Capstone Project-2 Milestone Report -1

Topic: Mining Amazon Product Reviews using NLP

1. **Introduction:**
   Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same product's availability and pricing at different e-retailers. But online shopping comes with its own limitations. One of the biggest challenges is verifying the authenticity of a product. Is it as good as advertised on the e-commerce site? Will the product last more than a year? Are the reviews given by other customers true or are they false advertising? These are important questions customers need to ask before spending their money.

2. **Problem Statement:**
   Ratings alone do not give a complete picture of the products we wish to purchase. So secondary option is looking at the reviews. Review's plays an important role in the decision-making process. If the number of reviews is less, it is easy to read and understand but what if there are thousands of reviews. So, the problem is How we can analyze great number of online reviews using Natural Language Processing (NLP).

   Client: Amazon customers

This project will serve three purposes

- Helps consumers to understand the sentiment of the review.
- Help the consumers to get consumer feedback in the form of topics covered by the reviews without having to go through all of them.
- Enable consumers to quickly extract the summary of the reviews without reading the entirely.

3. Data Description:

Dataset: https://www.kaggle.com/snap/amazon-fine-food-reviews

Context: This dataset consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all ~500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories.

```
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
#   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
0   Id                      568454 non-null int64
1   ProductId               568454 non-null object
2   UserId                  568454 non-null object
3   ProfileName             568438 non-null object
4   HelpfulnessNumerator    568454 non-null int64
5   HelpfulnessDenominator   568454 non-null int64
6   Score                   568454 non-null int64
7   Time                    568454 non-null int64
8   Summary                 568427 non-null object
9   Text                    568454 non-null object
```

For the analysis I have used only three features (Score, Summary, Text)

4. Text Wrangling and Pre-processing:

There are usually multiple steps involved in cleaning and pre-processing textual data. I have done some of the most important steps which are used heavily in Natural Language Processing (NLP) pipelines.

Step1: Removing HTML Tags

Unstructured text contains a lot of noise, HTML tags are typically one of these components which do not add much value towards understanding and analyzing text. So, in the first step removed unnecessary HTML tags and retain the useful textual information from the review.

Step:2 Expanding Contractions

Contractions are the shortened versions of words like don't for do not and how'll for how will. These are used to reduce the speaking and writing time of words. We need to expand these contractions for a better analysis of the reviews. I have created a dictionary of common English contractions that I will use for mapping the contractions to their expanded forms. Using a function called *'remove_contractions'* I have removed all the contractions in the review.

Step:3 Removing Special Characters

Special characters and symbols are usually non-alphanumeric characters or even occasionally numeric characters (depending on the problem), which add to the extra noise in unstructured text. Usually, simple regular expressions (regexes) can be used to remove them.
I have kept removing digits as optional, because often we might need to keep them in the pre-processed text.

Step4: Lemmatization

Lemmatisation is the algorithmic process of determining the lemma of a word based on its intended meaning. Remove word affixes to get to the base form of a word. However, the base form in this case is known as the root word, also known as the ***lemma***, will always be present in the dictionary. WordNetLemmatizer() is used for lemmatization process.

Step5: Removing Stop words

Words which have little or no significance, especially when constructing meaningful features from text, are known as stopwords or stop words. These are usually words that end up having the maximum frequency if you do a simple term or word frequency in a corpus. Typically, these can be articles, conjunctions, prepositions and so on. Some examples of stopwords are *a*, *an*, *the*, *and* the like. used a standard English language stopwords list from **nltk.**

5. Exploratory Data Analysis (EDA)

We have already cleaned our data and have our corpus ready, yes – it is finally time for Exploratory Data Analysis! It is a crucial part of any data science project because that is where you get to know more about the data. In this phase, we can reveal hidden patterns in the data and generate insights from it.

Analysis of 'Score' has been done using "sns.countplot()" method, to understand the count of each score.

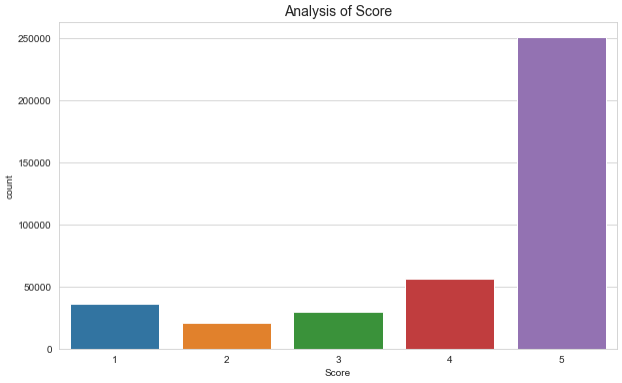

Fig:1

From the graph we can conclude that most of the reviews have score 5 and very a smaller number of reviews got score 2.

Word count distribution plot.

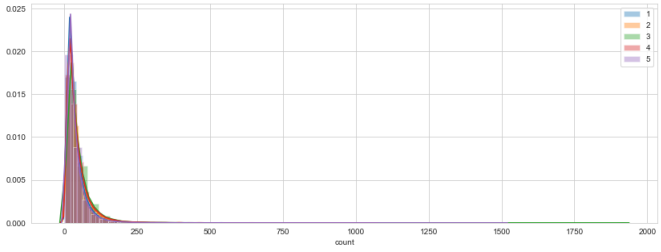

Fig:2

The word count distribution plot tells us that the length of all most all reviews are within the range of 0-250, but there are few outliers.

6. Sentiment Analysis

Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether the review is positive or negative or neutral. Sentiment Analysis is done using TextBlob (*TextBlob* is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.)

