IS567 - Project Progress Report

A) Project Summary:

The aim of this project is to classify the amazon product reviews into appropriate product categories and further perform sentiment analysis on the reviews. Our goal is to perform multilingual product classification for two languages - English and German and find the most appropriate model for the product classification.

B) Descriptive Statistics:

We performed the basic checks like presence of null values and duplicate rows in the dataset. Our dataset does not contain null values or duplicate rows.

Before preprocessing descriptive stats

```
min(english_train_reviews['review_body'].str.len())
```
15

```
max(english_train_reviews['review_body'].str.len())
```
3908

```
english_train_reviews['review_body'].str.len().mean()
```
178.84632

```
min(german_train_reviews['review_body'].str.len())
```
16

```
max(german_train_reviews['review_body'].str.len())
```
3869

```
german_train_reviews['review_body'].str.len().mean()
```
207.918865

Fig 1. Basic descriptive statistics of review_body column before preprocessing activities for English and German languages

Before performing preprocessing, these are the statistics as seen in Fig 1. Fig 2 represents the word cloud for the column containing the product reviews and we see that the most repeated words include ‘one’, ‘use’, ‘product’, ‘work’, ‘bought’ and so on which are related to the words expected when one gives reviews about the products for English language. In the German language, we find words like ‘ich’ (I), ‘habe’ (have), ‘fur’ (for), ‘den’ (the), ‘gut’ (good).

Word Cloud for English and German language product reviews:

Fig 2. Word Cloud for the English and German product reviews before preprocessing

Counts of instances in different target categories for English language reviews

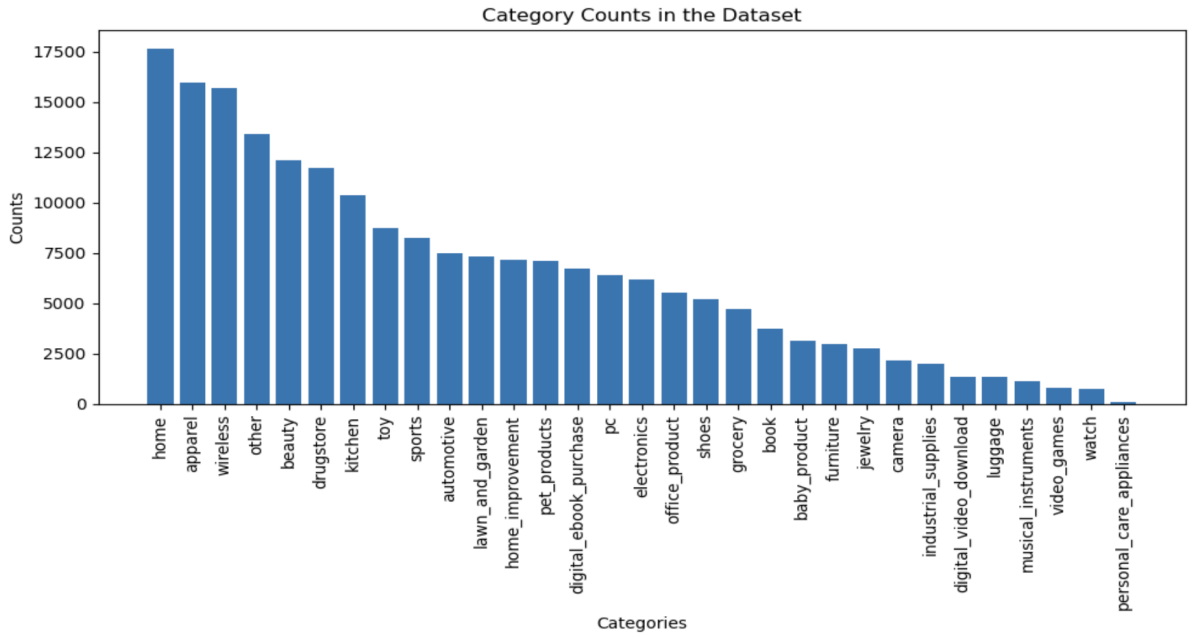

Fig 3a. Counts of instances in different target product categories for English language

