Sentiment Analysis and Product Recommendation on Amazon's Electronics Dataset Reviews

Capstone Project 2

INTRODUCTION:

The Internet has revolutionized the way we buy products. In the retail e-commerce world of online marketplace, where experiencing products are not feasible. Also, in today's retail marketing world, there are so many new products are emerging every day. Therefore, customers need to rely largely on product reviews to make up their minds for better decision making on purchase. However, searching and comparing text reviews can be frustrating for users. Hence we need better numerical ratings system based on the reviews which will make customers purchase decision with ease.

Goal:

During their decision making process, consumers want to find useful reviews as quickly as possible by rating system. Therefore, models able to predict the user rating from the text review are critically important. Getting an overall sense of a textual review could in turn improve consumer experience. Also, it can help businesses to increase sales, and improve the product by understanding customer's needs.

In this project, the amazon review dataset for electronics products will be considered. The reviews and ratings given by the user to different products as well as reviews about user's experience with the product(s) will be considered.

The main goal for this project is to develop a model to predict user rating, usefulness of review and recommend most similar items to users based on collaborative filtering.

Client:

Amazon is our client. The company wants to develop a software tool that will identify the positive and negative words which customers use when they write reviews for the home and kitchen products as their purchase inclination.

Getting an overall sense of a textual review could in turn improve better consumer decision making experience. Also, it can help businesses to increase sales, and improve the product by understanding customers' needs and problems.

DATA COLLECTION:

For this project, the electronics dataset consist of reviews and product information from amazon were collected. This dataset includes reviews (ratings, text, helpfulness votes) and product metadata (descriptions, category information, price, brand, and image features).

1. Product Complete Reviews data:

This dataset includes electronics product reviews such as ratings, text, helpfulness votes. This dataset was obtained from http://jmcauley.ucsd.edu/data/amazon/. The original data was in json format. The json was imported and decoded to convert json format to csv format. The sample dataset is shown below:

	asin	helpful	Rating	reviewText	reviewTime	reviewerID	reviewerName	summary	unixReviewTime
0	0528881469	[0, 0]	5	We got this GPS for my husband who is an (OTR)...	06 2, 2013	AO94DHGC771SJ	amazdnu	Gotta have GPS!	1370131200
1	0528881469	[12, 15]	1	I'm a professional OTR truck driver, and I bou...	11 25, 2010	AMO214LNFCEI4	Amazon Customer	Very Disappointed	1290643200
2	0528881469	[43, 45]	3	Well, what can I say. I've had this unit in m...	09 9, 2010	A3N7T0DY83Y4IG	C. A. Freeman	1st impression	1283990400
3	0528881469	[9, 10]	2	Not going to write a long review, even thought...	11 24, 2010	A1H8PY3QHMQQA0	Dave M. Shaw "mack dave"	Great grafics, POOR GPS	1290556800
4	0528881469	[0, 0]	1	I've had mine for a year and here's what we go...	09 29, 2011	A24EV6RXELQZ63	Wayne Smith	Major issues, only excuses for support	1317254400

Fig 1: Sample Product Reviews Dataset

Each row corresponds to a customer review and includes the following variables:

reviewerID : ID of the reviewer, e.g. A2SUAM1J3GNN3B - type: object
asin : ID of the product , e.g. 0000013714 – type: object
reviewerName : name of the reviewer – type: object
helpful : helpfulness of the review, e.g. 2/3 – type: object
reviewText : text of the review – type: object
overall : Rating (1,2,3,4,5)– type: float64
summary : summary of the review – type: object
unixReviewTime : time of the review (unix time) – type: int64
reviewTime : time of the review (raw) – type: object

2. Product Metadata:

This dataset includes electronics product metadata such as descriptions, category information, price, brand, and image features. This dataset was obtained from http://jmcauley.ucsd.edu/data/amazon/. The json was imported and decoded to convert json format to csv format. The sample product meta dataset is shown below:

	asin	imUrl	description	categories	title	price	salesRank	related	brand
0	0132793040	http://ecx.images-amazon.com/images/I/31JIPhp%...	The Kelby Training DVD Mastering Blend Modes i...	[[Electronics, Computers & Accessories, Cables...	Kelby Training DVD: Mastering Blend Modes in A...	NaN	NaN	NaN	NaN
1	0321732944	http://ecx.images-amazon.com/images/I/31uogm6Y...	NaN	[[Electronics, Computers & Accessories, Cables...	Kelby Training DVD: Adobe Photoshop CS5 Crash ...	NaN	NaN	NaN	NaN
2	0439886341	http://ecx.images-amazon.com/images/I/51k0qa8f...	Digital Organizer and Messenger	[[Electronics, Computers & Accessories, PDAs, ...	Digital Organizer and Messenger	8.15	{'Electronics': 144944}	{'also_viewed': ['0545016266', 'B009ECM8QY', '...	NaN

Fig 2: Sample Product Meta Dataset

Each row corresponds to product and includes the following variables:

asin: ID of the product, e.g. 0000031852

title: name of the product

price: price in US dollars (at time of crawl)

imUrl: url of the product image

related: related products (also bought, also viewed, bought together, buy after viewing)

salesRank: sales rank information

brand: brand name

categories: list of categories the product belongs to

DATA WRANGLING:

Merging Dataframes:

Product reviews and meta datasets in json files were saved in different dataframes and two dataframes were merged together using left join and “asin” was kept as common merger. Final merged data frame description is shown below:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1689188 entries, 0 to 1689187
Data columns (total 17 columns):
asin              1689188 non-null object
helpful           1689188 non-null object
Rating            1689188 non-null int64
reviewText        1689188 non-null object
reviewTime        1689188 non-null object
reviewerID        1689188 non-null object
reviewerName      1664458 non-null object
summary           1689188 non-null object
unixReviewTime    1689188 non-null int64
imUrl             1687975 non-null object
description       1655511 non-null object
categories        1689188 non-null object
title             1643686 non-null object
price             1639882 non-null float64
salesRank         810070 non-null object
related           1662142 non-null object
brand             954251 non-null object
dtypes: float64(1), int64(2), object(14)
memory usage: 232.0+ MB
```

Fig 3: Merged dataframe info

In order to reduce time consumption for running models, only headphones products were chosen and the following method was adopted.

1. Out of 1689188 rows, 45502 rows were null values in product title. Those rows were dropped.
2. Dataset with product title named “Headphones”, “Headphones”,”headphones”,”headphone” were extracted from merged dataframe. Final headphones dataset was 64305 rows (observations).

Handling Duplicates, Missing Values:

1. 22699 rows in brand column were observed as null values. To solve this, brand name was extracted from title and replaced null values in brand.
2. Dropped missing values in “reviewerName”,”price”,”description”,”related” were dropped.
3. “reviewText” and “summary” were concatenated and was kept under review_text feature
4. Helpful feature was split into positive and negative feedback.

5. Ratings greater than or equal to 3 was categorized as “good” and less than 3 was classified as “bad”.
6. Helpfulness ratio was calculated based on pos feedback/total feedback for that review
7. Dropped duplicates based on “asin”, “reviewerName”,”unixReviewTime”. After dropping duplicates, the dataset consisted 61129 rows and 18 features as shown in Fig 4.
8. ReviewTime was converted to datetime '%m %d %Y format.
9. Columns were renamed for clarity purpose.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 61129 entries, 1260 to 1689187
Data columns (total 18 columns):
product_id          61129 non-null object
rating              61129 non-null int64
reviewer_id         61129 non-null object
reviewer_name       61129 non-null object
unix_review_time    61129 non-null int64
url                 61129 non-null object
description         61129 non-null object
categories          61129 non-null object
product_title       61129 non-null object
price               61129 non-null float64
related             61129 non-null object
brand_name          61129 non-null object
review_text         61129 non-null object
pos_feedback        61129 non-null int64
neg_feedback        61129 non-null int64
rating_class        61129 non-null object
help_prop           61129 non-null float64
review_time         61129 non-null datetime64[ns]
dtypes: datetime64[ns](1), float64(2), int64(4), object(11)
memory usage: 8.9+ MB
```

Fig 4: Description of headphones dataset

Descriptive Statistics:

The following summary statistics was obtained

Number of reviews: 61129

Number of unique reviewers: 42062

Number of unique products: 1878

Average rating score: 4.056

Average helpful ratio score: 0.351

Number of positive feedback: 25222

Number of negative feedback: 14202

Number of good ratings: 52807

Number of Bad ratings: 8322

Ratings	Number of Reviews
1	3870
2	4452
3	7102
4	14639
5	31066

The summary statistics for headphones dataset is shown below:

	rating	pos_feedback	neg_feedback	help_prop
count	61129.000000	61129.000000	61129.000000	61129.000000
mean	4.056438	3.082269	0.641709	0.350887
std	1.217506	31.516073	2.777990	0.442493
min	1.000000	0.000000	0.000000	0.000000
25%	3.000000	0.000000	0.000000	0.000000
50%	5.000000	0.000000	0.000000	0.000000
75%	5.000000	1.000000	0.000000	0.947368
max	5.000000	3800.000000	144.000000	1.000000

Fig 5. Summary Statistics

Preprocessing Text:

Since, text is the most unstructured form of all the available data, various types of noise are present in it and the data is not readily analyzable without any pre-processing. The entire process of cleaning and standardization of text, making it noise-free and ready for analysis is known as text preprocessing. In this section, the following text preprocessing were applied.

1. **Removing HTML tags**

HTML tags which typically does not add much value towards understanding and analyzing text. HTML words were removed from text.

2. **Removing accented characters**

Accented characters/letters were converted and standardized into ASCII characters.

3. **Expanding Contractions**

Contractions are shortened version of words or syllables. They exist in either written or spoken forms. Shortened versions of existing words are created by removing specific letters and sounds. In case of English contractions, they are often created by removing one of the vowels from the word.

By nature, contractions do pose a problem for NLP and text analytics because, to start with, we have a special apostrophe character in the word. Ideally, we can have a proper mapping for contractions and their corresponding expansions and then use it to expand all the contractions in our text.

4. **Removing Special Characters**

One important task in text normalization involves removing unnecessary and special characters. These may be special symbols or even punctuation that occurs in sentences. This step is often performed before or after tokenization. The main reason for doing so is because often punctuation or special characters do not have much significance when we analyze the text and utilize it for extracting features or information based on NLP and ML.

5. **Lemmatization**

The process of lemmatization is to remove word affixes to get to a base form of the word. The base form is also known as the root word, or the lemma, will always be present in the dictionary.

6. **Removing stopwords**

Stopwords are words that have little or no significance. They are usually removed from text during processing so as to retain words having maximum significance and context. Stopwords are usually words that end up occurring the most if you aggregated any corpus of text based on singular tokens and checked their frequencies. Words like a, the , me , and so on are stopwords.

7. **Building a Text Normalizer**

Based on the functions which we have written above and with additional text correction techniques (such as lowercase the text, and remove the extra newlines, white spaces, apostrophes), we built a text normalizer in order to help us to preprocess the new_text document.

After applying text normalizer to 'the review_text' document, we applied tokenizer to create tokens for the clean text. As a result of that, we had 3070479 words in total. Eventually, after completing all data wrangling and preprocessing phases, we save the dataframe to csv file as a 'Cleaned_Reviews_Home_and_Kitchen.csv. After cleaning, we have 25276 observations.

A clean dataset will allow a model to learn meaningful features and not overfit on irrelevant noise. After following these steps and checking for additional errors, we can start using the clean, labelled data to train models in modeling section.

EXPLORATORY DATA ANALYSIS (EDA):

After collecting data, wrangling data then exploratory analyses were carried out. The following insights were explored through exploratory analyses.

