Amazon's Customer Building Blocks –Reviews & Ratings

Amazon is the most recognized name in the modern world, no matter which State you go to in the US or which country you visit. This enterprise has achieved great results in the past few decades. The product and supply chain established at Amazon is one of a kind that has been studied by many experts and has set a standard for others in the retail ecommerce arena.

DATA OVERVIEW

- DATASET:

  The data used is historic and from Amazon available on the website http://jmcauley.ucsd.edu/data/amazon/.

- GOAL & QUESTIONS ASKED:

  Picking the **Star Ratings (labelled as 'overall' in the dataset)** and checking the dependency on **Review WordCount and Product Category** as feature dependencies is an attempt to be quantified in this study.

  Some of the questions asked are:

  1. Does the wordcount in the review text matter? Do more words mean higher overall Star Ratings?
  2. Do all Overall Star Ratings (1-5) have the same behavior across Product Categories i.e. Do Amazon Customers give more 1 stars to Baby products or more 5 stars to Android Apps?
  3. Can we predict Overall Star Ratings given the word count of the review text?

This data compilation has 2 subsets namely 1. Reviews and 2. Ratings.

1. Reviews Dataset

The first dataset allows for us to get an understanding of Reviews that are product based and Reviewer ID based. and the derived feature called 'WordCount_ReviewText' will help bring about good insights as to how the customer's review length and rating are dependent on each other. E.g. of the data types and counts is as follows:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 228967 entries, 0 to 231779
Data columns (total 10 columns):
reviewerName      228967 non-null object
asin              228967 non-null object
overall           228967 non-null float64
reviewerID        228967 non-null object
reviewTime        228967 non-null object
unixReviewTime    228967 non-null int64
helpful           228967 non-null object
summary           228967 non-null object
reviewText        228967 non-null object
ProductType       228967 non-null object
dtypes: float64(1), int64(1), object(8)
memory usage: 19.2+ MB
```

Amazon's Customer Building Blocks –Reviews & Ratings

2. Ratings Dataset

This second dataset we will use to define a timeline of the review and get the associated Star Rating dependency. Star Ratings maybe dependent on a seasonal, weekly or monthly cycle. Since the timeline of our data expands from 1996 to 2016, this should give us enough weeks, months and seasons to test out. The dataset is also segregated by Product Category, the Star Rating will be tested and predicted for each product category. e.g. Instant Videos, Apps for Android, Automotive etc. e.g. of the data types and counts is as follows:

```
<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1324752 entries, 0 to 1324751
Data columns (total 5 columns):
reviewerID        1324752 non-null object
asin              1324752 non-null object
overall           1324752 non-null float64
unixReviewTime    1324752 non-null int64
ProductType       1324752 non-null object
dtypes: float64(1), int64(1), object(3)
memory usage: 50.5+ MB
```

CLIENT AND PROPOSED APPROACH

The 'imaginary' client could be Amazon trying to get an understanding of its customer base as to what leads to getting the customer to provide a good online review which then translates to a good overall rating and proportional sales increase due to popularity of a product. Currently in the retail world, tasks like product placement, inventory management, customized offers, product bundling etc. are being smartly handled using data science techniques.

Amazon could use this predictive analytical knowledge to decide which third-party vendors/sellers they would extend or end contracts with. Should Amazon continue or discontinue certain products, or this data would help decipher if product expansion needs to be considered based on customer reviews. So, this begs the question how do we conduct a meaningful analysis?

We start with a dataset that has product information, customer review timings and the actual customer initiated worded reviews. This last feature will help us extract the wordcount which we will use for our study extensively together with predictive analysis to predict the overall rating of a product based on the review given.

The overall approach will be to use graphs, conduct exploratory data analysis, use analytical tools, go through the inferential statistics stage and running machine learning algorithms which help decide on which algorithm will be the best fit and provide good results for predictive purposes.

IMPORTING RELEVANT PACKAGES & MODULES

A common practice that was adopted was to use aliases for the imported packages and modules. The following is a list of the packages and modules imported.