The sentiment function of textblob returns two properties, polarity, and subjectivity. Polarity is float which lies in the range of [-1.0,1.0] where 1 means positive statement and -1 means a negative statement. Subjective sentences generally refer to opinion, emotion, or judgment whereas objective refers to information. Subjectivity is also a float which lies in the range of [0,1].

I have classified the reviews into three, Positive, Negative and Neutral based on the polarity.

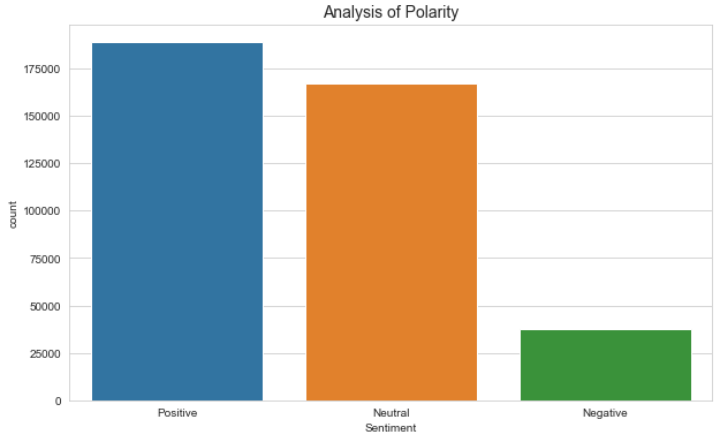

Fig:3

The above figure shows the distribution of polarity among reviews. Most of the reviews are classified as positive and a very less reviews classified as Negative.

WordCloud

A WordCloud is a visual representation of text data, typically used to depict keyword metadata (tags) on websites, or to visualize free form text. Tags are usually single words, and the importance of each tag is shown with font size or color. This format is useful for quickly perceiving the most prominent terms to determine its relative prominence.

Fig:4 wordcloud for Positive and Negative reviews

The above figure shows that the words ‘highly recommend’, ’well’,’ taste great’, ‘delicious’ are the common words used to represent positive review and the words ‘think’, ‘product’, ’use’ are used to represent negative review.

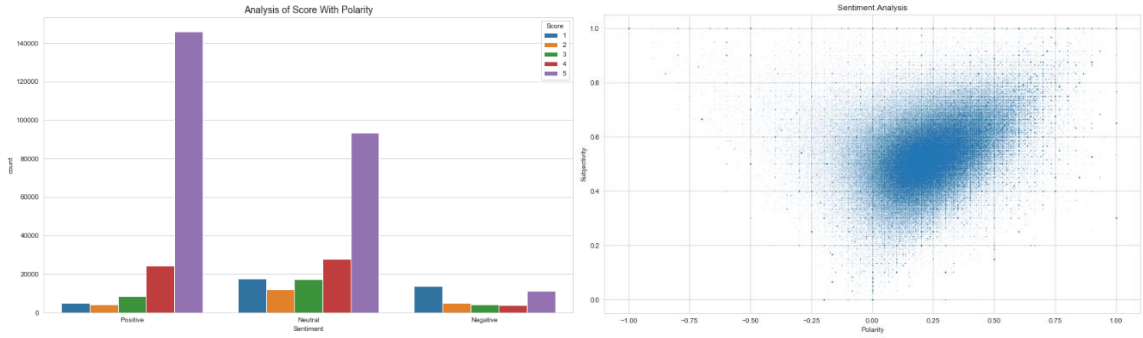

Fig:5

The Sentiment VS Score bar chart shows that, most of the positive reviews got score 5, but there are few positive reviews, which got very less Score. On the other side some of the negative reviews got high rating.

The Polarity VS Subjectivity scatter plot tells us that, most of the people are happy with Amazon food services.

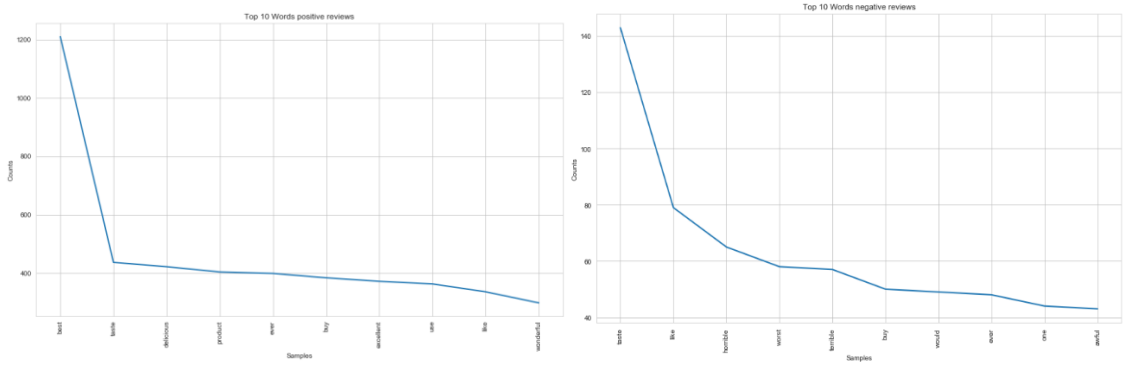

Fig:6

Top 10 Words for positive reviews are [('best', 1211),('taste', 437),('delicious', 422),('product', 404),('ever’, 399),('buy', 384),('excellent', 372),('use'363),('like', 336),('wonderful', 298)] and the Top 10 Words for negative reviews are [('taste', 143),('like', 79),('horrible', 65),('worst', 58),('terrible', 57),('buy', 50),('would', 49),('ever', 48),('one', 44),('awful', 43)].