- Predicting ratings based on reviews
- Usefulness on large volume of reviews
- Rating vs number of reviews
- Rating vs proportion of reviews
- Helpful proportion vs Number of reviews
- Rating vs helpfulness ratio
- Top 20 most reviewed products
- Bottom 20 reviewed products
- Positive and negative words
- World cloud for different ratings, brand name etc

1. Top 20 Most Reviewed Brands

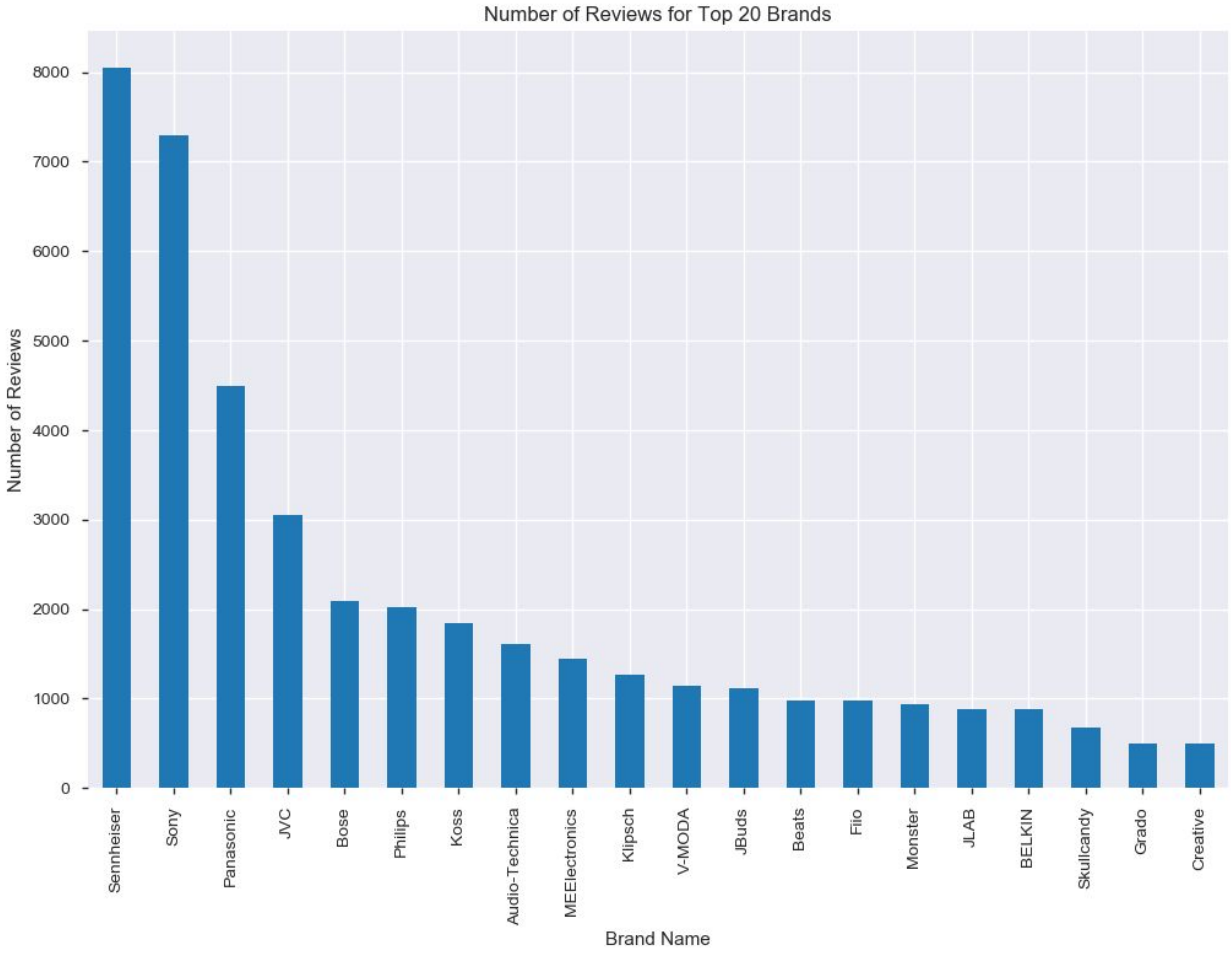


Fig 6. Top 20 most reviewed brands

2. Top 20 Least Reviewed Brands

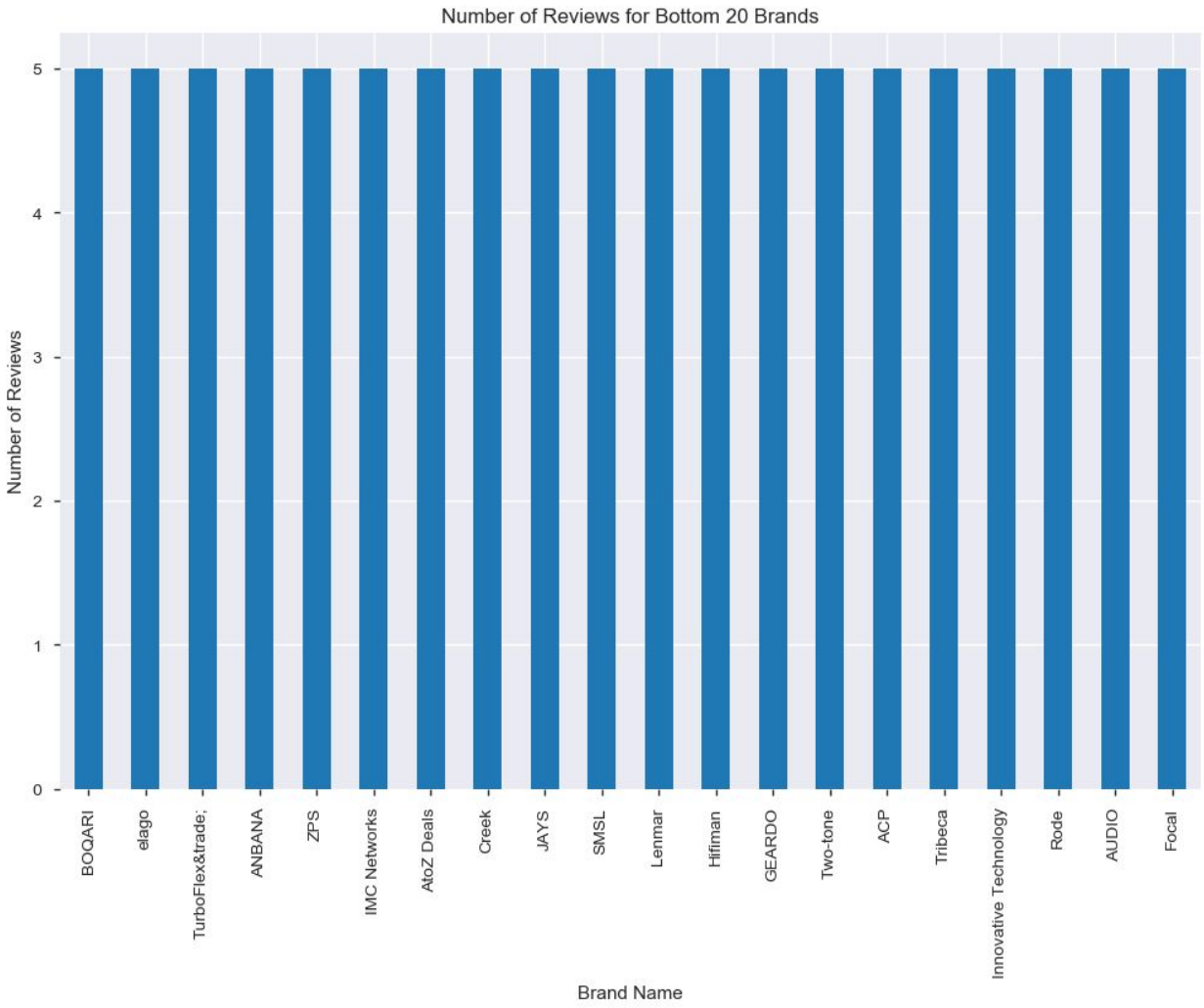


Fig 7. Top 20 least reviewed brands

3. Top 20 Most Reviewed Products

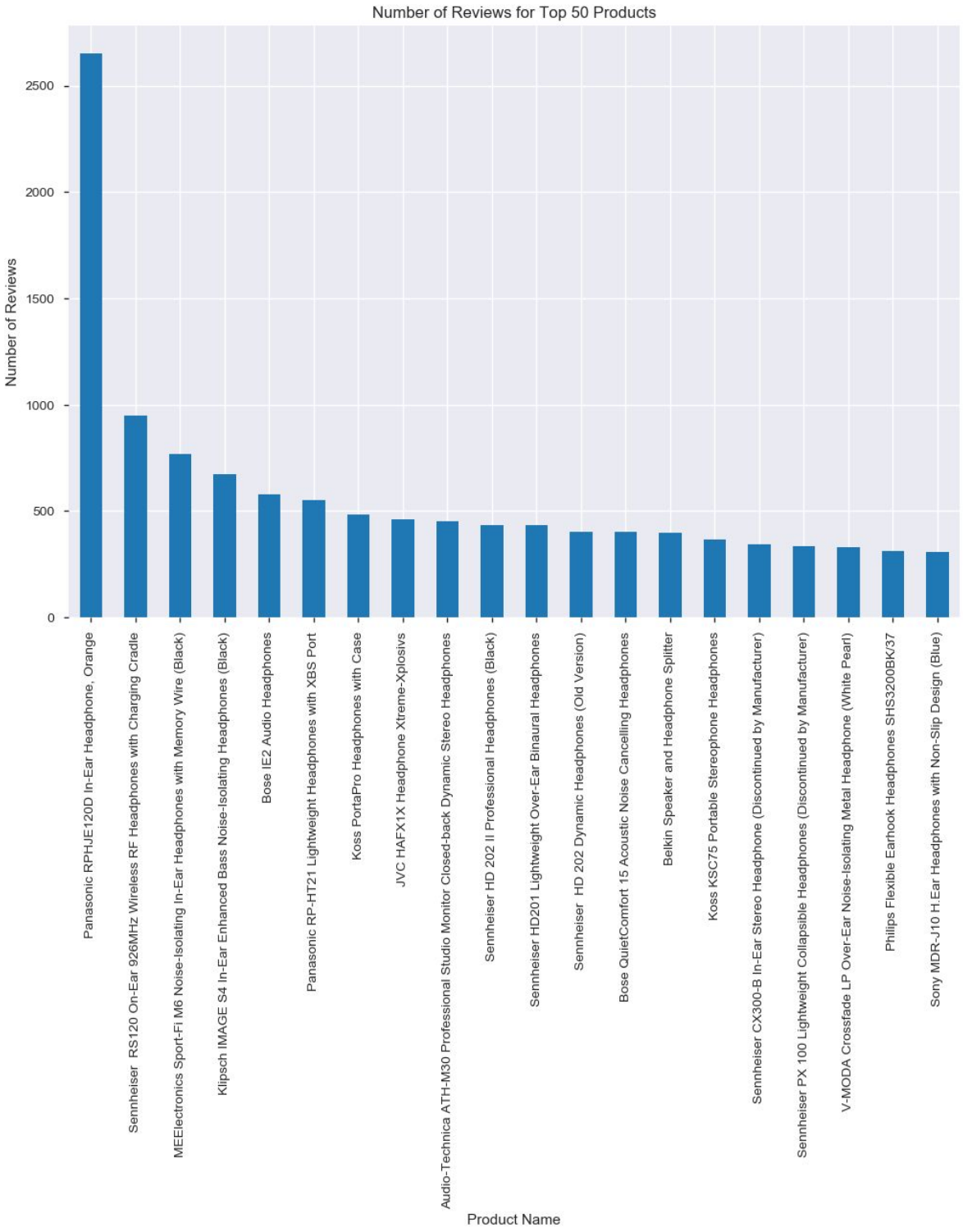


Fig 8. Top 20 most reviewed products

4. **Most Positively Reviewed Headphone**

Below product was the most reviewed product in Amazon under headphones category. Product name is “Panasonic ErgoFit In-Ear Earbud Headphones RP-HJE120-D (Orange) Dynamic Crystal Clear Sound, Ergonomic Comfort-Fit”. This product had overall good rating more than 3.

Fig 9a. Most reviewed headphone

5. **How the above product will go up after some years?**

The panasonic earbud headphone had overall positive review from 2010 onwards. Fig 9b shows the distribution of rating over a period of time. This product had overall good mean rating more than 4.

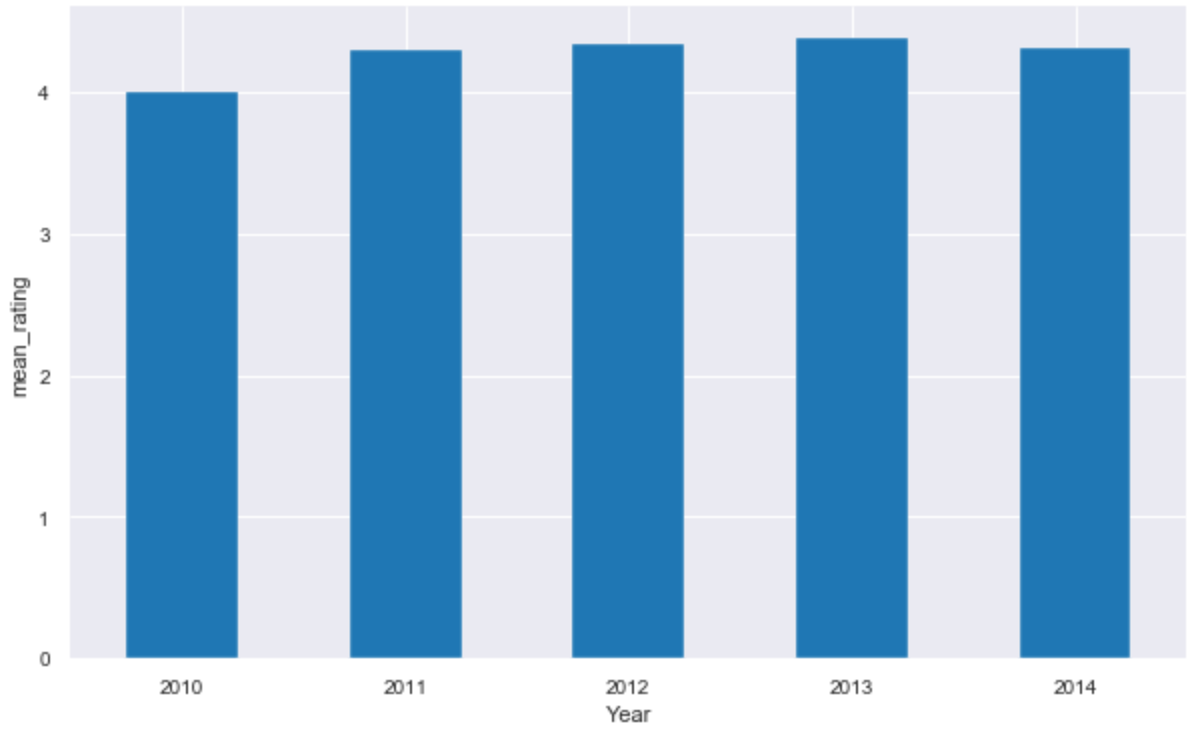


Fig 9b. Year vs mean rating for Panasonic earbud headphone

Fig 9c shows the word cloud from good rating reviews for the above product. It shows major insight in terms of sellers perspective. **It indicates most of the positive customers agree with "great fit", "good price" and least with "sound quality".** Similarly, Fig 9d shows The shows the word cloud from bad rating reviews for the above product. **It indicates most of the customers agree with "poor quality" and "terrible sound". From the sellers perspective, this product needs to be updated with "better sound" and "quality" in order to get positive feedback from customers.**

Fig 9c. Insight words from good rating reviews for Panasonic headphone

Fig 9d. Insight words from bad rating reviews for Panasonic headphone

6. Most Negatively Reviewed Headphone

Below product was the most negatively reviewed product in Amazon under headphones category. Product name is “My Zone Wireless Headphones”. This product had overall bad rating less than 3.