1. Pandas
2. Gzip
3. MatplotLib[module pyplot]
4. MatplotLib[module style]

Amazon's Customer Building Blocks –Reviews & Ratings

5. Datetime
6. Calendar
7. Numpy
8. String [module Punctuation]
9. __future__ [module division]
10. Nltk.tokenize [module word_tokenize]
11. Nltk.tokenize [module sent_tokenize]
12. Nltk.tokenize [module regexp_tokenize]
13. Re
14. Seaborn
15. IPyhton.core.debugger [module Pdb]

DATA CLEANSING

Collecting the raw data from the website http://jmcauley.ucsd.edu/data/amazon/ . The Reviews dataset had a total of 24 files that were 'gzipped' and in the JSON format. The Ratings dataset had a total of 24 files that were in CSV format. These files were segregated by 'Product Category' i.e. 24 and they are as follows:

Product Categories	
Amazon Instant Video	Health and Personal Care
Apps for Android	Home and Kitchen
Automotive	Kindle Store
Baby	Movies and TV
Beauty	Musical Instruments
Books	Office Products
CDs and Vinyl	Patio Lawn and Garden
Cell Phones and Accessories	Pet Supplies
Clothing, Shoes and Jewelry	Sports and Outdoors
Digital Music	Tools and Home Improvement
Electronics	Toys and Games
Grocery and Gourmet Foods	Video Games

DATA MASSAGING

Abstraction:

- Function to GUnzip a File
- Function to return a Dataframe from a JSON - Gzip file
- Function to Calculate Character Count in the ReviewText
- Function to Calculate Word Count in the ReviewText

Data Munging:

- Reading the Reviews into DataFrames
- Adding the Product Type Column to each DataFrame
- Adding individual DFs to AmazonReviews DataFrame List
- Labelling: Add Column Names to Ratings Dataframe since it is missing the Header
- Reading the Ratings into DataFrames

Amazon's Customer Building Blocks –Reviews & Ratings

- Adding the Product Type Column to each DataFrame
- Adding individual DFs to AmazonRatingsDataFrame List
- Use Value_counts() to understand how many columns are of type object
- Printing Reviews and Ratings Dataframes
- Dataframe df_AzReviews[23] and df_AzRatings[23] Before dropna
- Dataframe df_AzReviews[23] and df_AzRatings[23] After dropna - Shows no change for Ratings row numbers
- Dataframe df_AzReviews - running the dropna method on the list of DFs
- Dataframe df_AzRatings - running the dropna method on the list of DFs
- Printing Reviews & Rating Dataframe Heads to check data integrity
- Printing Reviews & Rating Dataframe Tails to check data integrity
- Indexing: Setting the index for both dataFrame Lists i.e. Reviews and Ratings as 'reviewerID'

Sampling:

- For both datasets of Reviews and Ratings we would like to now take only a sample set of the entire DataSet. I have decided on a fraction= 0.5 i.e. 5 % of each Product category a nd appending them into one dataframe of Sample_AzReviews and Sample_AzRatings to give a total count of 9020370 and 40368768 records respectively.

Feature Engineering:

- Adding reviewertext features such as character_count and word_count to the Reviews dataframe list
- Adding helpful column features such as helpful_numerator, helpful_denominator and helpful_percentage to the Reviews dataframe list
- Adding time series features such as reviewYear, reviewMonth, reviewDate, reviewDayofWeek and ReviewWeekofYear to the Reviews and Ratings DataFrame lists.

DATA VISUALIZATION OF ENTIRE DATASET

Type of Plot	Dependent Variable	Independent Variable	DataSet/Product Category	# of Observations	Observation
Scatter SubPlots	OverAll	WordCount_ReviewText	Sample_AzReviews Dataset: Video Games	Head(100)	Rating 4 has higher wordcount. Rating 5 has the maximum records concentration in the 0 - 100 range wordcount.
Scatter SubPlots	Overall	Helpful_Denominator	Sample_AzReviews Dataset: Video Games	Head(100)	Rating 1 has the outlier with the maximum helpful_denominator i.e. helpful Factor at 54.
Scatter SubPlots - Tight Layout	OverAll	WordCount_ReviewText	Sample_AzReviews Dataset:	Head(500) of each set.	The Baby Product Category got the most Ratings of 1 and 5 i.e.

Amazon's Customer Building Blocks –Reviews & Ratings

			1. Instant Video 2. Android Apps 3. Automotive 4. Baby		wordier reviews up to 700 wordcount.
Swarmplot	ReviewDate	Overall	Sample_AzReviews Dataset	Head(500)	More Reviews of Rating 5 were given in total, next maximum number of reviews given was towards Rating 4. More Reviews were Rated in the first 20 days of the month.
