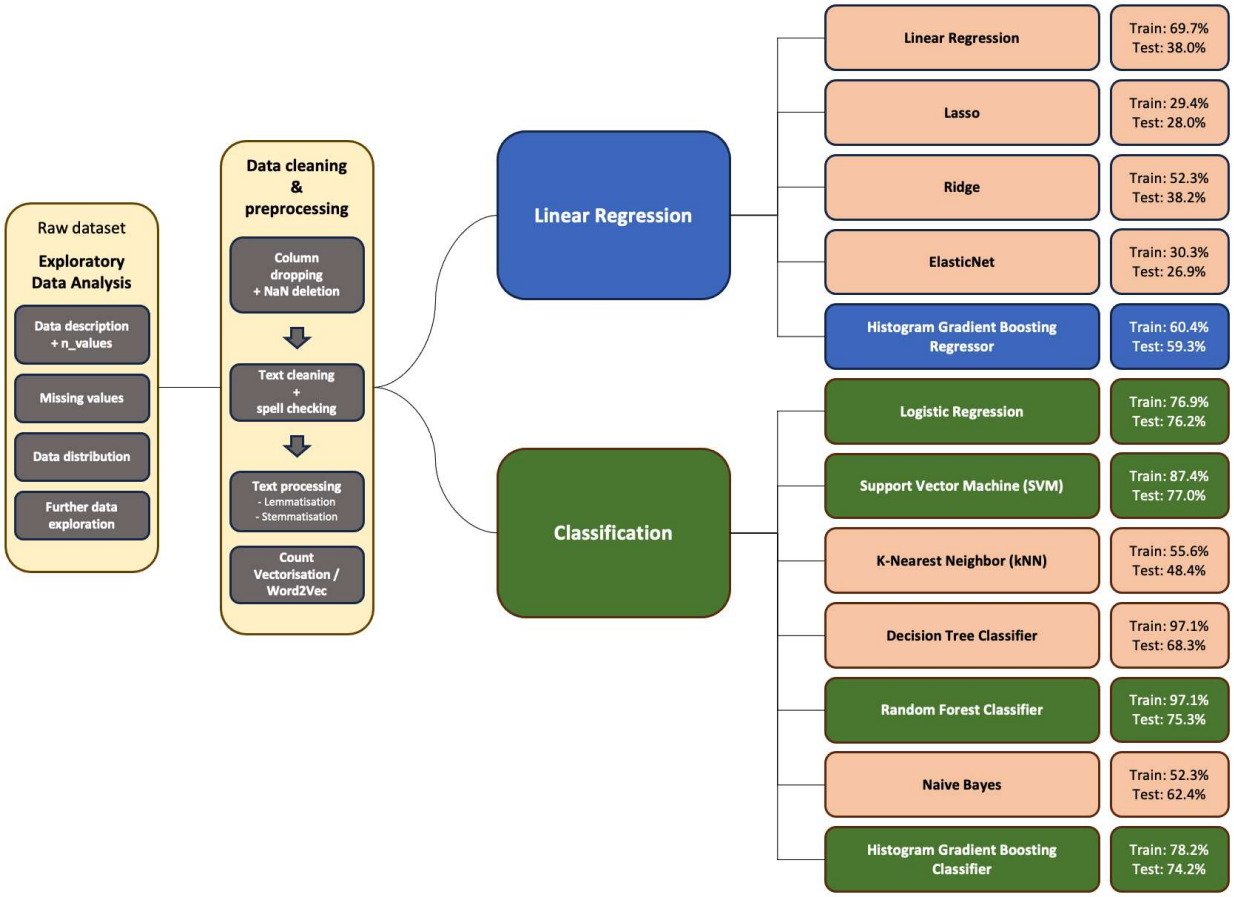
Conclusion

The machine learning models generated in this report failed to generate an accuracy for the suggested business cases. At best they match or marginally improve upon the baseline target accuracy of 69%, at worst the models dramatically reduce the accuracy below that target value.

The best model accuracy was 77%, only an 8% improvement on the baseline accuracy.

These models cannot be used to predict review ratings with a high degree of confidence. Extreme values can be predicted with better confidence than more nuanced middle values. The nuance is lost in the modelling. Further improvements are required to create a suitable model that provides confidence in the results provided.

Fig 11. Final model Schema with best achieved Accuracies



The classification methods produced better scores on Test data for all measurements, disproving the hypothesis that regression models and classification models can be interchangeably used to predict the rating, or sentiment of the customer behind the review.

However, this report has discovered a lot of interesting nuance in the regression models which could have a positive impact on future investigations and modelling.

Further Work

To create a better review sentiment analysis prediction model, combining preprocessing techniques and modelling techniques into a Deep Learning model is a sensible next step to pursue. Additionally, since the input data is heavily imbalanced and the greatest prediction improvements were seen by increasing the size of the dataset, it could be wise to expand and diversify the dataset by sampling from other amazon categories to generate more distinction, or by using a big data methodology to enable the models to be trained on datasets with millions of records. This will need more time devoted to the computation of models.