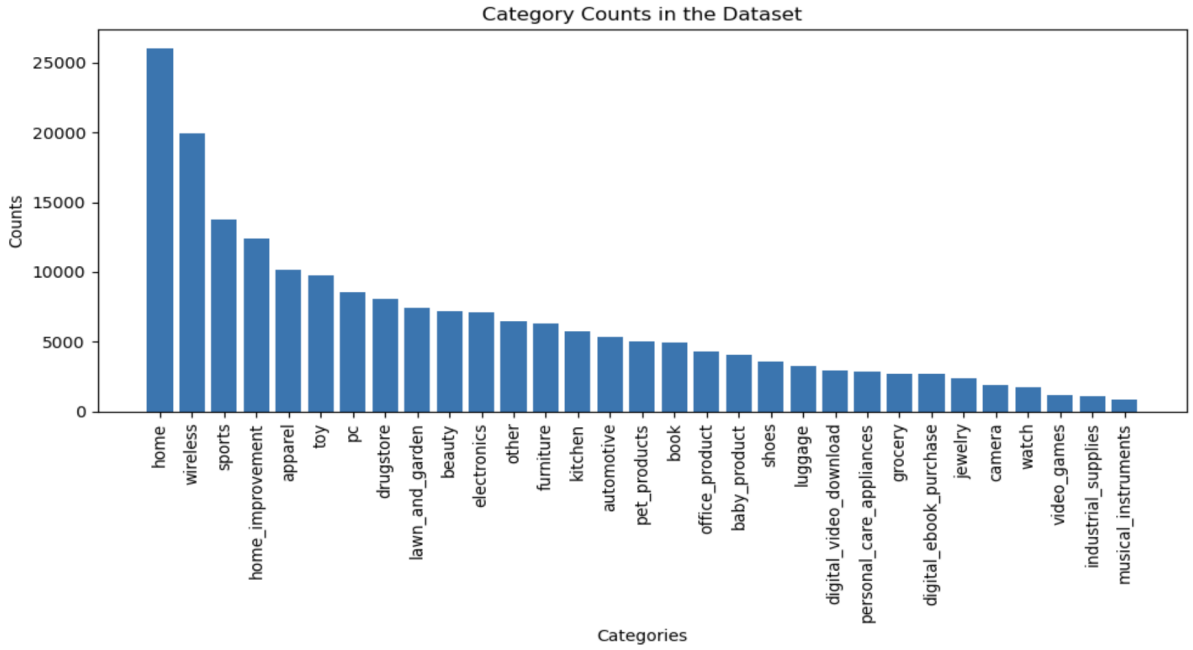

Fig 3b. Counts of instances in different target product categories for German language

We observe that our dataset is highly imbalanced across the categories, with 17500+ instances in ‘home’ and roughly 75 instances in ‘personal_care_appliances’ as seen in Fig 3a for the English language. The same plot when plotted for the German language (Fig 3b) shows similar results, with ‘home’ having 26063 instances and ‘musical_instruments’ having 844 instances.

C) Pre-processing Steps:

1. Out of the 6 languages, we filtered out two languages of English and German and created separate dataframes of 200,000 rows each.
2. We defined a function to perform the basic text preprocessing activities on the review_body column by removing whitespace, HTML tags, and performing tokenization.
3. We are performing classification for 2 different languages. So, we preprocessed the datasets separately based on the stopwords for English and German languages respectively.
4. Initially, we used lemmatization, but that increased the ambiguity and made the classification of the reviews difficult. In an example where leggings and legs may be used, where the former represents a cloth and the latter may represent the legs of a chair, which might be a furniture category, this would cause ambiguity. So we chose not to use stemmers.
5. We observed that the list of tokens also included some standalone numbers like '00', '000', '2600', and many more. Since the task at hand is to perform product category classification, we thought that numbers might not be of much significance to us and hence decided to exclude numbers as features for modeling. The preprocessing of numbers just excludes numbers in their numerical format and not the word format. For example, '2600' will be excluded but not 'two'.

After preprocessing:

```
min(english_train_reviews['preprocessed_text'].str.len())
```
```
0
```
```
max(english_train_reviews['preprocessed_text'].str.len())
```
```
2270
```
```
english_train_reviews['preprocessed_text'].str.len().mean()
```
```
106.961
```
```
min(german_train_reviews['preprocessed_text'].str.len())
```
```
0
```
```
max(german_train_reviews['preprocessed_text'].str.len())
```
```
2915
```
```
german_train_reviews['preprocessed_text'].str.len().mean()
```
```
129.772325
```

Fig 4. Basic descriptive statistics of 'preprocessed_text' column after preprocessing activities for English and German languages

3 rows have length 0 after preprocessing is performed and hence we will discard those instances.