Type of Plot	**Dependent Variable**	**Independent Variable**	**DataSet/Product Category**	**# of Observations**	**Observation**
Swarmplot	ReviewDayofWeek	Overall	Sample_AzReviews Dataset	Head(500)	datetime.weekday()--Return the day of the week as an integer, where Monday is 0 and Sunday is 6. Overall Rating ->Popular Days 1 -> Tue, Fri 2-> Mon, Sun 3-> Tue,Thu, Sun 4-> All Days equal 5-> All Days equal
Swarmplot	ReviewMonth	Overall	Sample_AzReviews Dataset	Head(500)	Overall Rating ->Popular Months 1 -> Feb 2-> Feb, Apr, May (None in Dec) 3-> Feb, June (None in Nov) 4-> Least in Oct 5-> All Months equal
Swarmplot	ReviewYear	Overall	Sample_AzReviews Dataset	Head(500)	
Swarmplot	Wordcount_reviewtext	Overall	Sample_AzReviews Dataset	Head(500)	The Ratings 4 and 5 have wordier outliers in 1000+ wordcount range.

Amazon's Customer Building Blocks –Reviews & Ratings

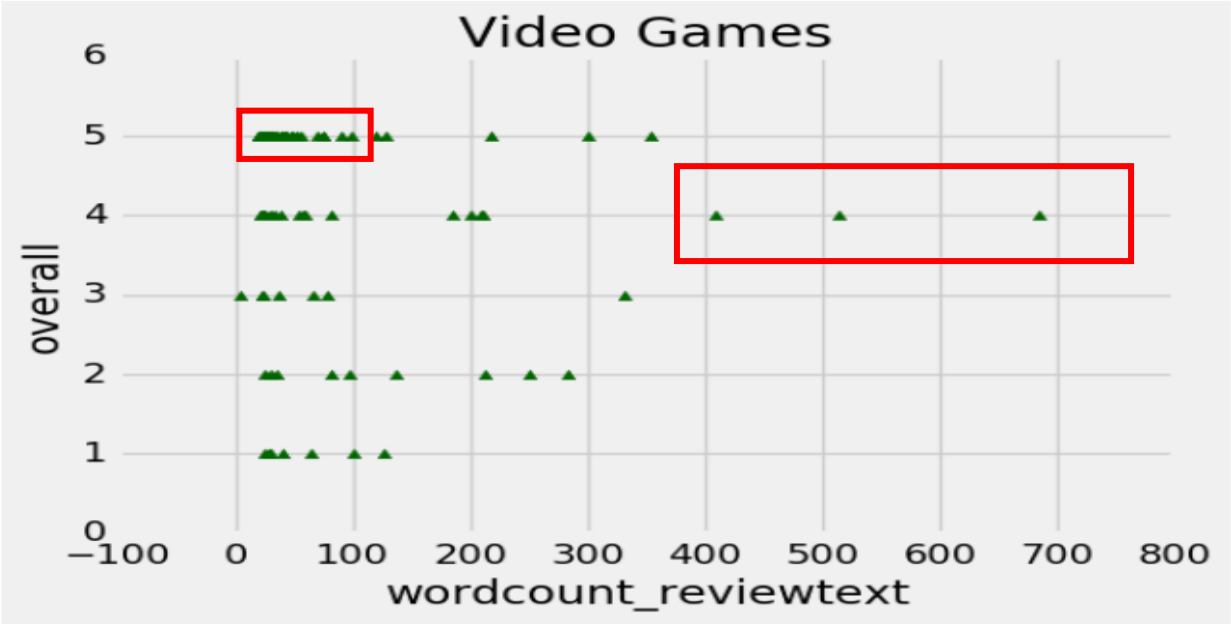
Seaborn Heatmap		Overall Helpful_numerator Helpful_denominator ReviewMonth ReviewDayofWeek	score_Reviews_data	Head(10)	
Seaborn Clustermap		Overall Helpful_numerator Helpful_denominator ReviewMonth ReviewDayofWeek	score_Reviews_data	Head(10)	
Seaborn Pairplot	Overall Charcount_reviewtext Helpful_numerator Helpful_denominator ReviewMonth ReviewDayofWeek	Overall Charcount_reviewtext Helpful_numerator Helpful_denominator ReviewMonth ReviewDayofWeek	score_Reviews_data	Head(500)	CharCount of the reviews shows that the Star Ratings of 2, 4 and 5 are outliers and gravitate towards the wordier reviews (10,000+ characters).