In the case of simpler machine learning methodologies, two areas for improvement were identified:

- Improving Preprocessing, by implementing feature reduction
- Improving Modelling, by using more advanced modelling Techniques.

Considering the poor results, some additional work was undertaken as a contribution to the next steps of the project. These investigations were heavily time constricted, so only the 'low hanging fruit' options were investigated.

Preprocessing Improvements

Recursive Feature Extraction

The greatest potential improvement in preprocessing lies in the perceived removal of diffusion. Sampling methods often worsened model metrics, and so it can be assumed that either the data contains a lot of unimportant features which are being falsely replicated, few important features which are not being fairly replicated due to the accompanying noise, or a combination of the two. As such a feature extraction process should help the models to extract the relationships between the most important features.

Ideally, a Recursive Feature Elimnation (RFE) with Cross Validation (RFECV) could be run on the vocabulary to identify the words that have the greatest validated impact on the results. However, the RFECV takes a long time to run, each step trains the chosen estimator against the split dataset, tests the results against the remainder of the dataset and all steps are repeated by the number of cross validations. On top of this there is some computational conglomeration time to consider. For a detailed study this will take an extremely long time, more time than was realistically available in the tenure of this project.

Instead, a simple RFE was run on the entire dataset by quarters. The chosen estimator was the Random Forest Estimator which featured in the top 3 best test accuracy scores and has a reasonable computational time. A step of 1000 was implemented with a desired number of features set to 3000.

The 4 quarters produced a list of most important words which were merged to create a list of 3922 unique features. A significant reduction from 28,822 features to 3,922.

The extracted features were parsed through the custom lemmatize function, instead of using English stop words as a no-go gauge, the function was inverted, and the important feature list was implemented as a go through gauge. In this way the model only considered words that were deemed important by the RFE process.

The reduced feature Random Forest model metrics did not improve. The accuracy score dropped to 67.8% and the mean squared error score increased to 2.30. This indicates that this feature selection has hurt the modelling process more than it has improved it as the accuracy has now fallen below the baseline accuracy.

It is most probable that the RFE was applied overzealously due to time constraints, possibly reducing the feature variables too harshly, so that many records had little to no data to build a prediction from.

In hindsight, the Random Forest may have had the best accuracies, but a regression model may have been a better choice, as the detailed negative mean squared error could have been implemented as the important reduction scorer and the detailed output may have produced a more reliable list of important features.

Customizing the Word2Vec model

Another potential avenue is the revisiting and customizing of the google word2vec vectorizer. This vectorizer measures whole sentences and provides a relational summary of the sentence. Customizing the vectorizer to this sentiment analysis use case could yield a substantial score improvement. Theoretically, the model should be able to detect the connections between the words featured in the review that other modelling techniques may miss. For example, 'not bad' is a positive or at least average phrase but the individual words could both potentially score a review negatively. The word2vec vectorizer should see past the unique words and consider the relationship between them. This may also alleviate the lack of nuance in the models.

With customization that includes a significant portion of sarcastic comments, it may also improve the sentiment analysis model's ability to detect sarcasm.

Improving the dataset

In model building, the greatest accuracy improvement was seen when the amount of parsed data was increased. With 28k unique features and 602k records, large computational times were expected. At first a reduced dataset of just 60k records was used, but the accuracies were so poor that efforts were increased towards using as much data as possible. For some models this meant leaving the model running for up to a week, which limited the scope of the project, but the accuracies of the models improved by around 10-20%, so this became the base data approach.

The dataset was heavily imbalanced and even when sampling a small amount of reviews bias affected the contributor perceptions of the assigned rating. As an example, the review assigned ID 5 earlier in this report was assigned rating 3, but the text is overwhelmingly negative. It was agreed that the text should likely have been assigned a rating of 1, but at no point were the ratings tampered with.

Quote 12. Representing Bias

"It doesn't stay cold enough to safely store food. I've tried every trick, actually considering returning it."

Surprisingly, the Random Forest Classifier accurately predicted this review which indicates that the personal bias of the contributors may have overlooked something that the model was able to detect, such as certain words or phrases that only feature in middling reviews.

All these items suggest that increasing the size and diversity of the dataset would increase the model's ability to perform an accurate sentiment analysis. In a truly ideal environment, with no time restriction, big data methods such as Spark could be implemented to allow analysis of the entire amazon review data, including all 233.1 million reviews across all categories.

With these resources, sampling could be performed without risk of losing important features. Meaning that the dataset can be balanced with confidence. Additionally, the English Wiktionary[i] contains 739,313 total headwords, and the larger dataset would increase the total number of unique lemmas to match better match that number, which would in turn mean that the model would be more robust against unexpected words. At 28k unique lemmas in the current model there are potentially 700k unique lemmas that the model would not consider or expect in sentiment analysis.