Fig 10a. Least reviewed headphone

7. **How the above product will go up after some years?**

My zone wireless headphone had overall negative review from 2010 onwards except 2012. Fig 10b shows the distribution of rating over a period of time. This product had overall bad mean rating of around 2.5.

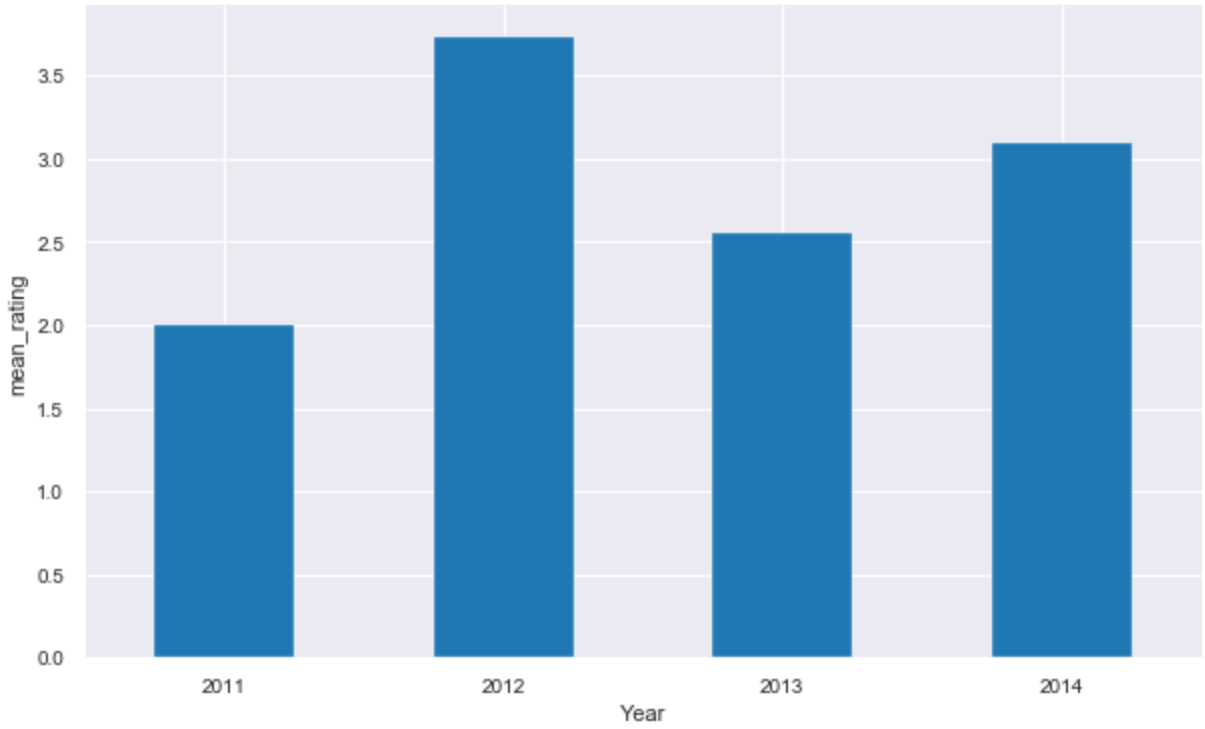


Fig 10b. Year vs mean rating for My zone wireless headphone

Fig 10c shows the word cloud from good rating reviews for the above product. It shows major insight in terms of sellers perspective. **It indicates most of the positive customers agree with "easy setup", "work with TV" and least agree with "work great".** Similarly, Fig10d shows The shows the word cloud from bad rating reviews for the above product. **It indicates most of the customers agree with "battery issue" and "horrible reception" and "static interference". From the sellers perspective, this product needs to be updated with "good quality battery", "reception issue" and "static issue" in order to get positive feedback from customers.**

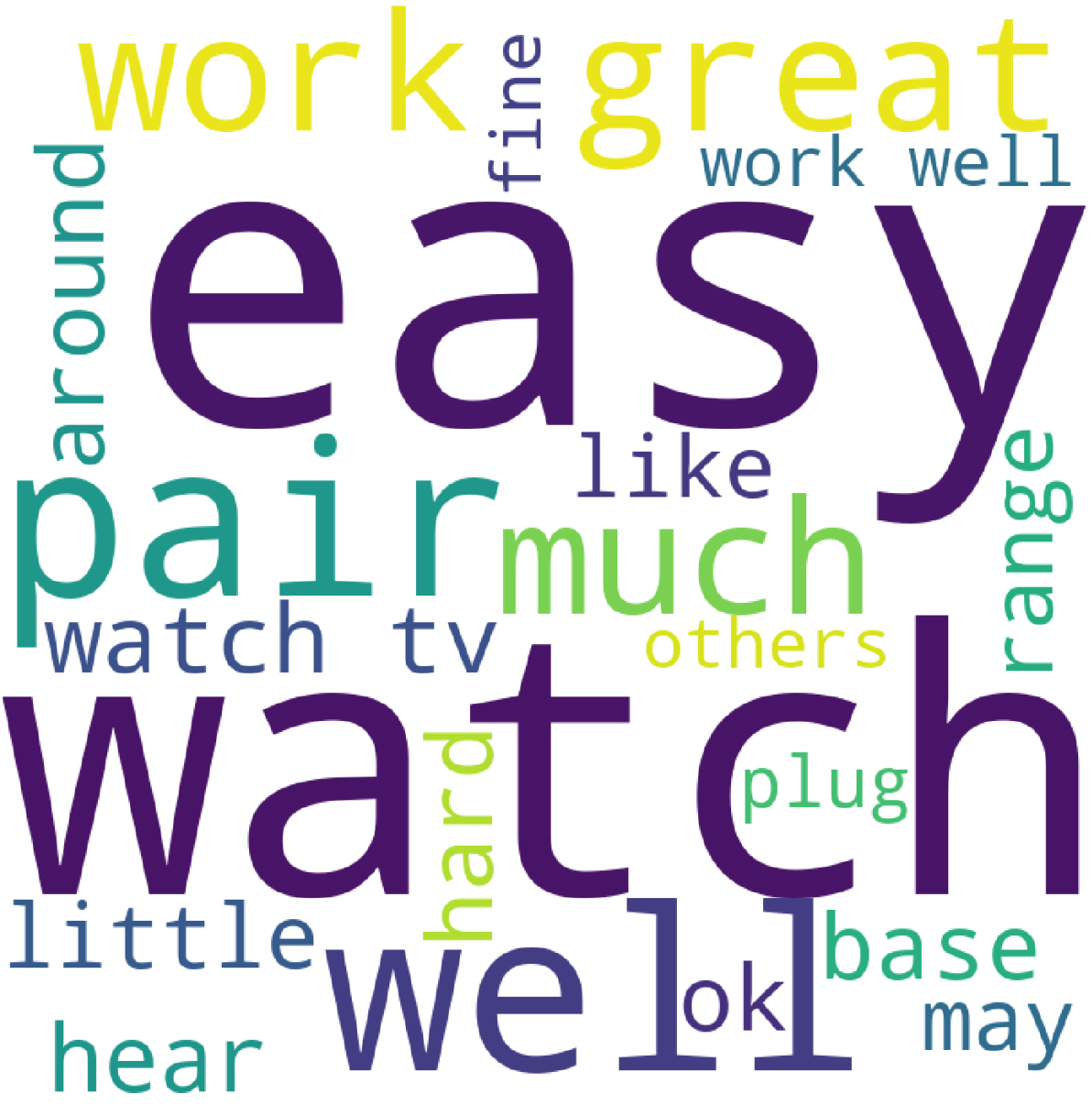


Fig 10c. Insight words from good rating reviews for My zone wireless headphone

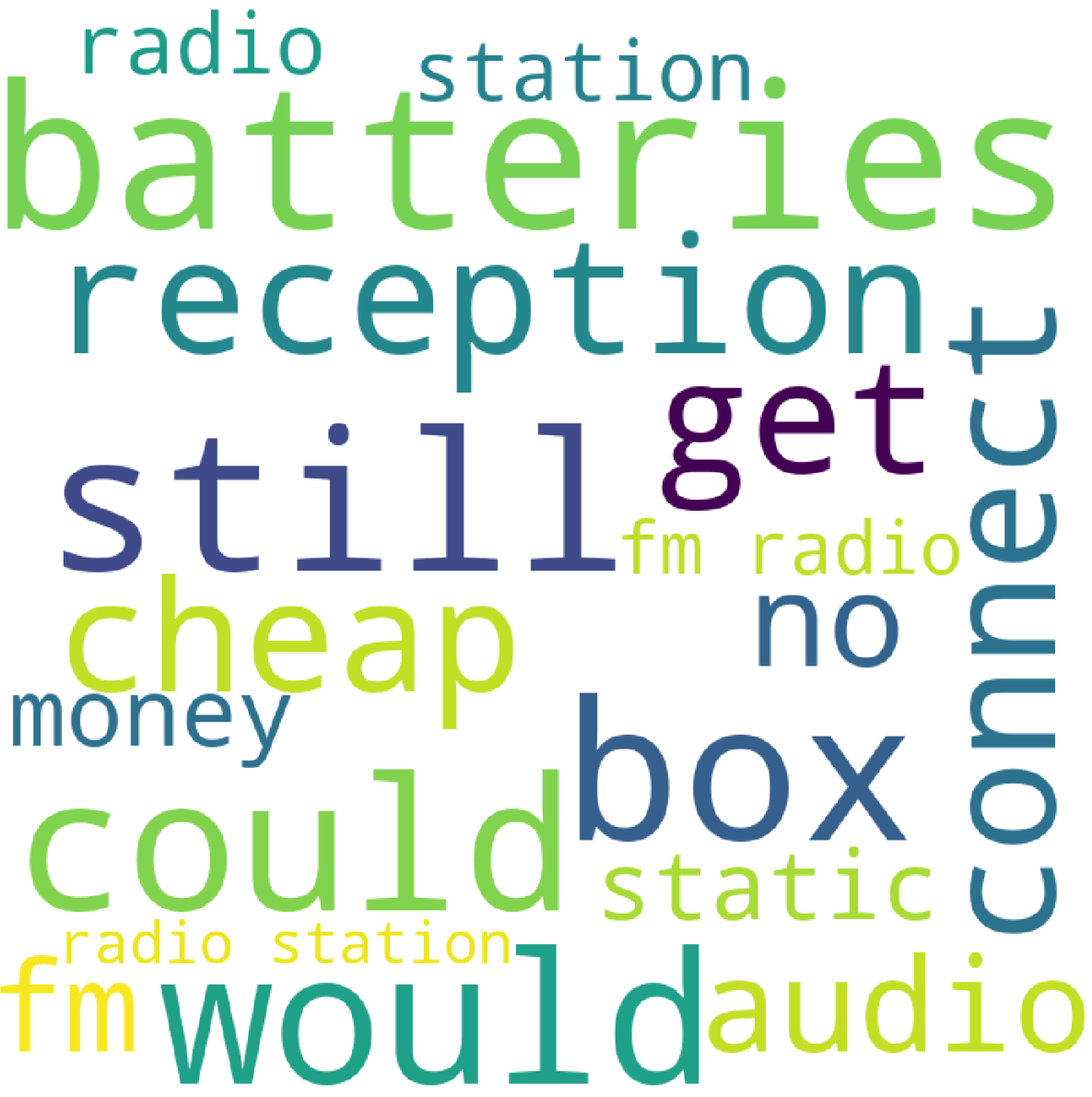


Fig 10d. Insight words from bad rating reviews for My zone wireless headphone

8. Which Rating got Highest Number of Reviews?

Customers have written reviews and ratings were given from 1 to 5 for headphones they bought from Amazon between 2000 to 2014. The distribution and percentage of ratings vs number of reviews is shown in Fig.11. Number of reviews for rating 5 were high compared to other ratings. Overall, customers were happy about the products they purchased. About 50% customers gave 5 rating for the products they purchased. Only 15% customers gave ratings less than 3 (Fig 12).