Seaborn Pairplot	Overall Charcount_reviewtext Helpful_numerator Helpful_denominator ReviewMonth ReviewDayofWeek	Overall Charcount_reviewtext Helpful_numerator Helpful_denominator ReviewMonth ReviewDayofWeek	score_Reviews_data	Tail(500)	CharCount of the reviews shows that the Star Ratings of 2, 4 and 5 are outliers and gravitate towards the wordier reviews (6,000-10,000 characters).

DATA VISUALIZATION BASED INITIAL FINDINGS

Typically one would guess that the WordCount of the reviews is the most or the least for Overall Star Ratings of 5 i.e. a Star Ratings when the best gets a 'lot of praise' or 'is expressed in a few words' – there are 2 extremes and this relates to human behavior directly of when being happy expresses oneself in either monosyllables or with a mouthful of words. The same goes for Overall Star Ratings of 1 which builds on the emotion of anger – one is either expressive in few words or the other extreme of being very wordy.

Question is then does Star Rating of 1 and 5 get the most and least word count association? Let's look at the visuals that tell us a story:

Amazon's Customer Building Blocks –Reviews & Ratings



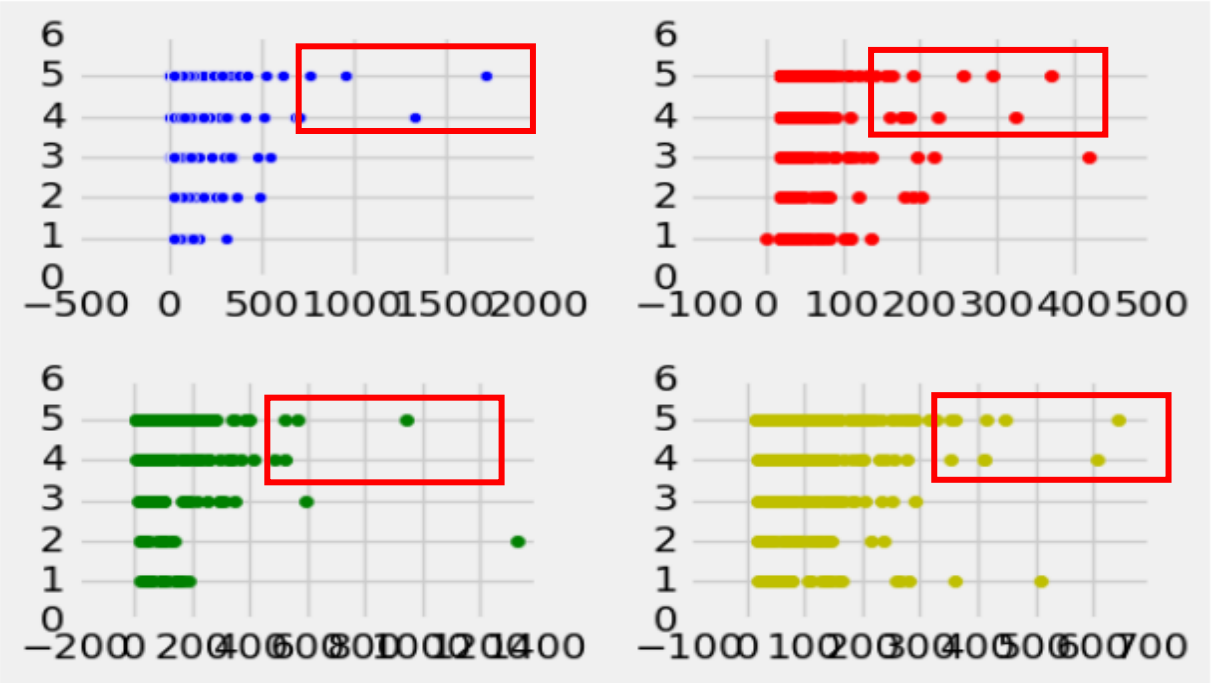
Observation: Rating 4 has higher wordcount. Rating 5 has the maximum records concentration in the 0 - 100 range wordcount. **The lower wordcounts are focused across ALL Overall Star Ratings (1-5).**

Instant Video

Android Apps

Amazon's Customer Building Blocks –Reviews & Ratings

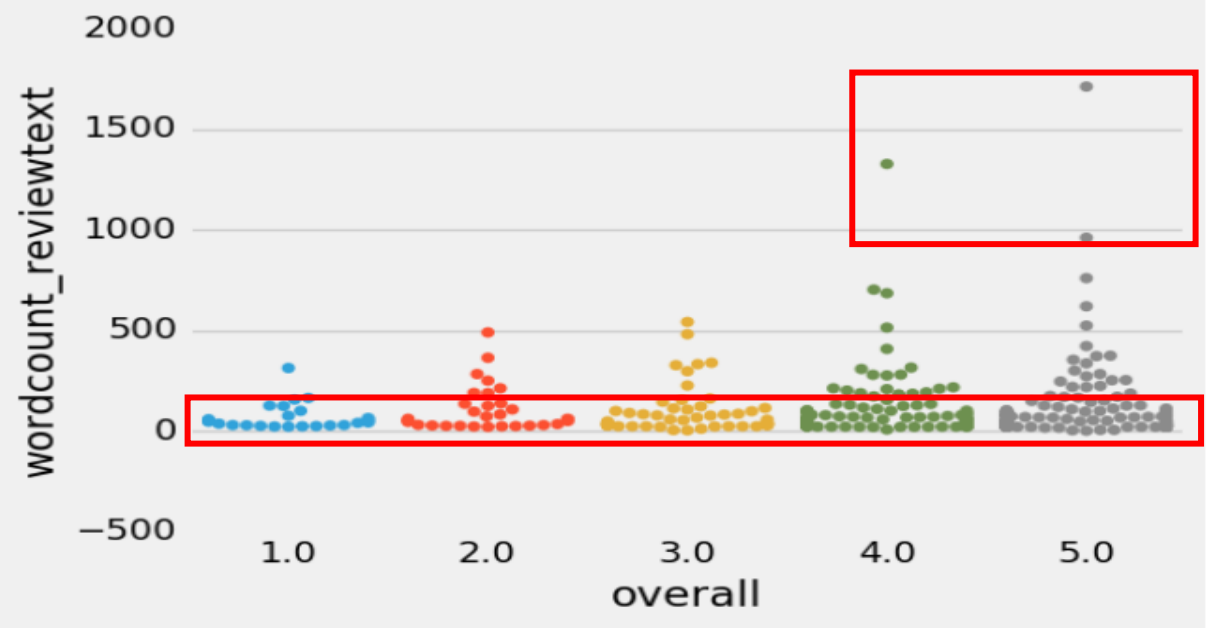
WordCount_reviewtext (X-axis) vs Overall Rating (Y-axis)



Automotive Baby

Observation: The Baby Product Category got the most Ratings of 1 and 5 i.e. wordier reviews up to 700 wordcount. All Product Types have the wordier reviews in the Star Rating 4 and 5 set. **The lower wordcounts are focused across ALL Overall Star Ratings (1-5) and Product Types.**