Modelling Improvements

Implementing more complex models – SHAP Analysis

Another analysis tool that has not yet been applied to the dataset is the SHAP (SHapley Additive exPlanations). SHAP analysis is a powerful machine learning technique that explains the influence of individual features on the predictions of a model. It can provide deeper insights into the relationships between words and their impact on the results. The following figures show examples of SHAP analysis based on a section representing 5% of the dataset. A maximum of 100 features was applied using the Count Vectorizer function. The computation time of this limited partition took 1832 minutes (about 1 and a half days).

Fig 13. Overall SHAP analysis for all 5 classes

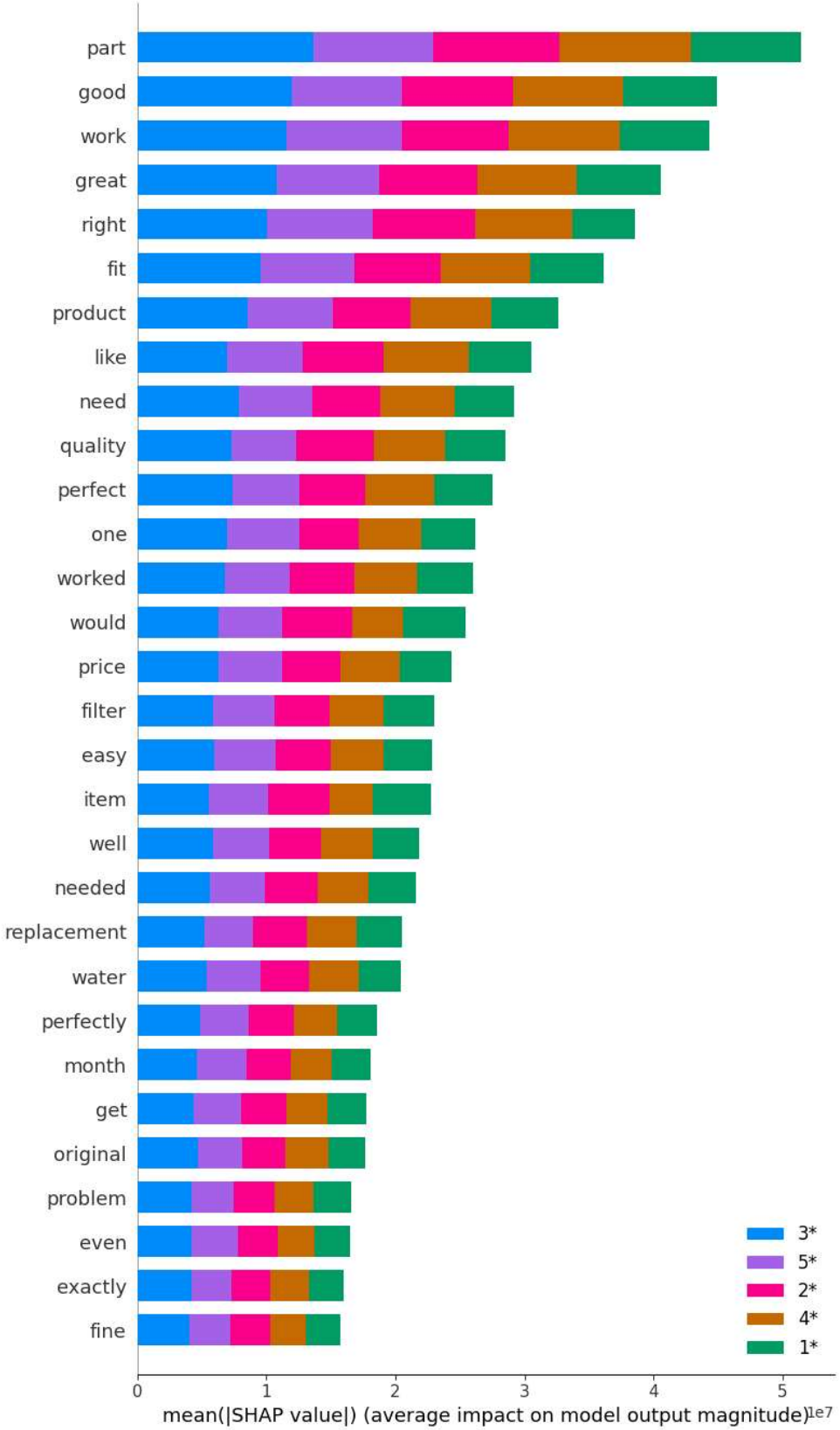


Fig 14. SHAP Analysis for Rating Class 1

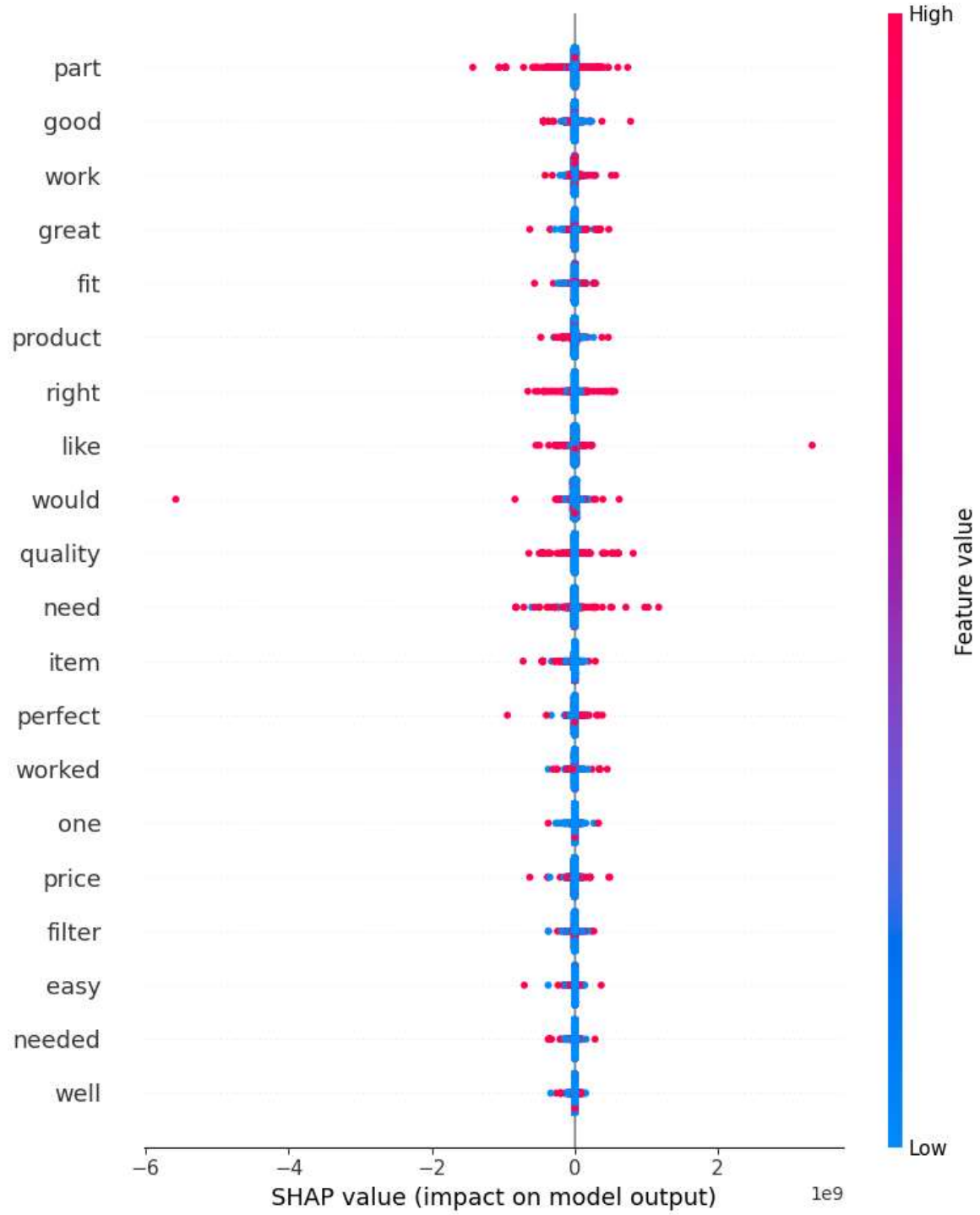


Fig 15. SHAP Analysis for Rating Class 2