Example of preprocessed text:

```
english_train_reviews.loc[5, 'preprocessed_text']
```
```
'poor quality material fuzzy day one got discolored less month even though kept outdoors covered porch returned ite
m'
```

Fig 5. Word Cloud for the product reviews after preprocessing

After performing preprocessing, these are the statistics, as seen in Fig 4. Fig 5 represents the word cloud for the column containing the preprocessed product reviews for English and German languages. We see that the most repeated words include 'one', 'use', 'even', 'make', 'work', 'used', and 'love'. The words like 'work', 'used' indicate the quality and state of the product purchased, and 'love' indicates that customers really liked the product. In German reviews, the most common words are related to negative sentiment towards the product such as 'allerding', 'leider', 'jedoch' which closely mean however, indicating a negative word associated with products.

D) <u>Feature Extraction and Feature Selection:</u>

There are numerous approaches for feature extraction including CountVectorizer, TF-IDF, word embeddings, bag of words, Principal Component Analysis (PCA) and so on. We tried two approaches: CountVectorizer and TF-IDF vectorizer for feature extraction.

<u>Baseline Model Results using CountVectorizer and TF-IDF vectorizer:</u>

Model	Extraction Technique	Number of categories	Precision	Recall	F1 Score
Multinomial Naive Bayes	TF-IDF	31	0.486	0.461	0.441
	Count Vectorizer	31	**0.489**	**0.486**	**0.475**

Table 1. Comparison of Count Vectorizer and TF-IDF techniques

Preliminary results from Table 1 show us that count vectorizer performs better than TF-IDF vectorizer, which was the basis of our decision to utilize count vectorizer for further feature selection and model run.

Model Results after k-best Feature Selection:

Type	Model	Precision	Recall	F1 Score
Chi-Square (k = 2500)	Linear SVC	0.488	0.419	0.425
	Logistic Regression	0.482	0.408	0.424
	Multinomial Naive Bayes	0.489	0.428	0.429
Chi-Square (k = 3500)	Linear SVC	0.484	0.421	0.426
	Logistic Regression	0.478	0.408	0.423
	Multinomial Naive Bayes	0.489	0.438	0.439
Chi-Square (k = 5000)	Linear SVC	0.483	0.424	0.429
	Logistic Regression	0.477	0.411	0.426
	Multinomial Naive Bayes	**0.493**	**0.446**	**0.446**
Mutual Information (k = 5000)	Linear SVC	0.462	0.411	0.415
	Logistic Regression	0.469	0.402	0.416
	Multinomial Naive Bayes	0.498	0.444	0.445

Table 2. Comparison of classification metrics using feature selection across linear models

E) Preliminary Models

Baseline model results

To get an idea of the performance of the classification task, we decided to choose linear classification models like Logistic Regression, Naive Bayes and Linear SVC as our baseline models to compare our results. We have used the 'weighted' metrics as this approach takes into account the balance of classes. We decided to include all the 31 categories and observed the results as highlighted in Table 3.

Model	Number of categories	Precision	Recall	F1 Score
Linear SVC	31	0.433	0.445	0.434
Logistic Regression	31	0.458	0.46	0.45
Multinomial Naive Bayes	31	**0.489**	**0.486**	**0.475**

Table 3. Comparison of classification metrics across the baseline linear models

Undersampling and Oversampling to handle imbalanced data:

Since the dataset is highly imbalanced, with a ratio of 235:1 of the majority class to the minority class, we decided to balance it by performing a combination of upsampling and downsampling. We decided to obtain 5000 training instances for each class to achieve improvement in performance and reduce computation time complexity through random undersampling and SMOTE oversampling techniques.
We ran the same linear baseline models and the classification metrics are summarized in Table 4.