Amazon's Customer Building Blocks –Reviews & Ratings



Observation: The Ratings 4 and 5 have wordier outliers in 1000+ wordcount range. **The lower wordcounts are focused across ALL Overall Star Ratings (1-5).**

Amazon's Customer Building Blocks –Reviews & Ratings

Observation: Of all the correlations, the Overall Star Rating vs the Charcount_Reviewtext shows that there is a direct relationship between these features in the above pair-plot. CharCount of the reviews shows that the Star Ratings of 2, 4 and 5 are outliers and gravitate towards the wordier reviews (6,000-10,000 characters). **The lower wordcounts are focused across ALL Overall Star Ratings (1-5).**

Finally, no matter which visual tool is used, or dataset is chosen, we see this across the board that the **lower wordcounts are focused across ALL Overall Star Ratings (1-5).** We had predicted that the 1 and 5 overall star ratings would see a low and high word count, from the visuals see the prediction come true only partially and there is a slight twist to it too.

At this point we will try and answer some of the original questions:

Q1. Does the wordcount in the review text matter? Do more words mean higher overall Star Ratings?

A1. Visually we see that the wordcount for each customer review does not matter because a lower wordcount is a trend seen across all Overall Star Ratings given, whether the customer gave a 1, 2, 3, 4 or

Amazon's Customer Building Blocks –Reviews & Ratings

5. Visually higher counts for the text of the worded reviews is visually evident as being present for only the higher Overall Star Ratings (4 or 5). Statistically we see that the averages of the wordcounts seem to even out and the one-way ANOVA and Tukey's tests both certify that the means of wordcounts are equal no matter which Overall Rating number is selected.