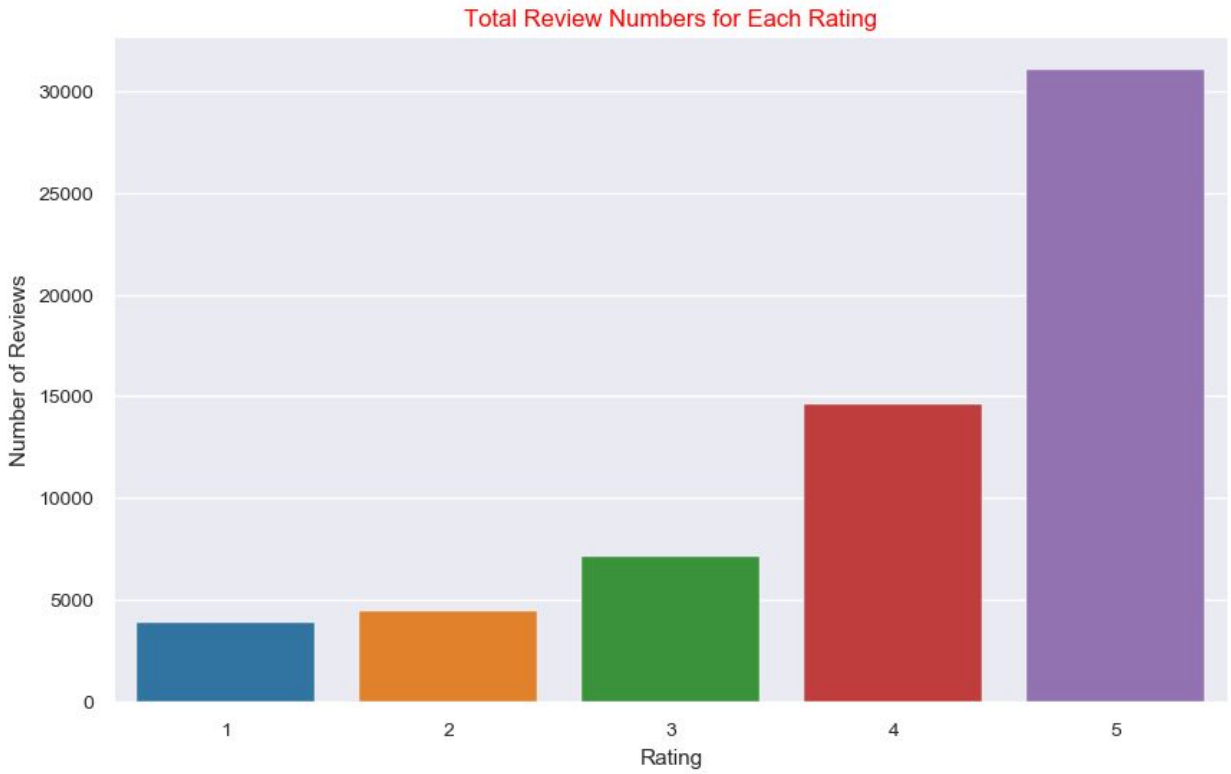


Fig 11. Rating Vs Number of reviews

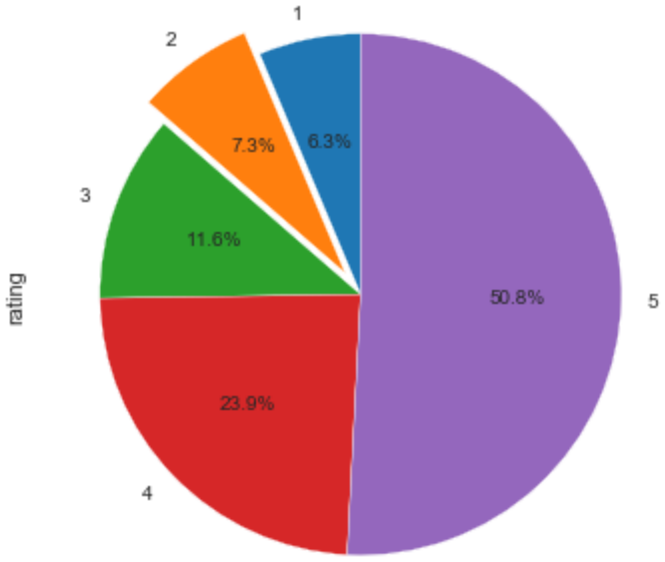


Fig 12. Distribution of ratings Vs Number of reviews

The ratings were divided into two categories. The rating below 3 were classified as “bad” and the remaining ratings were grouped as “good”. The distribution of rating class vs number of reviews is shown in Fig 13. It indicates about 50000 reviews were identified as good rating.

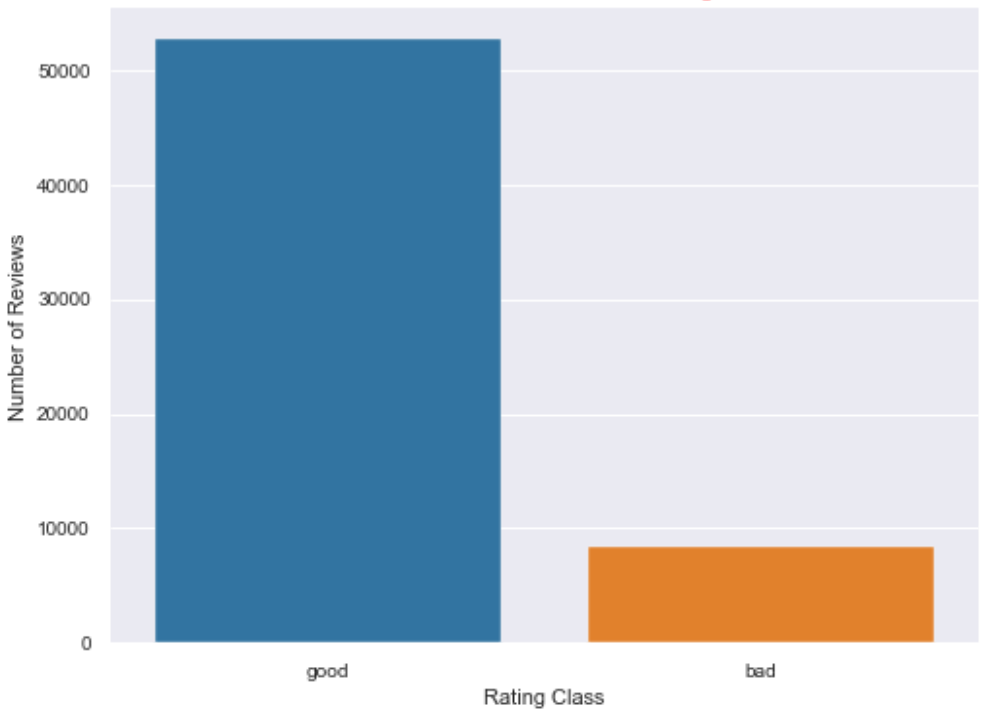


Fig 13. Total review numbers for each rating class

The distribution of ratings vs helpfulness ratio is shown in Fig 14. It indicates that all ratings have same helpfulness ratio.

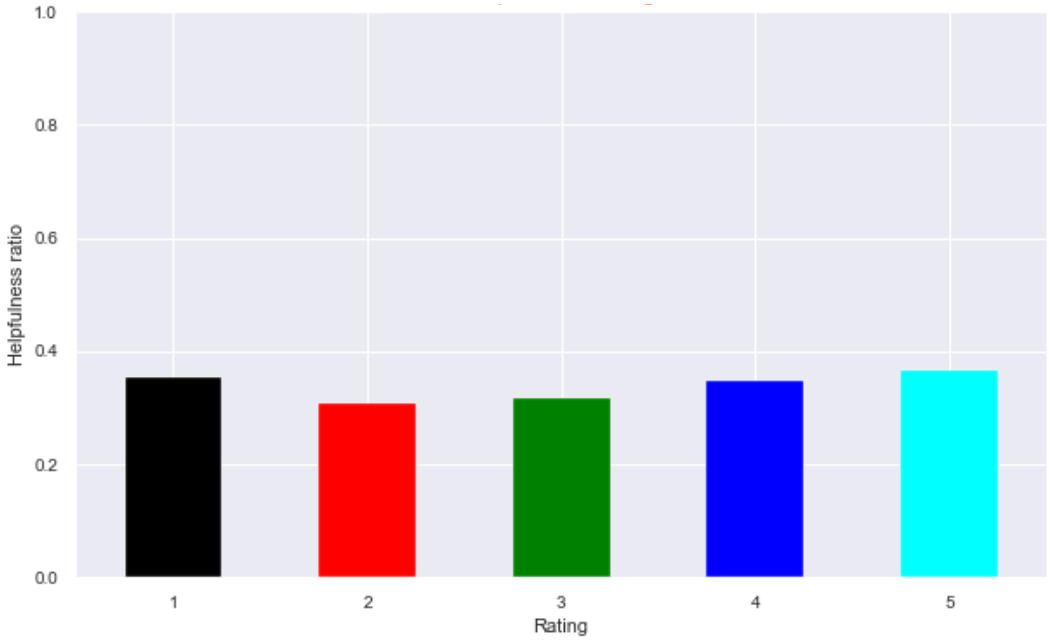


Fig 14. Helpfulness in rating

9. Which Rating got Highest Number of Review Length?

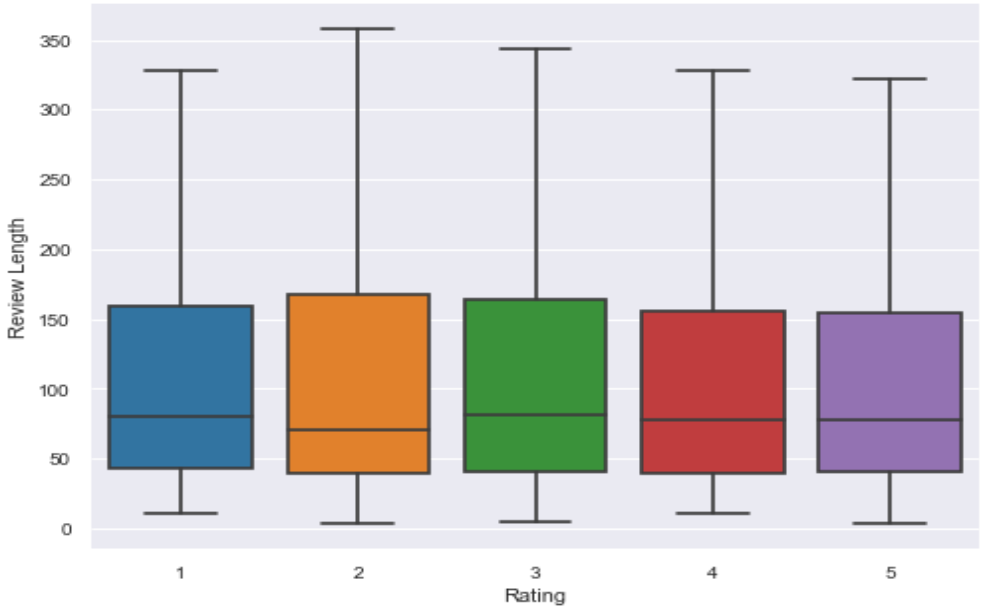


Fig 15. Rating vs Review length

10. Which Year has the Highest Number of Reviews?

Total review numbers for each year is shown in Fig 16. Number of reviews were low during 2000-2010. 2013 has the highest number of reviews.

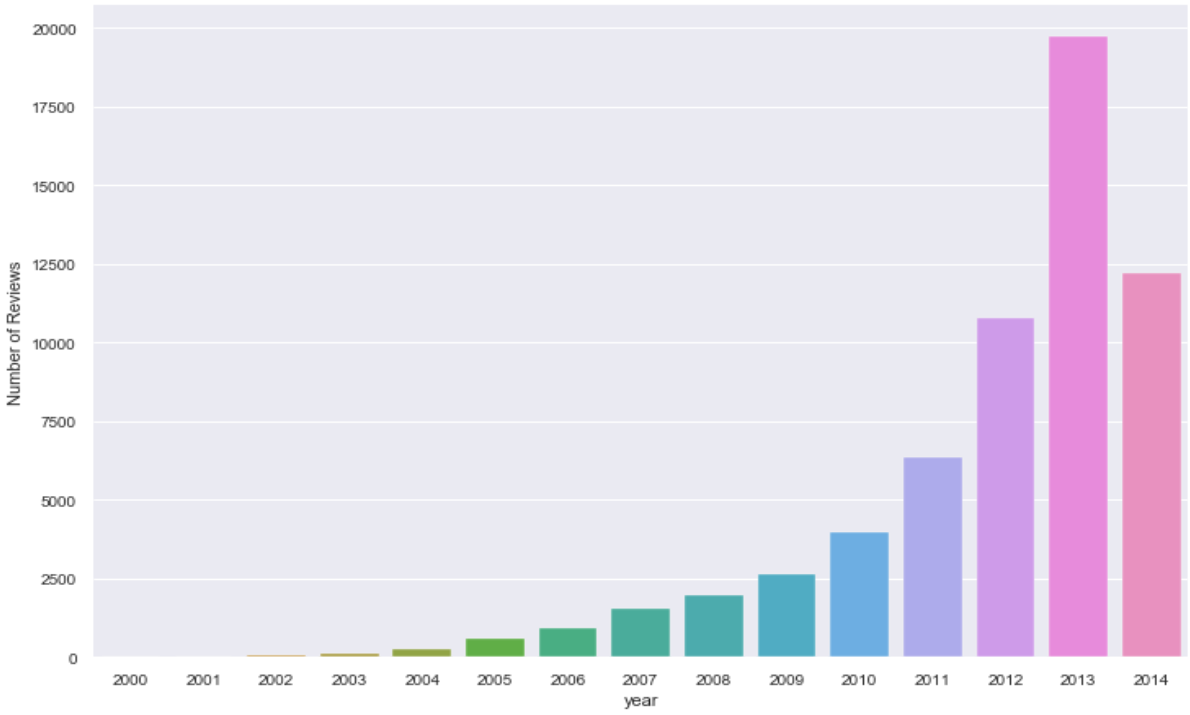


Fig 16. Rating vs Review length

11. Which Year has the Highest Number of Customers?

Total unique customers for each year is shown in Fig 17. Number of unique customers were low during 2000-2010. 2013 has the highest number of customers.