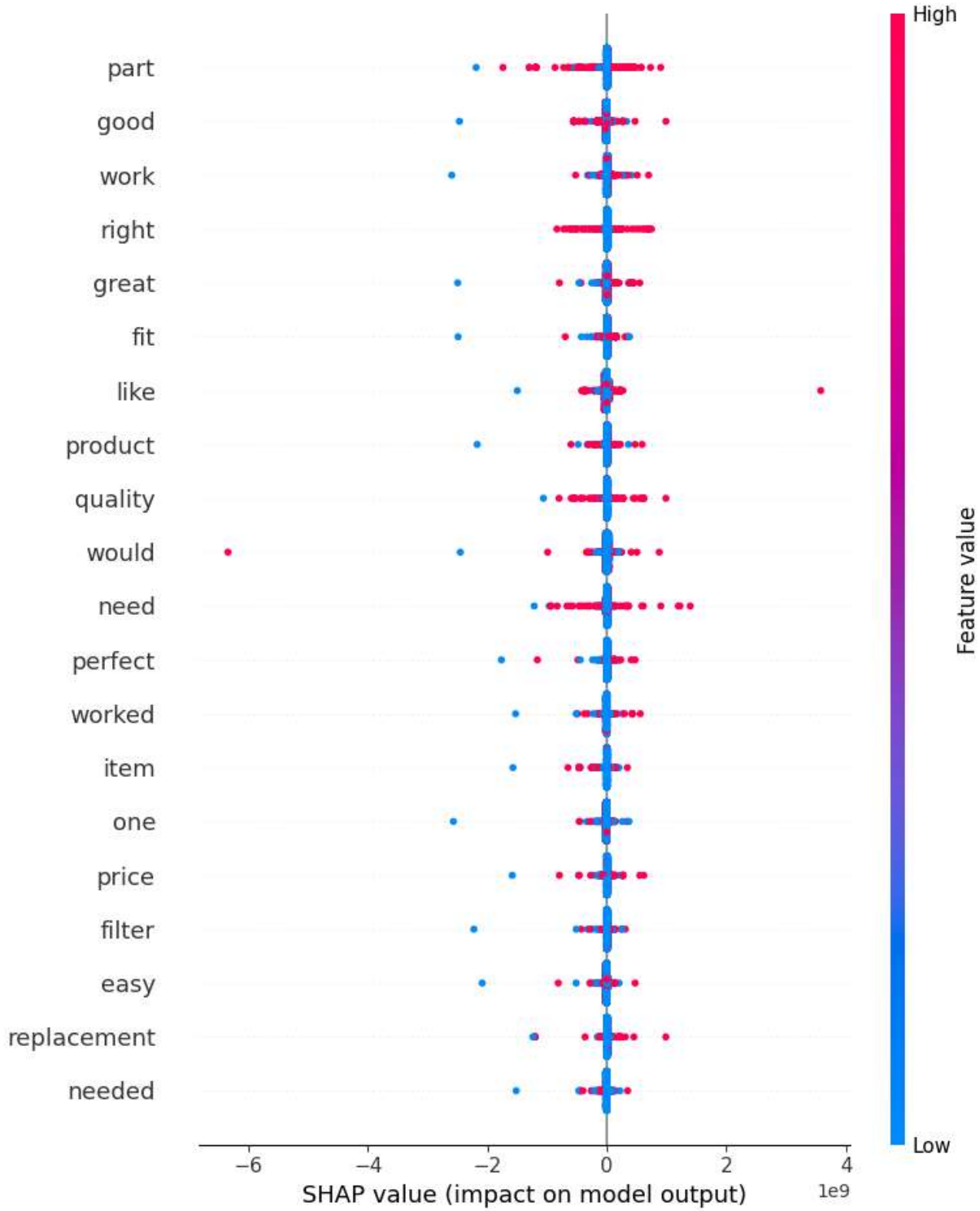


Fig 16. SHAP Analysis for Rating Class 3

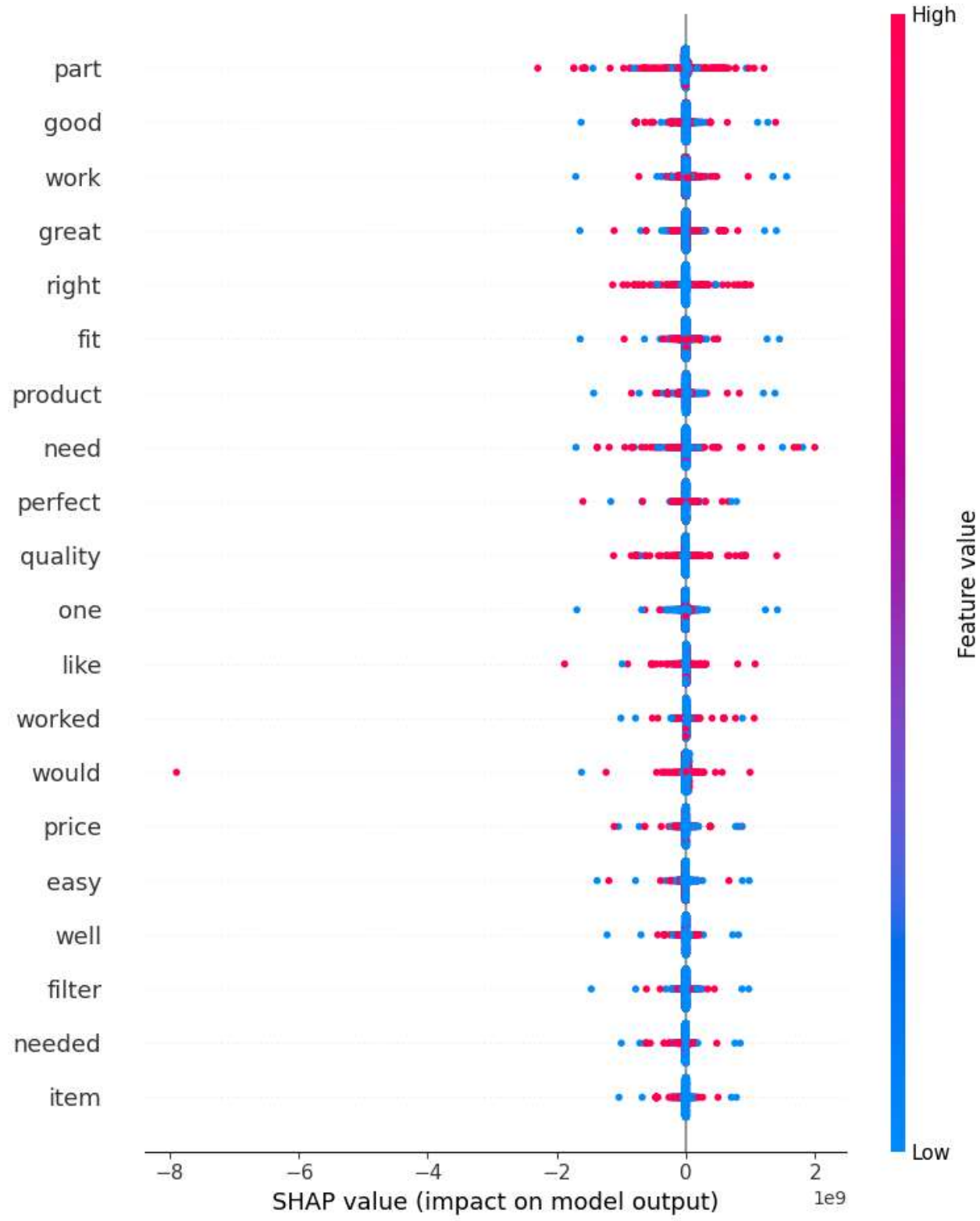


Fig 17. SHAP Analysis for Rating Class 4

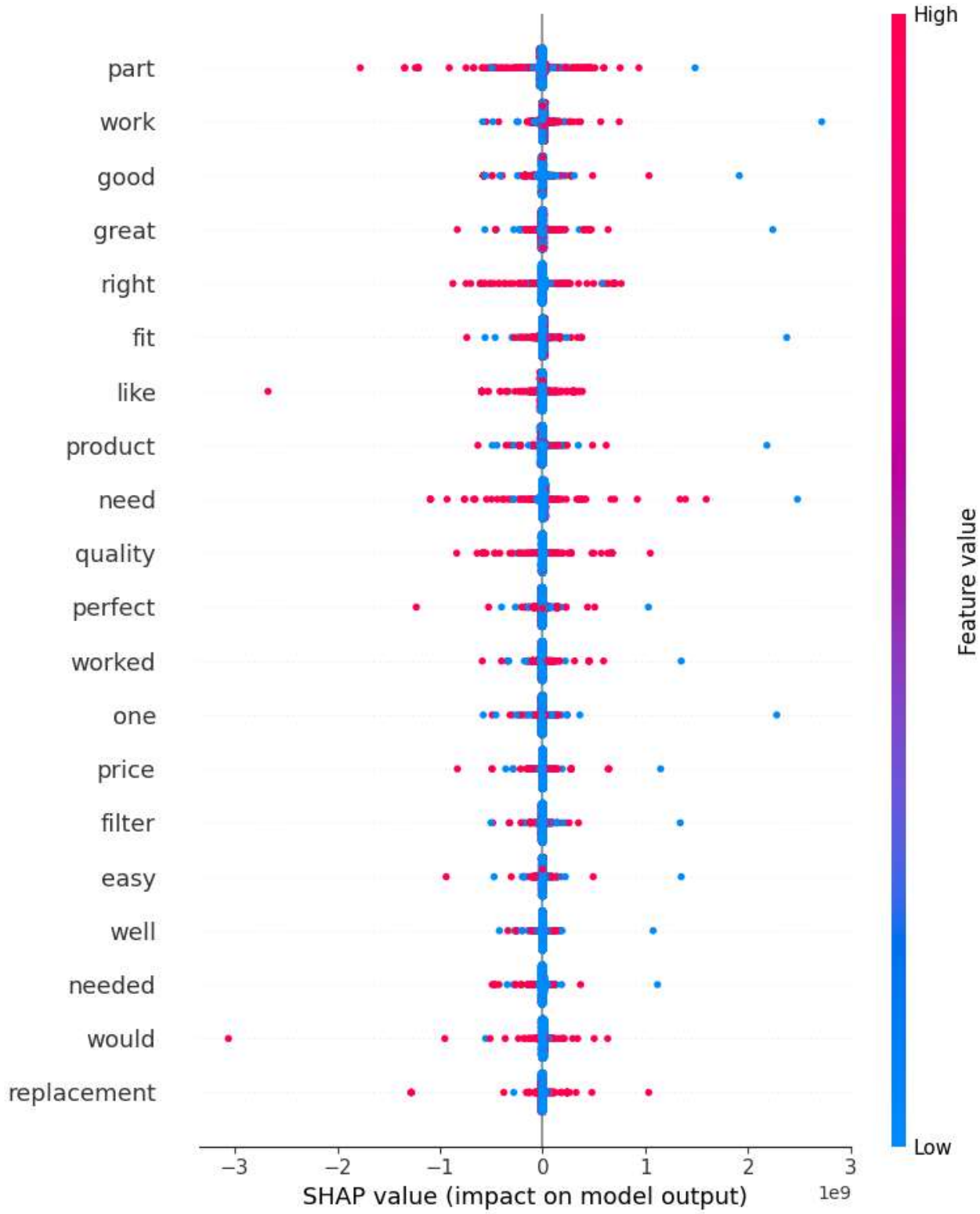
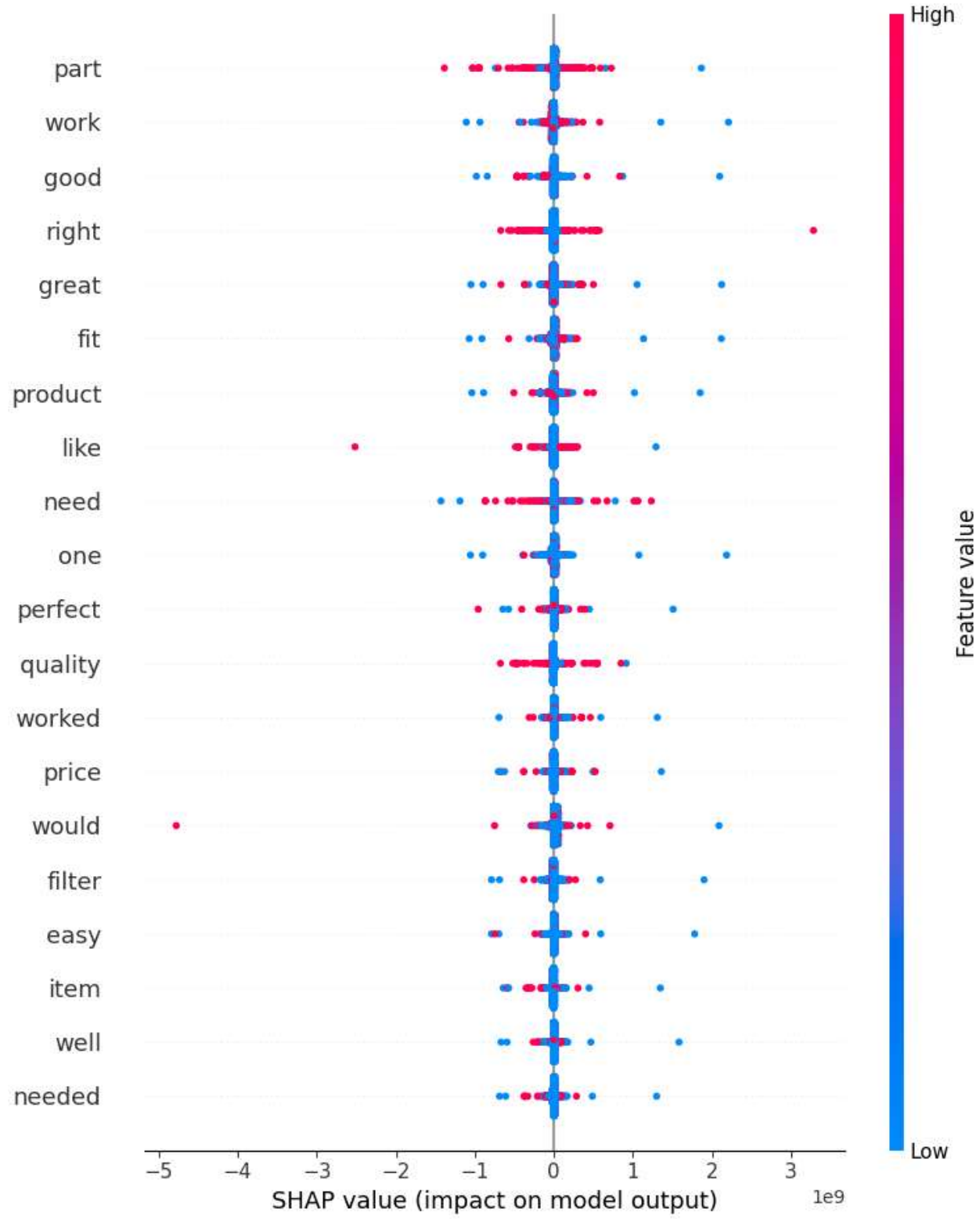


Fig 18. SHAP Analysis for Rating Class 5



Deep Learning Hypothesis

A key advantage of the regression modelling results is the analytical potential of the uncut results. These results allow observations to be drawn from the distribution of each class. In comparison, a key advantage of the classification models is the categorical output with accompanying model confidence.