Q2. Do all Overall Star Ratings (1-5) have the same behavior across Product Categories i.e. Do Amazon Customers give more 1 stars to Baby products or more 5 stars to Android Apps?

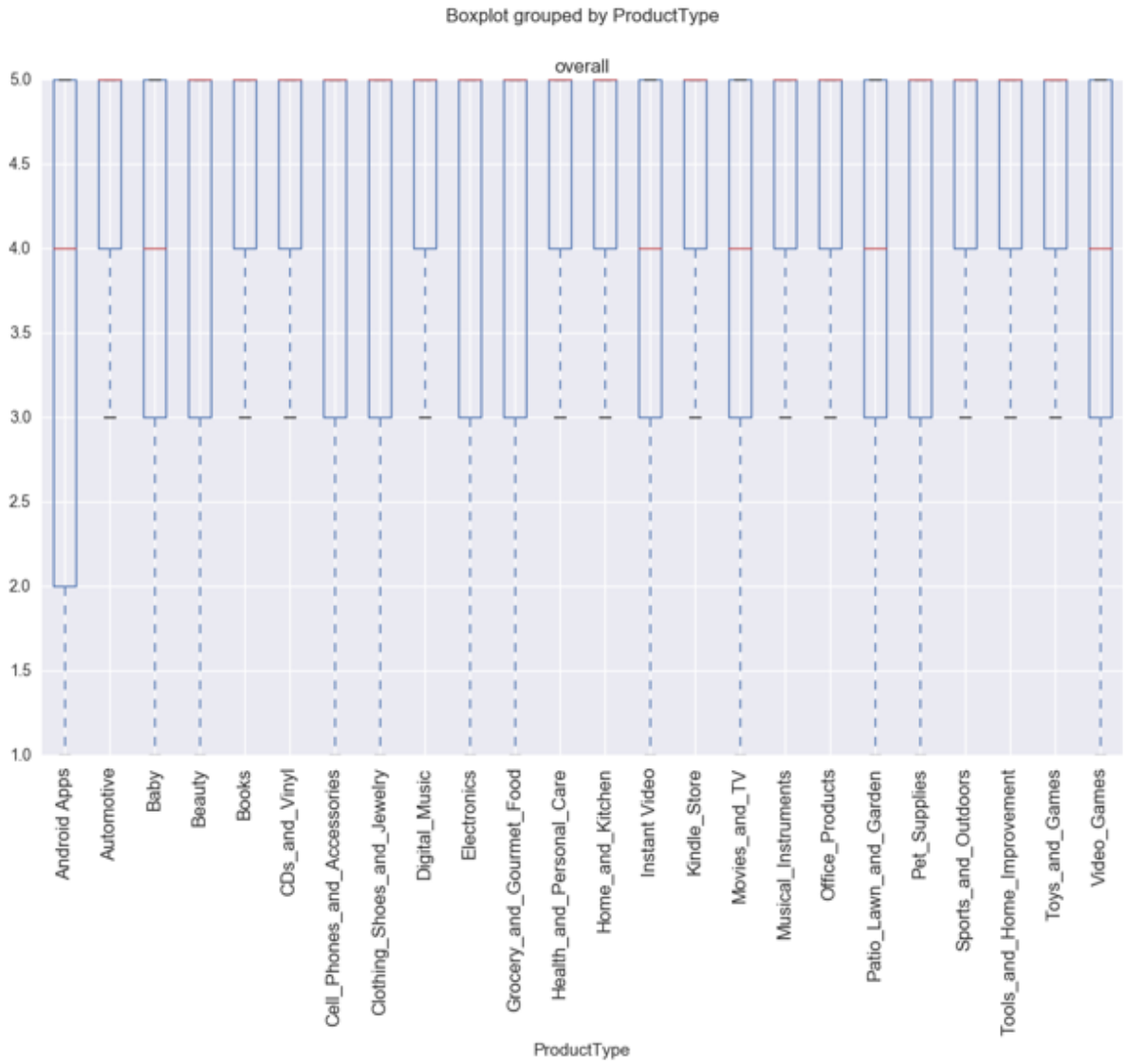
A2. The averages of Overall Star Ratings across all Product categories (as seen from the box plot below) are different. For e.g. Android Apps and Baby products have lower average Overall Star Ratings compared to Automotive, Books and Digital Music. The ANOVA one-way test proves this statistically.