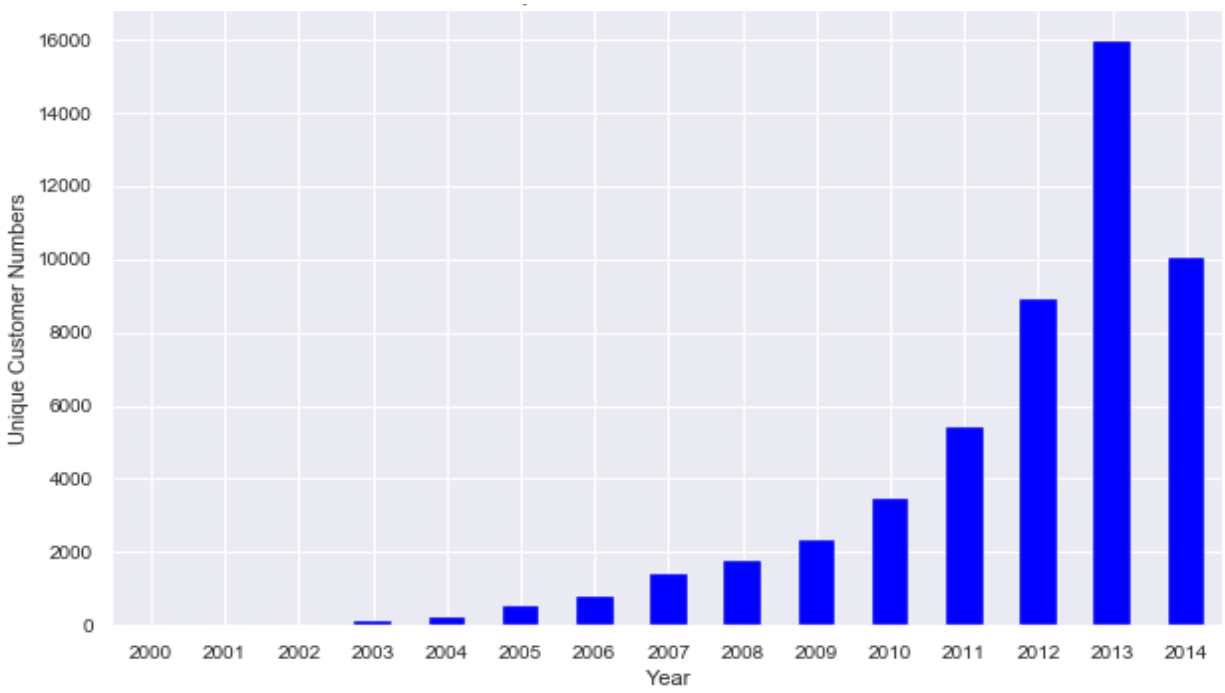


Fig 17. Unique customers in each year

12. Which Year has the Highest Number of Product?

Total unique product numbers for each year is shown in Fig 18. Number of unique products were low during 2000-2010. 2013 has the highest number of products.

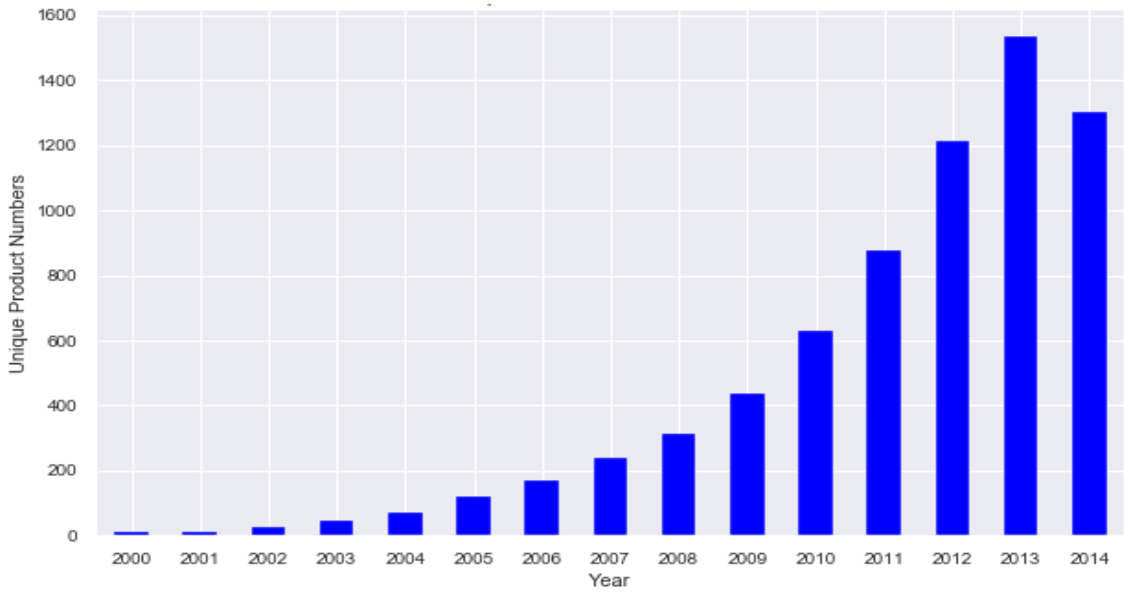


Fig 18. Unique products in each year

13. Which Year has the Highest Number of Product?

Except 2001, 'good ratings' percentage is progressing over 80%. 2001 has the lowest good ratings with 69% overall. 'good ratings' percentage is 90% in 2000. As it might be seen in the graph, the overall good rating is progressing between 81% and 90% in headphones products.

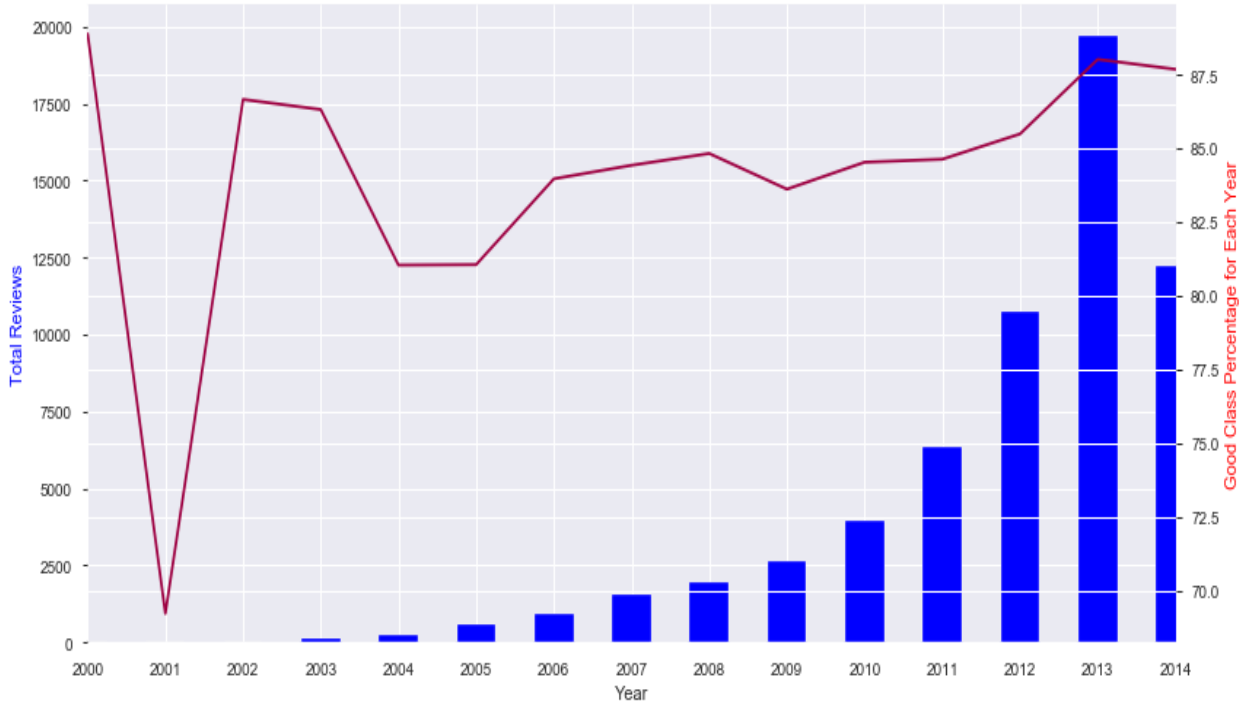


Fig 19. Distribution of good rating class over the period of time

14. Average Helpfulness of the Product

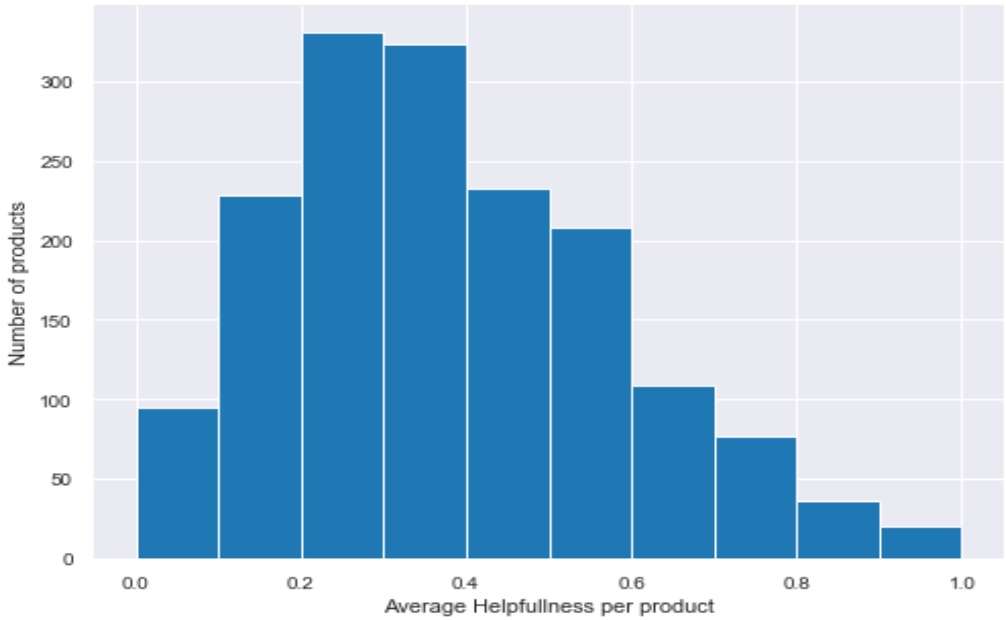


Fig 20. Distribution of average helpfulness per product

15. Distribution of Review Length

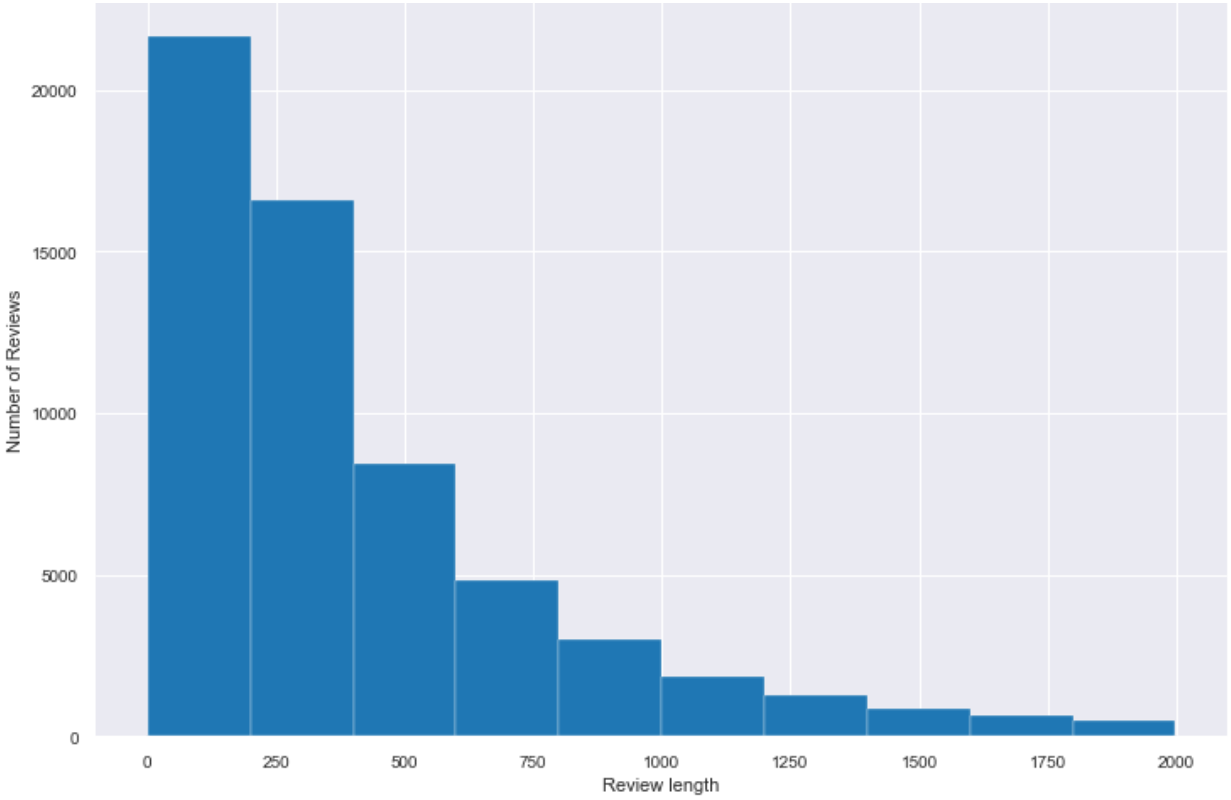


Fig 21. Distribution of review length

16. Which review length bin has the Highest Number of Good Rating?

As it might be seen in the Fig 22, the highest percentage of good rating reviews lies between 0-1000 words with 96 % whereas lowest percentage of good rating review lies between 1700-1800 words with 80%. As the review length extends, the good rating tends to increase. Generally, the customers who have write longer reviews (more than 1900 words) tends to give good ratings.