Undersampling and Oversampling

```
# Define the over-sampling and under-sampling methods
over = SMOTE(sampling_strategy={2:5000, 4:5000, 5:5000, 7:5000, 10:5000, 11:5000, 14:5000, 15:5000, 18:5000,
                                19:5000, 23:5000, 28:5000, 29:5000})
under = RandomUnderSampler(sampling_strategy={0:5000, 1:5000, 3:5000, 6:5000, 8:5000, 9:5000, 12:5000,
                                              13:5000, 16:5000, 17:5000, 20:5000, 21:5000, 22:5000,
                                              24:5000, 25:5000, 26:5000, 27:5000, 30:5000})

# Create the pipeline
steps = [('o', over), ('u', under)]
pipeline = Pipeline(steps=steps)

# Transform the dataset
X_resampled, y_resampled = pipeline.fit_resample(X_train, y_train_encoded)

# Inverse transform the labels to get the original class labels
y_resampled_original = label_encoder.inverse_transform(y_resampled)
```

```
# summarize the new class distribution
# Oversample with SMOTE and random undersample for imbalanced dataset
from collections import Counter
counter = Counter(y_resampled)
print(counter)
```

```
Counter({0: 5000, 1: 5000, 2: 5000, 3: 5000, 4: 5000, 5: 5000, 6: 5000, 7: 5000, 8: 5000, 9: 5000, 10: 5000, 11: 50
00, 12: 5000, 13: 5000, 14: 5000, 15: 5000, 16: 5000, 17: 5000, 18: 5000, 19: 5000, 20: 5000, 21: 5000, 22: 5000, 2
3: 5000, 24: 5000, 25: 5000, 26: 5000, 27: 5000, 28: 5000, 29: 5000, 30: 5000})
```

Model	Number of categories	Precision	Recall	F1 Score
Linear SVC	31	0.443	0.401	0.406
Logistic Regression	31	0.466	0.407	0.421
Multinomial Naive Bayes	31	**0.469**	**0.455**	**0.442**

Table 4. Comparison of classification metrics across the baseline linear models with undersampling and oversampling

Comparing Table 3 and Table 4, we see that there is no significant improvement in the performance of the model in identifying product categories, even though the dataset is balanced. In fact, the F1 score (which ties both precision and recall) has dropped for Linear SVC and Logistic Regression models. This gives us an indication that the model is not performing well even with balanced data.

Filtering dataset based on more than 10000 reviews for product categories

Since the dataset has 31 classes, the traditional models are not able to clearly delineate the boundaries between each class. So, to improve the performance of the models and better classify and distinguish between the different categories, we decided to concentrate on those product categories that have more than 10,000 reviews. After doing this, we obtained 7 different classes, namely: home, apparel, wireless, other, beauty, drugstore, and kitchen for the English language. Similar filtering on the German language resulted in 5 different classes namely: 'apparel', 'home', 'home_improvement', 'sports' and 'wireless'. The traditional models are trained on this filtered dataset, and the results are summarized in Table 5. The filtered dataset has very less imbalance as the ratio of the majority to the minority class is 1.7:1. So, there is no need to perform any undersampling or oversampling.

Based on the results in Table 5, we see that with a reduction in the number of classes from 31 to 7 based on the criteria of 10,000 reviews, we were able to achieve a drastic improvement in the performance in overall accuracy from an average of 40% to 60%. This tells us that the traditional methods are now able to distinguish the classes better with this reduction criteria, even though this is a completely different classification task now.

Language	**Model**	**Number of categories**	**Precision**	**Recall**	**F1 Score**
English	Linear SVC	7	0.583	0.589	0.584
	Logistic Regression	7	0.594	0.599	0.595
	Multinomial Naive Bayes	7	**0.627**	**0.631**	**0.623**
	Multinomial Naive Bayes *	6	**0.67**	**0.674**	**0.668**
	Multinomial Naive Bayes with k-best feature selection (k = 5000) *	6	0.672	0.676	0.672
German	Linear SVC	5	0.624	0.63	0.624

	Logistic Regression	5	0.653	0.655	0.649
	Multinomial Naive Bayes	5	0.686	0.679	0.668
	Multinomial Naive Bayes with k-best feature selection (k = 5000)	5	**0.69**	**0.683**	**0.673**

Table 5. Comparison of classification metrics across the linear models on categories with more than 10000 reviews for English and German languages

* The category ‘other’ was removed, as it was a broad category containing a variety of reviews and due to the similarity of these reviews with the reviews under definite product categories, many reviews were getting misclassified as other, which was affecting the performance significantly. From the table it can be observed that, by removing that category alone, there was an improvement in the performance by 5% on an average. We have explained more about this in the error analysis section.

F) <u>Error Analysis:</u>

We performed error analysis on both top 7 categories and 6 categories after removing the ‘other’ category. The following are a few interesting examples.