INFERENTIAL STATISTICS

The first action is to create the score_Reviews_data as the dataframe that provides all numerical data that will be used for further statistical analysis and ML algorithms. It has the following features:

- Overall – Star Rating from 1 through 5
- Asin - Unique Product Code
- CharacterCount_ReviewText – Number of characters in Customer Review Text
- WordCount_ReviewText – Number of words in Customer Review Text
- Helpful_Numerator – Count of Helpful (ThumsUp) given to a Customer Review
- Helpful_Denominator – Count of total Customer Reviews given
- HelpfulPercentage – Percentage calculated from Numerator and Denominator numbers.
- ReviewYear – Year when the customer review was given.
- ReviewMonth - Month when the customer review was given.
- ReviewDate - Date when the customer review was given.
- ReviewDayofWeek – Day of the week when the customer review was given.
- ReviewWeekofYear – Week of the year when the customer review was given.

Amazon's Customer Building Blocks –Reviews & Ratings




```
k= 24             (number of conditions)
N= 4681373        (conditions times participants)
n= 139608         (Participants in each condition)
```

The above boxplot shows that the means of the 'Overall Star Ratings' across all Product Types is diff erent. Running the one-way ANOVA test and the Tukey test will either confirm or refute the findings.

The one-way ANOVA tests whether the mean of some numeric variable differs across the levels of one categorical variable. It essentially answers the question: do any of the group means differ from one another? The ANOVA test involves more calculations than the t-test, but the process is similar: you go through several calculations to arrive at a test statistic and then you compare the test statistic to a critical value based on a probability distribution. In the case of the ANOVA, you use the "f-distribution".

Amazon's Customer Building Blocks –Reviews & Ratings

The SciPy library has a function for carrying out one-way ANOVA tests called scipy.stats.f_oneway(). Let's use the ANOVA to compare average Overall Ratings across the Product groups and Wordcount groups.

- Using 1 way ANOVA using statmodel where we will compare grouped reviews based on producttype i.e. Instant Videos, Video Games, Android Apps, etc.
- Using 1 way ANOVA using statmodel where we will compare grouped reviews based on wordcount i.e. [0,443,639,803,1284] etc.

(1) Hypothesis for the set of test data of **overall Star Ratings** against the **Product Type** of the reviewText:

H_0: The average number of overall Star Ratings given for customer Reviews for each Product Type is equal.
H_a: The average number of overall Star Ratings given for customer Reviews for each Product Type is not equal.

AND

(2) Hypothesis for the other set of test data of **overall Star Ratings** against the **wordcount** of the reviewText:

H_0: The average number of overall Star Ratings given for customer Reviews for each word_count category is equal.
H_a: The average number of overall Star Ratings given for customer Reviews for each word_count category is not equal.

Results [per Product Type]:

- `F_onewayResult(statistic=40.728777413441748, pvalue=1.754135462561985e-10)`
- After running the ANOVA test, the F-statistic is `40.728777413441748` and the p-value is `1.754135462561985e-10`. Since the p-value is near zero we can reject the null hypothesis and conclude that the average number of overall Star Ratings given for customer Reviews for each product category is not equal.

	df	sum_sq	mean_sq	F	PR(>F)
ProductType	1.0	0.183824	0.183824	0.079064	0.782412
Residual	15.0	34.875000	2.325000	NaN	NaN

Results [per Wordcount Category i.e. [0,443,639,803,1284]]:

- `F_onewayResult(statistic=1.00672268907563, pvalue=0.44181603711228229)`

After running the ANOVA test, the F-statistic is `1.00672268907563` and the p-value is `0.44181603711228229`. Since the p-value is not near zero we cannot reject the null hypothesis and **conclude that the average number of overall Star Ratings given for customer Reviews for each word_count category is equal.**

- The test result suggests the groups don't have the same sample means in this case, since the p-value is significant at a 99% confidence level. To check which groups differ after getting a positive ANOVA result, you can perform a follow up test or "post-hoc test". One post-hoc test is to perform a separate t-test for each pair of groups. You can perform a t-test between all pairs by running each pair through the stats.ttest_ind(). The table below shows since the p-values are not below 0.05 we cannot reject the null hypothesis and **conclude that the pairs' mean values are equal.**

WordCount Pairs	**Ttest Results**
0 443	Ttest_indResult(statistic=-0.7844645405527361 8, pvalue=0.46260543015431632)
0 639	Ttest_indResult(statistic=-1.1239029738980328 , pvalue=0.30399938256846926)
0 803	Ttest_indResult(statistic=0.57173115475474245 , pvalue=0.59224601604340843)
443 639	Ttest_indResult(statistic=0.0, pvalue=1.0)
443 803	Ttest_indResult(statistic=1.2518642814237004, pvalue=0.26599088513252928)
639 803	Ttest_indResult(statistic=1.9720265943665387, pvalue=0.10564980084735728)

Tukey's range test, named after the American mathematician John Tukey, is a common method used as a post hoc analysis after the one-way ANOVA. This test compares all possible pairs and we can use it to precisely identify pairs where the difference between two means is greater than the expected standard error.