Fig 22. Distribution of review length vs good rating

17. Which review length bin has the Highest Number of Helpfulness Ratio?

As it might be seen in the Fig 23, the highest helpfulness ratio lies between 0-1200 words with 0.8 whereas lowest helpfulness ratio lies between 1200-1300 words with 0.6. As the review length extends, the helpfulness ratio tends to increase. Generally, the customers who have write longer reviews (more than 1300 words) tends to have high helpfulness ratio.

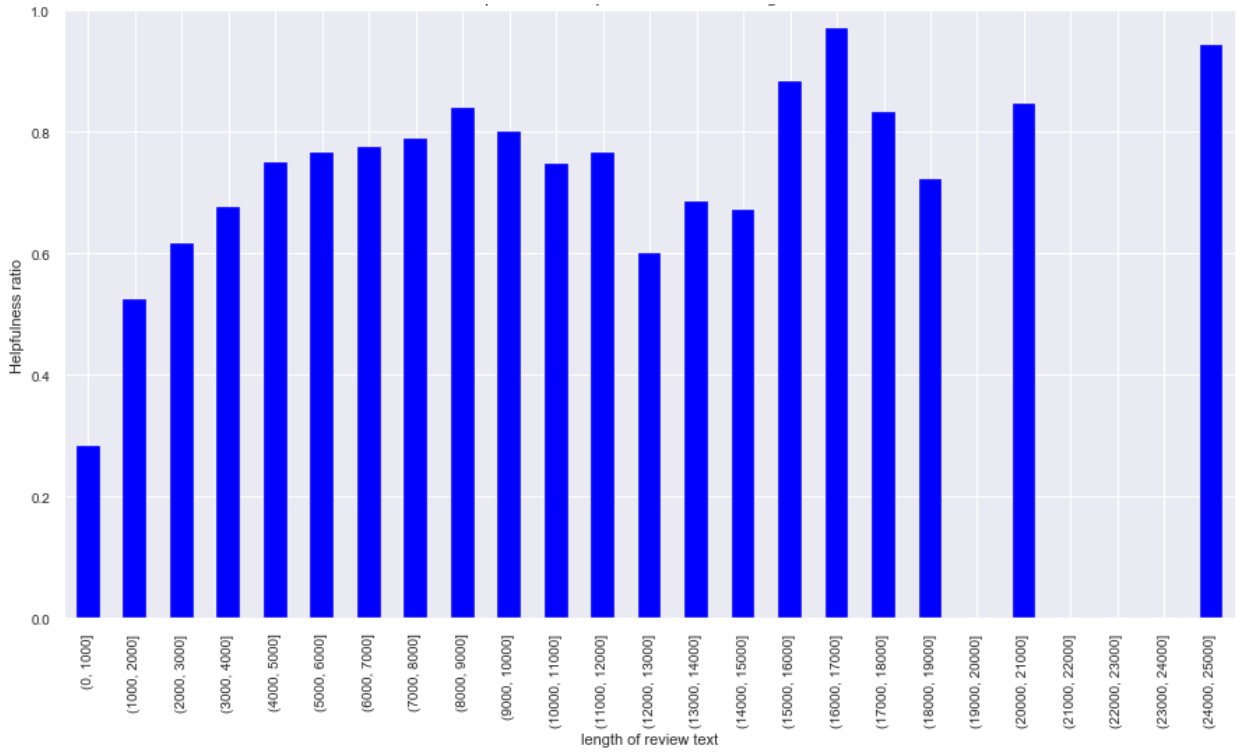


Fig 23. Distribution of review length vs helpfulness ratio

The frequency of review length for helpfulness and unhelpfulness is shown in Fig. 24. It indicates that overall helpfulness and unhelpfulness ratio were the same for larger review length. Unhelpfulness ratio were high in case of small length review.

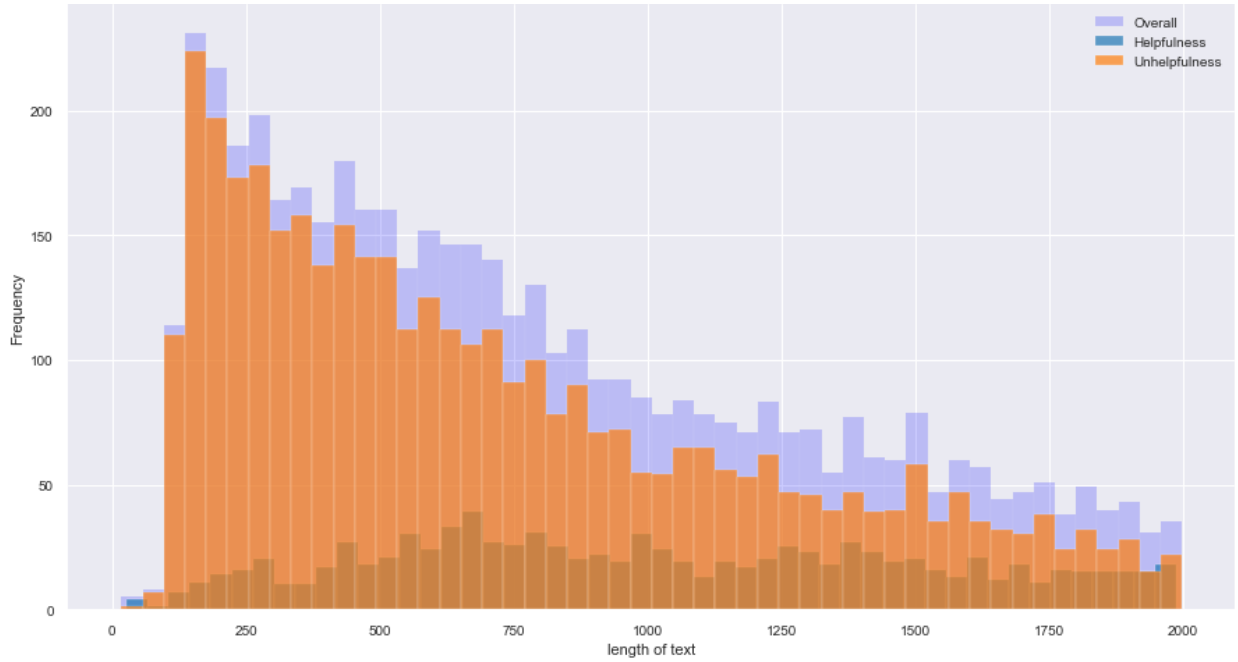


Fig 24. Distribution of review length vs helpfulness and unhelpfulness

18. Top Words for Good Rating?

The most common 50 words, which belong to good rating class, are shown in Fig 25. It shows the all good words from customers about the products.

Fig 25. Good rating words

19. Top Words for Bad Rating?

Similarly, the most common words, which belong to bad rating class, are shown in Fig 26. It shows all bad rating words from customers about the products.

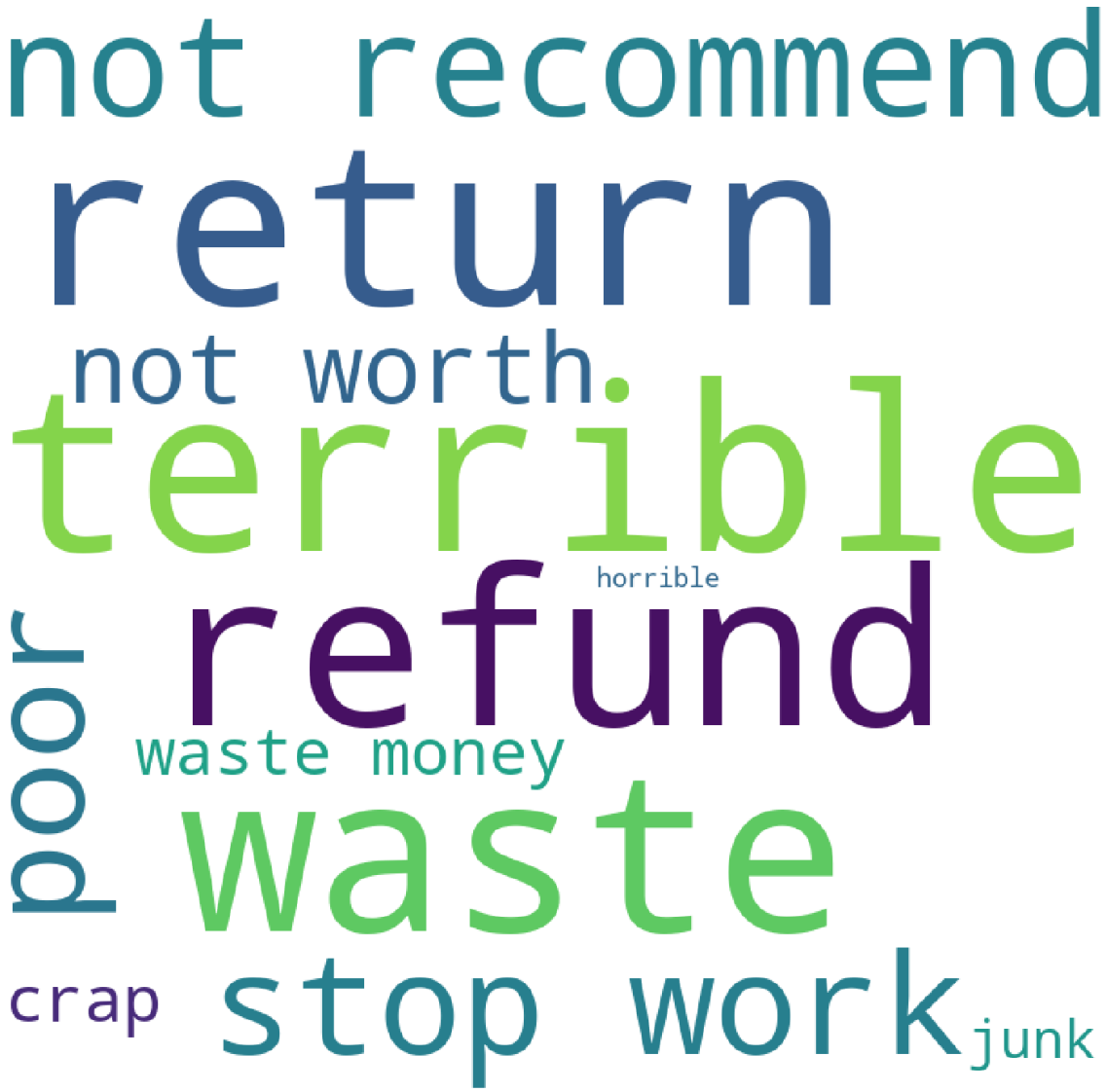


Fig 26. Bad rating words

Code:

Data Wrangling:
https://github.com/umaraju18/Capstone_project_2/blob/master/code/Amazon-Headphones_data_wrangling.ipynb

EDA:
https://github.com/umaraju18/Capstone_project_2/blob/master/code/Amazon-headphones_EDA.ipynb

SENTIMENT ANALYSIS:

Machine Learning models take numerical values as input. The reviews are made of sentences, so in order to extract patterns from the data; we need to find a way to represent it in a way that machine learning algorithm can understand, i.e. as a list of numbers.

FEATURE EXTRACTION:

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Features are usually numeric in nature and can be absolute numeric values or categorical features that can be encoded as binary features for each category in the list using a process called one-hot encoding. The process of extracting and selecting features is both art and science, and this process is called feature extraction or feature engineering.

As a part of this project, Bag of Words model, TF-IDF, Hashing Vectorizer, Word2Vec and adding most common words into the stopwords list, SMOTE, PCA, and Truncated SVD techniques into classification models in the following sections as a part of feature engineering and selection.

DATA PREPROCESSING:

Due to computational considerations, features with good_rating_class_reviews longer than 150 words reviewed earlier 2010 were removed. Final dataset consisted 15000 observations. From the dataset, "clean text" and "rating class" were treated as "X"(feature) and "Y"(variable) respectively. Dataset were divided into 75% as training and 25% as testing.

MACHINE LEARNING:

In this project, the model needs to predict sentiment based on the reviews written by customers who bought headphones from Amazon. This is a supervised binary classification problem. Python's Scikit libraries was used to solve this problem. Following machine learning algorithms were implemented.

1. Logistic Regression

Logistic regression, despite its name, is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.

2. Naive Bayes

Naive Bayes implements the naive Bayes algorithm for multinomial distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically

represented as word vector counts). This algorithm is a special case of the popular naïve Bayes algorithm, which is used specifically for prediction and classification tasks where we have more than two classes.