Fig 19. Boxplot of Regression model prediction distributions per True class

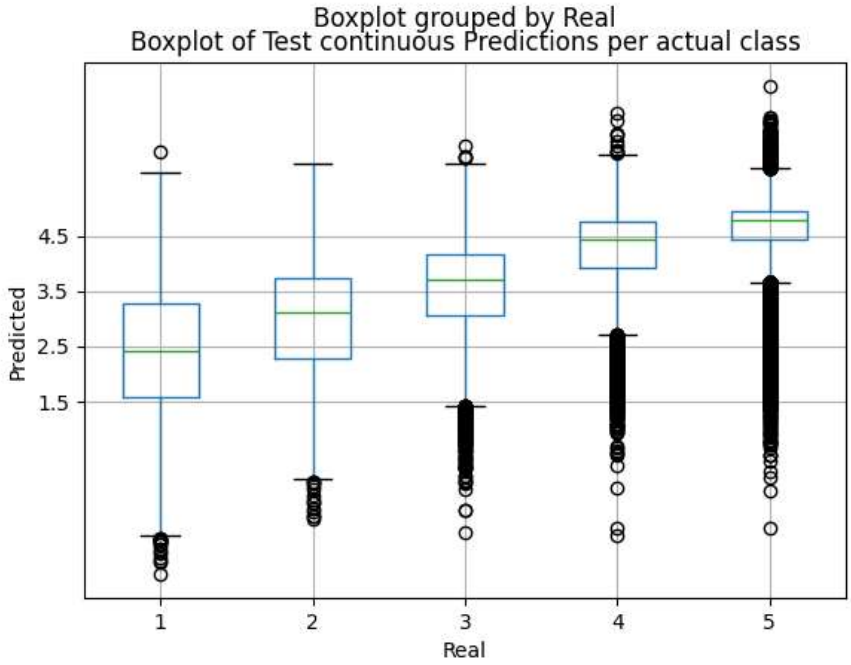
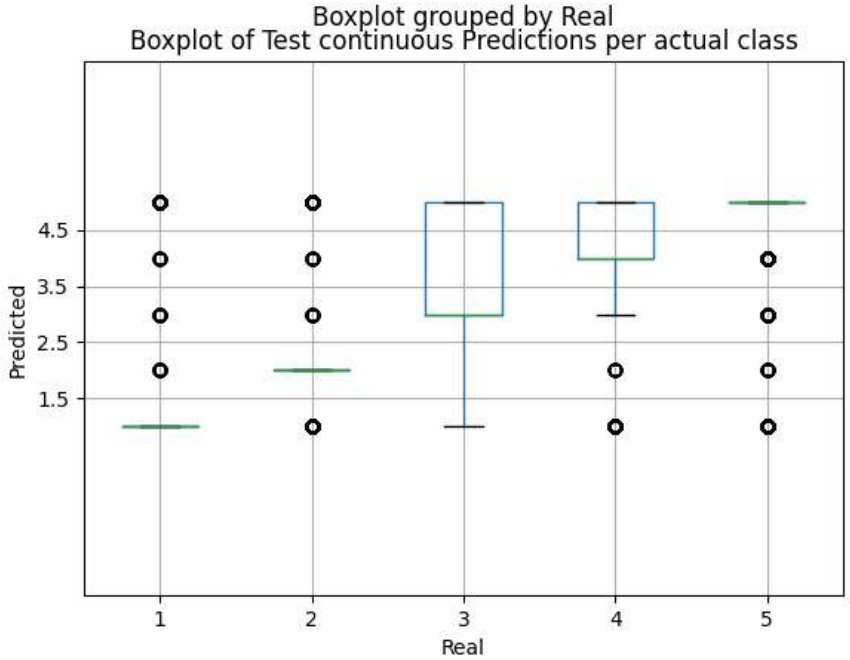


Fig 20. Boxplot of Classification model prediction distributions per True class



A potential ideal result could be provided by amalgamating a deep learning model which takes advantage of these benefits. In such a model, several stages of processing would be performed on the text data before it is classified. This would allow a combination of methods to be applied to the dataset which makes the most of the strengths of each stage.

Example Deep Learning model: (This model does not consider a specific neural network framework, but instead an amalgamation of seen processes.)

- Initial Parse: Custom Word2Vec model investigating word relations.
    - Input: Text data as series.
    - Output: vector matrix with stems as headers.
- Initial reduction: Recursive feature extraction using the strongest regression model.
    - estimator: Histogram Gradient Boosting Regressor.
    - scoring: negative mean squared error
    - target features: 0.95
    - reduction targets the vocabulary rather than the dataset.
    - Output: masked vector matrix from strongest 95% vocabulary items.
- Secondary parsing: text data parsed using the previous stem and vector techniques.
    - output: merged matrix on headers grouped by stemmed headers
- Secondary Reduction: Same as initial reduction.
    - Final Classification: Utilizing the strongest classifier.

The Word2vec pass reduction pair can be looped together. The Secondary reduction can also be performed multiple times until a satisfactory number of features is achieved. The key advantage of using a regression model to reduce the important features is the precise mean squared error calculation. Ending the process with a classification model still provides the user with a clear category and confidence interval of the prediction.

---

[i] Wikipedia.org - https://en.wikipedia.org/wiki/List_of_dictionaries_by_number_of_words