For each pair of mean values:
H_0: The means are equal.

H_a: The means are not equal.

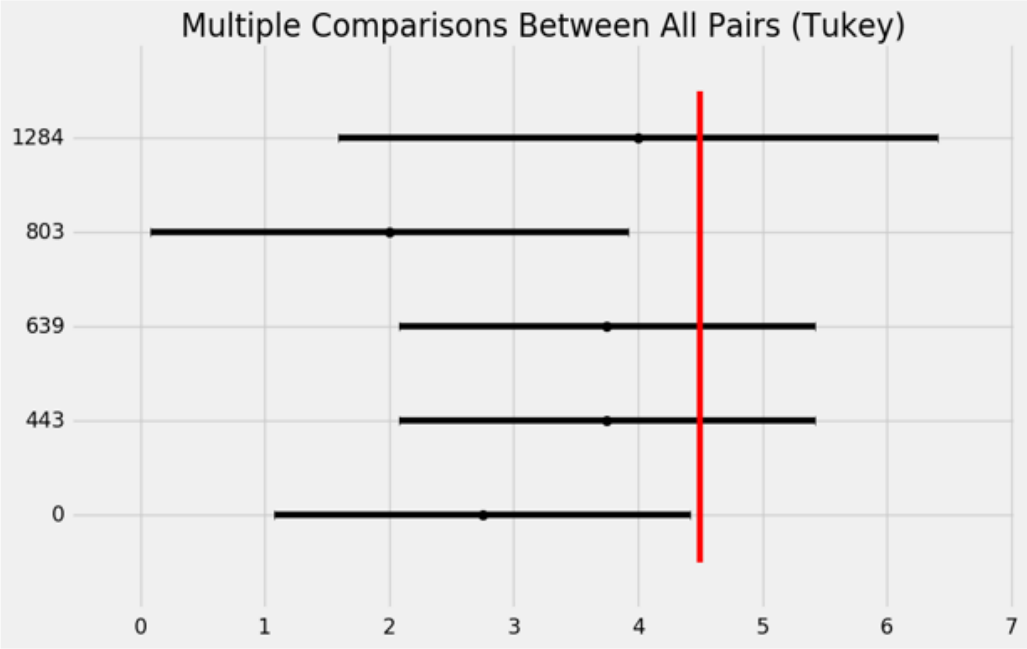
The result below shows each pairs' mean difference. Since the pair's mean values are not statistically significantly different, then we cannot reject the null hypothesis and **conclude that the pairs' mean values are equal**. In the table below, the 'reject' column has a False value.

	df	sum_sq	mean_sq	F	PR(>F)
wordcount_reviewtext	1.0	0.577909	0.577909	0.251404	0.623367
Residual	15.0	34.480915	2.298728	NaN	NaN

Amazon's Customer Building Blocks –Reviews & Ratings

Multiple Comparison of Means - Tukey HSD,FWER=0.05

group1	group2	meandiff	lower	upper	reject
0	443	1.0	-2.3342	4.3342	False
0	639	1.0	-2.3342	4.3342	False
0	803	-0.75	-4.3513	2.8513	False
0	1284	1.25	-2.8335	5.3335	False
443	639	0.0	-3.3342	3.3342	False
443	803	-1.75	-5.3513	1.8513	False
443	1284	0.25	-3.8335	4.3335	False
639	803	-1.75	-5.3513	1.8513	False
639	1284	0.25	-3.8335	4.3335	False
803	1284	2.0	-2.3044	6.3044	False



MACHINE LEARNING MODELS AND PREDICTIVE ANALYSIS

- KNN CLUSTERING MODEL
- LINEAR REGRESSION MODEL
- NAÏVE BAYES MODEL

NLTK and NLP PROCESSING OF REVIEWTEXT DATA