Observing 5 examples, which belong to ‘other’ category, but were misclassified:

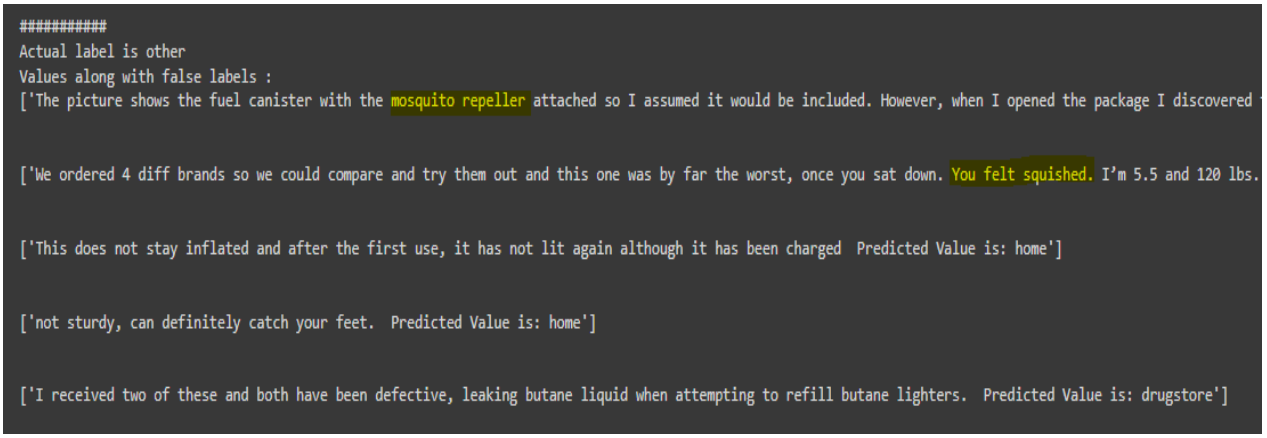

Fig 6. Misclassified labels for ‘other’ category

The first review is related to a ‘mosquito repellant’ which is wrongly classified as ‘home’ but should have been ‘other’. On analysis, one can understand that a better suited category for a mosquito repellent would in fact be ‘home’ rather than ‘other’, which contains reviews with varying context. The second example - *['We ordered 4 diff brands so we could compare and*

try them out and this one was by far the worst, once you sat down. You felt squished. I'm 5.5 and 120 lbs. My husband could barely fit. Predicted Value is: apparel'] is related to furniture. This review contains words such as squished, barely fit, etc. This instance is wrongly classified as 'apparel' and the actual label is 'other', whereas it should have been either 'home' or 'furniture' .

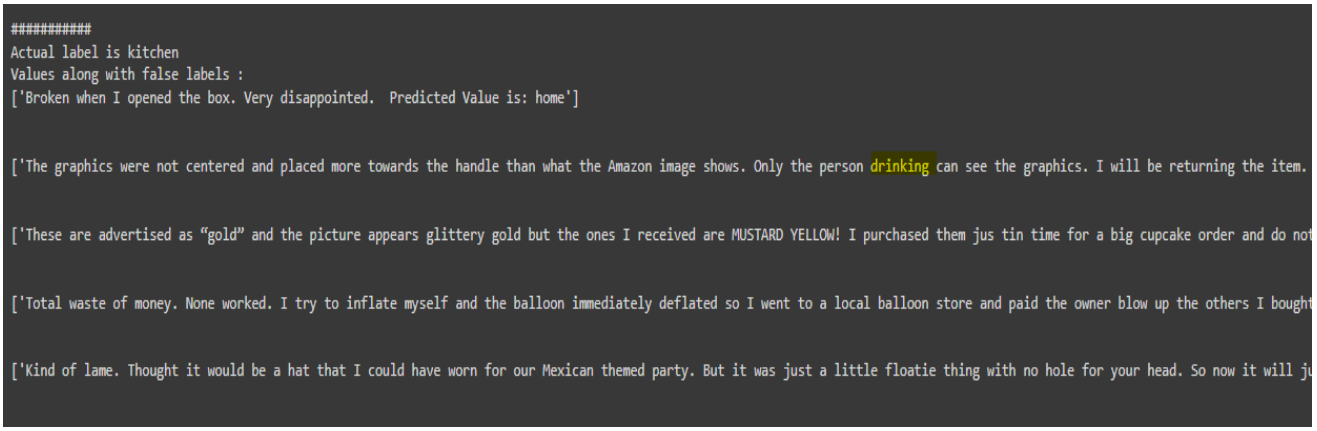

Fig 7. Misclassified labels for 'kitchen' category

When we analyzed the label 'kitchen' as shown above, we found out that one of the reviews was misclassified as 'other' as there was little contextual information in the review, but after human analysis it can be understood that the review should be categorized as 'kitchen'.