3. Random Forest Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).

4. XGBoost Classifier

XGBoost means eXtreme Gradient Boosting. XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now.

5. CatBoost Classifier

CatBoost is an algorithm for gradient boosting on decision trees. “CatBoost” name comes from two words “Category” and “Boosting”. the library works well with multiple Categories of data, such as audio, text, image including historical data.

EVALUATION METRICS:

Since we have a data imbalance in our case, the evaluation of the classifier performance must be carried out using adequate metrics in order to consider the class distribution and to pay more attention to the minority class. Based on this thought f1 score was used, which is harmonic average of precision and recall as my evaluation metric.

Understanding the types of errors our model makes are important. A good way to visualize that information is using a Confusion Matrix, which compares the predictions our model makes with the true label. With that in mind, confusion matrix was used besides our evaluation metric (f1 score).

MODELING:

Since the ratings of the reviews were not distributed normally, rating classes from 1-2 were classified as ‘Bad’ and Rating 3-4-5 were classified as 'Good'. For feature selection, threshold for word occurrence with using min_df/max_df, PCA and Singular Value Decomposition were applied. For feature engineering, CountVectorizer, TF-IDF, Hashing Vectorizer and Word2Vec were applied to the text data in order to turn a collection of text documents into numerical feature vectors.

1. Bag of Words Model

The Bag of Words model is perhaps one of the simplest yet most powerful techniques to extract features from text documents. This specific strategy (tokenization, counting and normalization) is called the Bag of Words or "Bag of n-grams" representation. The essence of this model is to convert text documents into vectors such that each document is converted into a vector that represents the frequency of all the distinct words that are present in the document vector space for that specific document. Fig 27. shows that **Logistic Regression is the winner with 0.896267 .**

				precision	recall	f1-score	support
vectorizer	model	accuracy	class				
CountVect	LogReg	0.896267	bad	0.688581	0.833799	0.754264	716.0
			good	0.958724	0.911009	0.934257	3034.0
			average	0.907144	0.896267	0.899891	3750.0
	Random Forest	0.857867	bad	0.969231	0.263966	0.414929	716.0
			good	0.851758	0.998022	0.919108	3034.0
			average	0.874188	0.857867	0.822843	3750.0
	Naive Bayes	0.898400	bad	0.790295	0.636872	0.705336	716.0
			good	0.918059	0.960119	0.938618	3034.0
			average	0.893664	0.898400	0.894077	3750.0
	XGBoost	0.890933	bad	0.880893	0.495810	0.634495	716.0
			good	0.892142	0.984179	0.935903	3034.0
			average	0.889994	0.890933	0.878355	3750.0
	CatBoost	0.896800	bad	0.818182	0.590782	0.686131	716.0
			good	0.909372	0.969018	0.938248	3034.0
			average	0.891961	0.896800	0.890111	3750.0

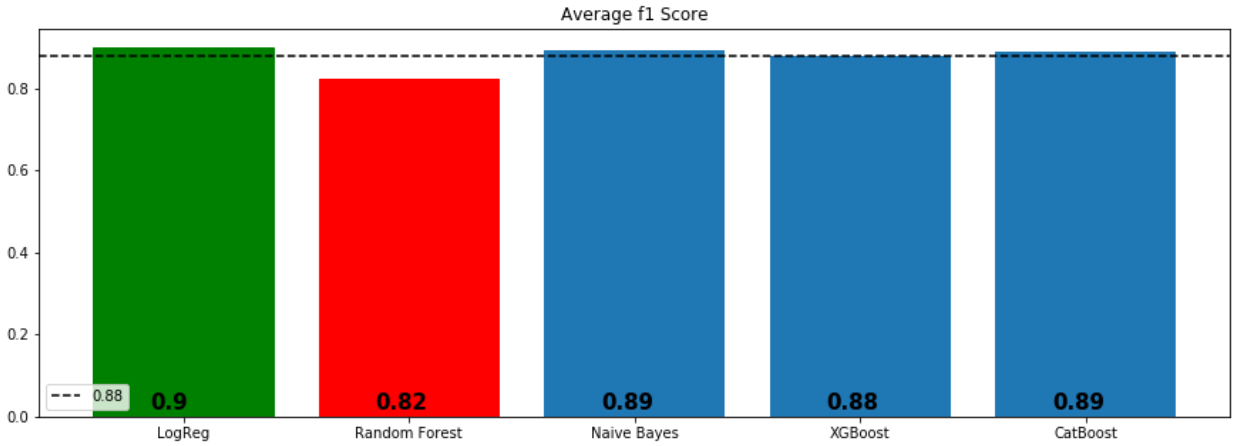


Fig 27. Average F1 score

2. TF-IDF Model

TF-IDF stands for Term Frequency-Inverse Document Frequency, a combination of two metrics: term frequency and inverse document frequency. In order to focus more on meaningful words, TF-IDF score (Term Frequency-Inverse Document Frequency) was used on top of our Bag of Words model. TF-IDF weighs words by how rare they are in our dataset, discounting words that are too frequent and just add to the noise. -IDF works by penalizing these common words by assigning them lower weights while giving importance to words which appear in a subset of a particular document. Fig 28. Shows that **CatBoosting is the winner with 0.896533 score.**

				precision	recall	f1-score	support
vectorizer	model	accuracy	class				
CountVect	LogReg	0.866933	bad	0.602070	0.893855	0.719505	716.0
			good	0.971716	0.860580	0.912777	3034.0
			average	0.901138	0.866933	0.875875	3750.0
	Random Forest	0.852533	bad	0.988024	0.230447	0.373726	716.0
			good	0.846218	0.999341	0.916427	3034.0
			average	0.873294	0.852533	0.812808	3750.0
	Naive Bayes	0.809600	bad	1.000000	0.002793	0.005571	716.0
			good	0.809498	1.000000	0.894721	3034.0
			average	0.845872	0.809600	0.724953	3750.0
	XGBoost	0.892267	bad	0.902062	0.488827	0.634058	716.0
			good	0.891136	0.987475	0.936836	3034.0
			average	0.893222	0.892267	0.879025	3750.0
	CatBoost	0.896533	bad	0.805970	0.603352	0.690096	716.0
			good	0.911637	0.965722	0.937900	3034.0
			average	0.891461	0.896533	0.890586	3750.0

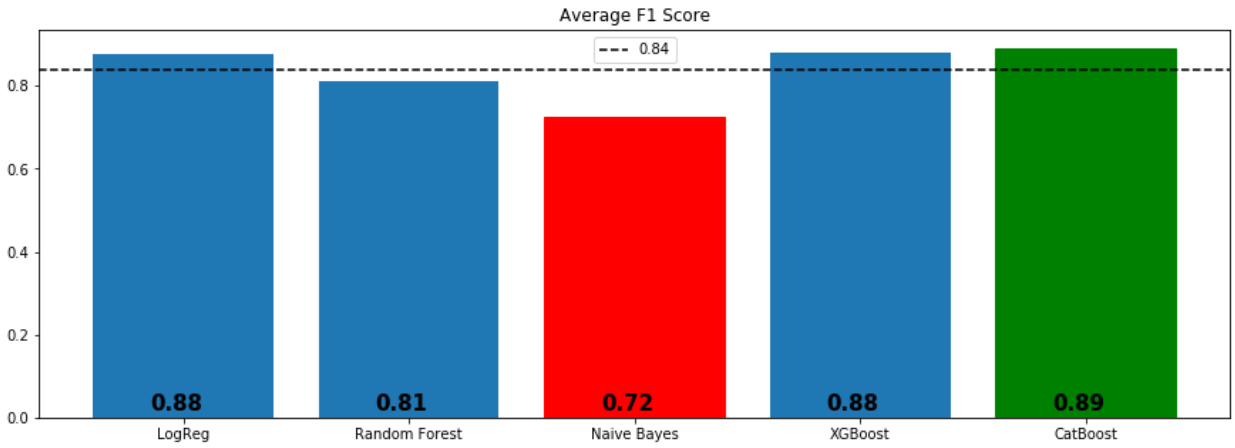


Fig 28. Average F1 score

3. Hash Vectorizer

Hash Vectorizer is designed to be as memory efficient as possible. Instead of storing the tokens as strings, the vectorizer applies the hashing trick to encode them as numerical indexes. The downside of this method is that once vectorized, the features' names can no longer be retrieved. Fig 29 shows that **CatBoosting is the winner with 0.894133 score.**

				precision	recall	f1-score	support
vectorizer	**model**	**accuracy**	**class**				
CountVect	**LogReg**	**0.842400**	**bad**	0.556256	0.863128	0.676519	716.0
			good	0.962865	0.837508	0.895822	3034.0
			average	0.885229	0.842400	0.853950	3750.0
	Random Forest	**0.870667**	**bad**	0.971429	0.332402	0.495317	716.0
			good	0.863623	0.997693	0.925830	3034.0
			average	0.884207	0.870667	0.843630	3750.0
	Naive Bayes	**0.816000**	**bad**	0.964286	0.037709	0.072581	716.0
			good	0.814884	0.999670	0.897869	3034.0
			average	0.843410	0.816000	0.740294	3750.0
	XGBoost	**0.889867**	**bad**	0.912807	0.467877	0.618652	716.0
			good	0.887378	0.989453	0.935640	3034.0
			average	0.892233	0.889867	0.875116	3750.0
	CatBoost	**0.894133**	**bad**	0.812133	0.579609	0.676447	716.0
			good	0.907070	0.968359	0.936713	3034.0
			average	0.888943	0.894133	0.887019	3750.0

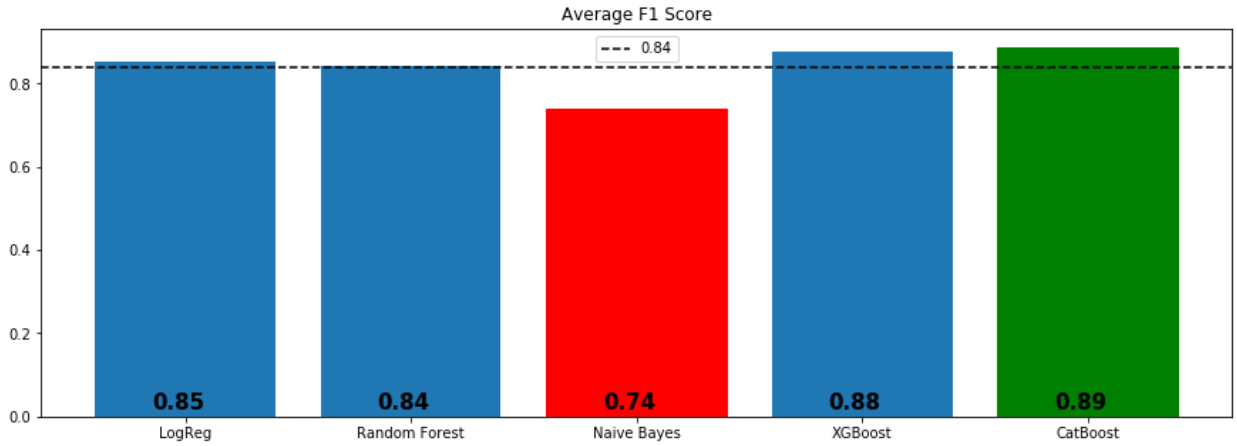


Fig 29. Average F1 score

4. Adding Most Common and Least Common Words to Stopwords List (Count Vectorizer)

Since there were not too many distinguisher words in different classes, the most and least common 70 words added to the stopwords list and models were applied in order to see any changes in evaluation metrics. Fig 30 shows that **CatBoosting is the winner with 0.890133 score**. Adding most and least common words to the stopword list didn't have impact on models' performance.

				precision	recall	f1-score	support
vectorizer	model	accuracy	class				
CountVect	LogReg	0.875200	bad	0.643519	0.776536	0.703797	716.0
			good	0.944560	0.898484	0.920946	3034.0
			average	0.887081	0.875200	0.879485	3750.0
	Random Forest	0.889333	bad	0.841270	0.518156	0.641314	716.0
			good	0.895739	0.976928	0.934574	3034.0
			average	0.885339	0.889333	0.878580	3750.0
	Naive Bayes	0.881600	bad	0.675258	0.731844	0.702413	716.0
			good	0.935440	0.916941	0.926099	3034.0
			average	0.885763	0.881600	0.883389	3750.0
	XGBoost	0.877600	bad	0.890578	0.409218	0.560766	716.0
			good	0.876352	0.988134	0.928892	3034.0
			average	0.879068	0.877600	0.858605	3750.0
	CatBoost	0.890133	bad	0.819328	0.544693	0.654362	716.0
			good	0.900428	0.971655	0.934686	3034.0
			average	0.884943	0.890133	0.881163	3750.0