After removing the 'other' category, affirming our assumption that the category 'other' creates ambiguity, the remaining reviews can be categorized more accurately. The same review got correctly categorized into the label 'kitchen' as follows -

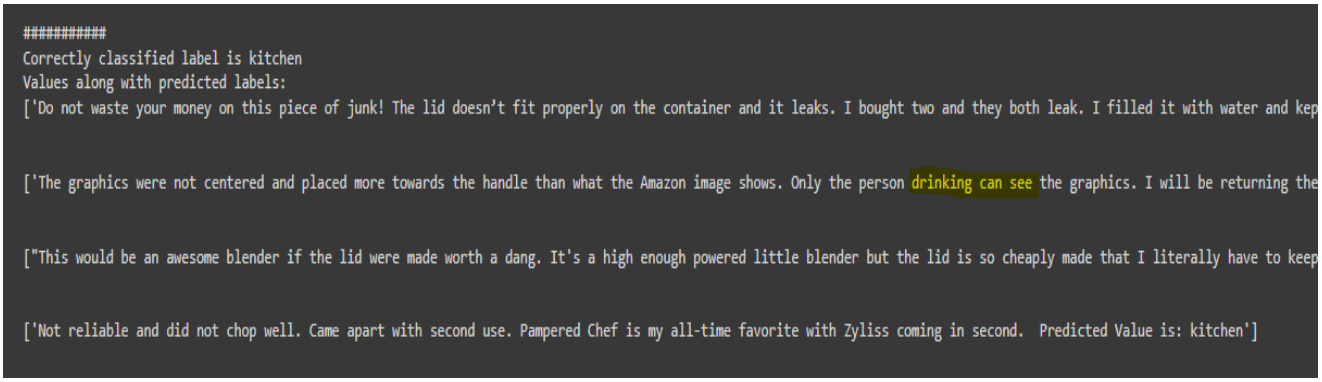

Fig 8. Correctly classified labels for 'kitchen' category after filtering out 'other'

Besides this, there is a general trend of misclassification across labels, where the review is related to failed delivery, or return of the product etc. From the features, it is difficult to understand which category should such reviews belong to as there is no description of the product being delivered or returned. Such types of reviews are always categorized as 'home' by the model, and even with better models or even a neural network, it might be difficult to categorize these reviews accurately as seen in the below examples:

These examples include:

1. Actual label is ‘drugstore’
   Values along with false labels :
   ['Amazon never deliver the items. Horrible customer service. Considering new purchase choices. Predicted Value is: home']

2. Actual label is ‘apparel’
   Values along with false labels :
   - ['The hot glued parts are falling off. Predicted Value is: home']
   - ['ordered product in MARCH has not arrived as of MAY 11. Attempt to contact seller and NO response! Predicted Value is: home']

3. Actual label is ‘wireless’
   Values along with false labels :
   ['Nothing like the photo. Boo. Predicted Value is: home']

However, there are other reviews where there are enough features to conclusively categorize them into the right labels, but the linear models fail to do so. On the other hand there are reviews which seem to be categorized correctly based on the review body, but have a different class label such as:

1. ["When I opened the bag the product smelled of chemical. Which isn't very surprising. Thought I just may have to wash it. I held it up and the fabric was awful. Itchy feeling and very heavy. I tried it on anyway once again I thought after wash this may change. Well it felt heavy on and it has no shape. This is not breathable fabric at all. So I did not wash it repackaged it and sent it back. I bought it to hang around the water park or beach with the kids as a cover up. Nope Predicted Value is: home"]

2. ['Smell terrible and neck is coming apart sent back Predicted Value is: home']

3. ['I wish I had taken a picture of the set before I sent it back. But just know that the fork tine split off and the knife had splinters on it that snagged the gauzy bag for storage. The reason I purchased this was to see how the bamboo ones differed. One good thing is that the weight of bamboo is quite a bit less than metal utensils. Couldn\'t comment on "Easy To Clean" since I didn\'t actually use it - afraid of splinters! Predicted Value is: kitchen'] - Although this review should belong to ‘kitchen’, its actual label is ‘home’.

There might be an overlap between the features, and due to the lack of proper distinction between the product categories, many reviews are misclassified. This overlap may reduce after employing a neural network model with cross-validation. This would be our primary focus going forward: improving the existing models and trying to incorporate contextual information via word embeddings. We will also be performing sentiment analysis tasks focusing on the review ratings.