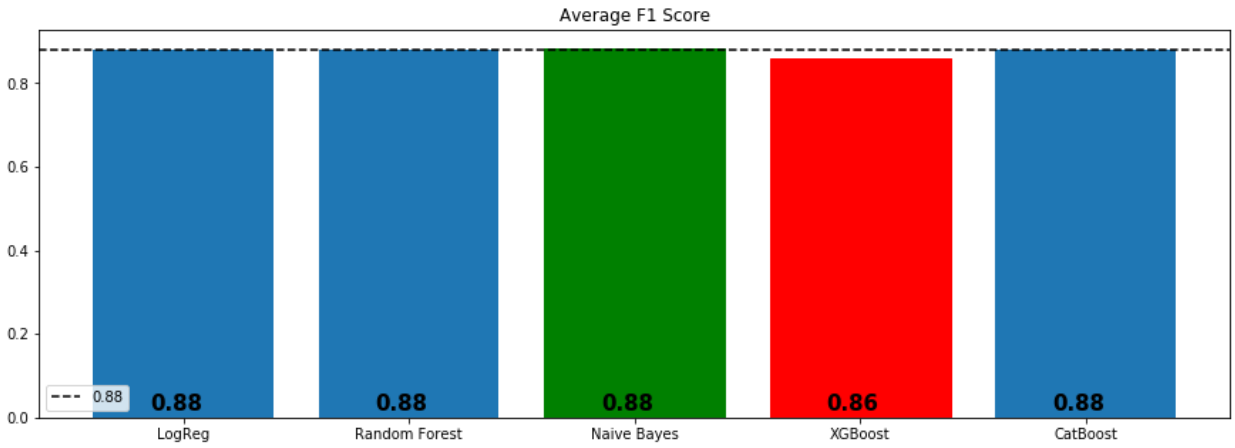
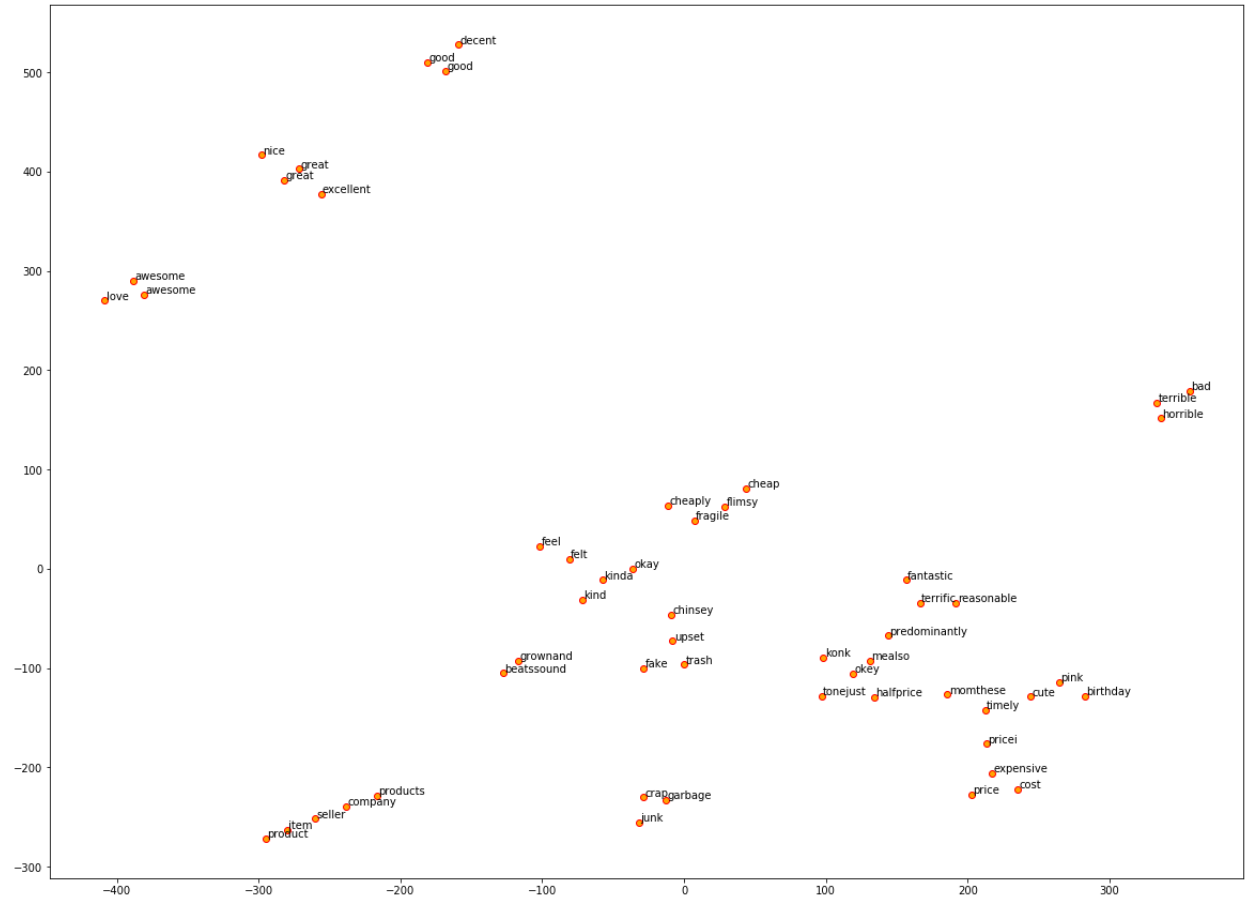


Fig 30. Average F1 score

5. Applying Word2Vec and Simple Neural Network

We created word vectors using Word2Vec and the model has 26548 unique words where each word has a vector length of 100. Then we used these dense vectors - word embeddings - in a simple neural network to predict. In training and validation accuracy graph, the model starts to overfit after 1st epoch. The accuracy for this simple neural network is 0.7992.

```
{'feel': ['felt', 'grownand', 'beatssound', 'kind', 'kinda'],
 'good': ['decent', 'great', 'predominantly', 'reasonable', 'nice'],
 'product': ['item', 'products', 'company', 'timely', 'seller'],
 'cheap': ['flimsy', 'chinsey', 'cheaply', 'fragile', 'mealso'],
 'junk': ['crap', 'garbage', 'upset', 'fake', 'trash'],
 'bad': ['terrible', 'horrible', 'okay', 'konk', 'okey'],
 'great': ['fantastic', 'excellent', 'good', 'awesome', 'terrific'],
 'price': ['cost', 'tonejust', 'expensive', 'pricei', 'halfprice'],
 'love': ['awesome', 'pink', 'cute', 'birthday', 'momthese']}
```



Visualization of the words of interest and their similar words using their embedding vectors after reducing their dimensions to a 2-D space with t-SNE is presented above. Similar words based on gensim's model can be viewed as well.

PRODUCT RECOMMENDATION:

Till recently, people generally tended to buy products recommended to them by their friends or the people they trust. This used to be the primary method of purchase when there was any doubt about the product. But with the advent of the digital age, that circle has expanded to include online sites that utilize some sort of recommendation engine.

A recommendation engine filters the data using different algorithms and recommends the most relevant items to users. It first captures the past behavior of a customer and based on that, recommends products which the users might be likely to buy.

If we can recommend a few items to a customer based on their needs and interests, it will create a positive impact on the user experience and lead to frequent visits. Hence, businesses nowadays are building smart and intelligent recommendation engines by studying the past behavior of their users. In this project, item-item collaborative filtering was used.

DATA PROCESSING:

After merging electronics rating dataset with product metadata, null values were removed from the dataset. Total features were 7530925. The final dataset is shown below.

	userID	prod_ID	rating	prod_name
0	AKM1MP6P0OYPR	0132793040	5.0	Kelby Training DVD: Mastering Blend Modes in A...
1	A2CX7LUOHB2NDG	0321732944	5.0	Kelby Training DVD: Adobe Photoshop CS5 Crash ...
2	A2NWSAGRHCP8N5	0439886341	1.0	Digital Organizer and Messenger
3	A2WNBOD3WNDNKT	0439886341	3.0	Digital Organizer and Messenger
4	A1GI0U4ZRJA8WN	0439886341	1.0	Digital Organizer and Messenger

Fig 31. Final Dataset for product recommendation

ITEM-ITEM COLLABORATIVE FILTERING:

This collaborative filtering is useful when the number of users is more than the items being recommended. In this project, the number of users (4053964) is more than the number of items (469625).

In this filtering, the similarity between each item pair was computed and based on that, similar items were recommended which are liked by the users in the past. The weighted sum of ratings of "item-users" were taken. The tem based filtering process is shown in Fig 32.

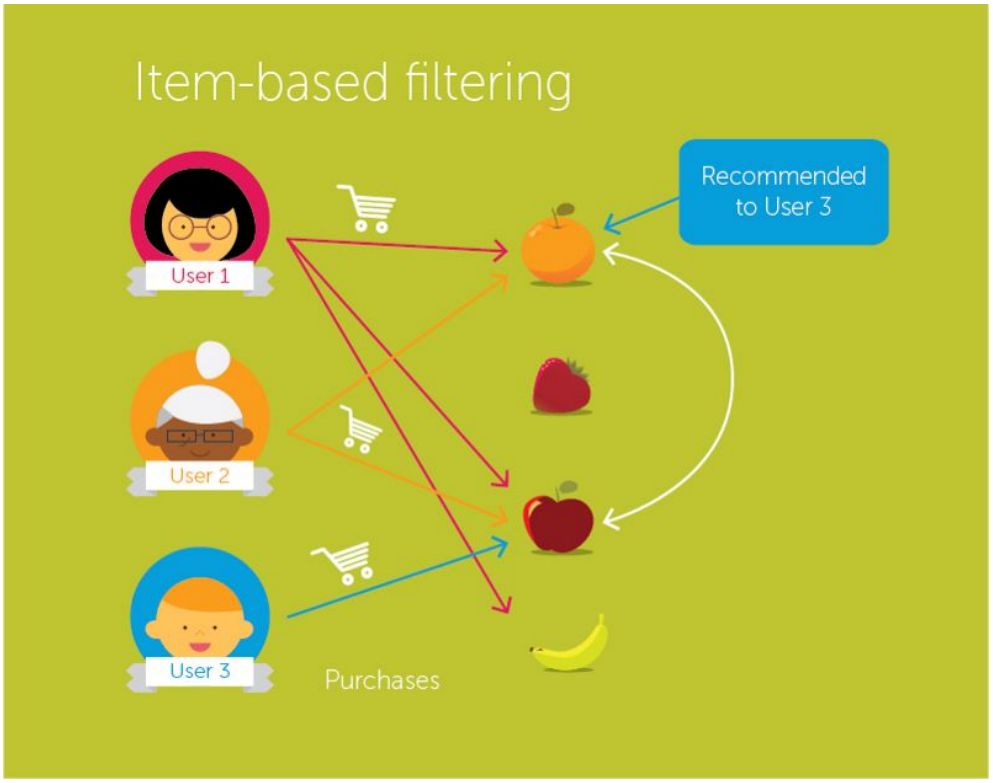


Fig 32. Item based filtering recommendation

1. **Top 10 Popular Products by Sum user Ratings**

	prod_ID	prod_name	ratings_sum
3313486	B003ES5ZUU	AmazonBasics High-Speed HDMI Cable - 15 Feet (...	846.0
6104981	B0088CJT4U	TP-LINK TL-WDR4300 Wireless N750 Dual Band Rou...	814.0
1198226	B000N99BBC	TP-LINK TL-SG1005D 10/100/1000Mbps 5-Port Giga...	755.0
5941380	B007WTAJTO	SanDisk Ultra 64GB MicroSDXC Class 10 UHS Memo...	741.0
6031403	B00829TIEK	Seagate Backup Plus 3TB USB 3.0 Desktop Extern...	626.0
6028195	B00829THK0	Seagate Backup Plus 1TB Desktop External Hard ...	560.0
6190152	B008DWCRQW	D-Link Wireless AC 1750 Mbps Home Cloud App-En...	524.0
4024382	B004CLYEDC	Micra Digital CAT5e Snagless Patch Cable, 5 Fe...	517.0
2796338	B002R5AM7C	Flip MinoHD Video Camera - Brushed Metal, 8 GB...	514.0
2883526	B002V88HFE	eneloop SEC-CSPACER4PK C Size Spacers for use ...	475.0

Fig 33. Top 10 popular products by sum user ratings

2. Product Recommendation

```
User A100WO06OQR8BQ has already purchased 91 items.
Recommending the highest 5 predicted  items not already purchased.
```

	prod_ID	prod_name
11745	B004CLYEDC	Micra Digital CAT5e Snagless Patch Cable, 5 Fe...
5649	B000N99BBC	TP-LINK TL-SG1005D 10/100/1000Mbps 5-Port Giga...
5012	B004CLYEFK	Micra Digital USB A to USB B Cable (6 Feet)
1169	B00829THK0	Seagate Backup Plus 1TB Desktop External Hard ...
656	B00834SJSK	Seagate Expansion 500GB Portable External Hard...

Fig 34. Product recommendation

CONCLUSION:

In this project, the rating scores based on the reviews left by the customers were predicted using Count Vector, TF-IDF, Hashing Vector, Word2Vec,Classification Models and Simple Neural Network and Adding most and least common words to CountVect. From the analyses, it was found that CatBoosting with TF-IDF (f1 score is 0.890586) or Logistic Regression with Count Vectorizing (f1 score is 0.899891) were top models. Adding most and least common words to the stopword list didn't have impact on models' performance.

FUTURE STUDY:

- Using different methods in order to minimize the effect of the matching words
- Using different AutoML tools.
- Implementation of Dask library for parallel processing to decrease run time.

Code:

Sentiment Analysis:

https://github.com/umaraju18/Capstone_project_2/blob/master/code/Amazon_headphones_Sentiment_Analysis_CV_IF_IDF_HASH.ipynb

https://github.com/umaraju18/Capstone_project_2/blob/master/code/Amazon_headphones_wordvec.ipynb

Recommendation System:

https://github.com/umaraju18/Capstone_project_2/blob/master/code/amazon_electronics_recommendation.ipynb