Managing Big Data Final Project
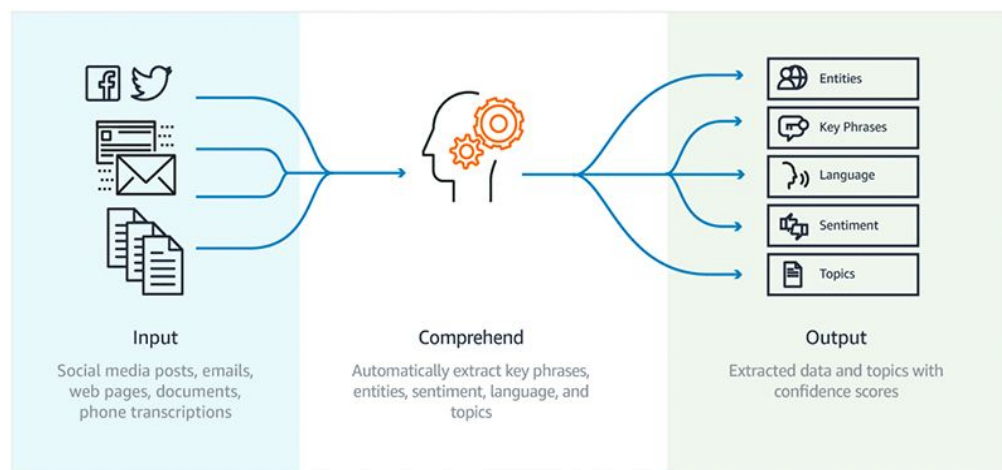
Team 4(04): Frank Fan, Carl Xi, Yaping Zhang, Jie Zhu

1. **Introduction to Amazon Comprehend**
   a. What is Amazon Comprehend:

   Amazon Comprehend utilizes NLP(natural language processing) to draw insights from the content of text documents. Amazon Comprehend is able to process any text file in UTF-8 format. Because Amazon Comprehend
   uses a pre-trained model to examine and analyze a document or set of documents to gather insights about it. This model is continuously trained on a large body of text so that there is no need for us to provide training data.

   b. How does it work:

Input	Comprehend	Output
Social media posts, emails, web pages, documents, phone transcriptions	Automatically extract key phrases, entities, sentiment, language, and topics	Extracted data and topics with confidence scores

   Amazon Comprehend takes social media posts, emails, web pages, phone transcriptions or any other documents you want to analyze, and automatically develops insights by extracting key phrases, identifying languages and recognizing entities and sentiments.

   c. Key Features:
   - Detecting the Dominant Language
     Amazon Comprehend identifies the dominant language in a document. Amazon Comprehend can identify over 100 languages.

   - Detecting Named Entities
     Amazon Comprehend returns a list of entities, such as people, places, and locations, identified in a document.

   - Keyphrase Extraction
     Amazon Comprehend extracts key phrases that appear in a document. For example, a document about a basketball game might return the names of the teams, the name of the venue, and the final score.
   - Sentiment Analysis
     Amazon Comprehend determines the emotional sentiment of a document. Sentiment can be

positive, neutral, negative, or mixed.

- Customized Classification
  Apart from the features above, Amazon Comprehend also allows us to create a customized text classification model. For example, This function enables us to automatically categorize input inquiry data by the problem type based on how the customer described the problem. It is very simple to create a custom model: we just need to provide documents for each of the labels we want to use, and Amazon Comprehend will train on these labels and documents to set the classifier.

2. **Amazon Comprehend Workshop**

We would love to share this part with our MSBA cohorts, who plan to use NLP for their capstone project.

a. Set up the environment:

After experimenting with different approaches to use Amazon Comprehend, we found the easiest way is to use Amazon Comprehend using a Jupyter Notebook based on Amazon SageMaker. The advantage is that we get to use Amazon Comprehend in a Python 3 environment that we are familiar with.

- Step 1: Set up your Amazon SageMaker notebook

  From the AWS Management Console, choose Services and then Amazon SageMaker under Machine Learning, and in the Amazon SageMaker console, under Notebook, choose Notebook instances. Now choose the Create Notebook Instance.
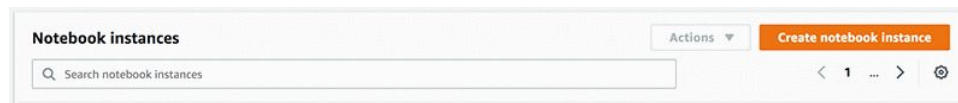
In the settings, we can specify the size based on your needs. But we need to create an IAM role that gives access to any necessary Amazon S3 buckets.
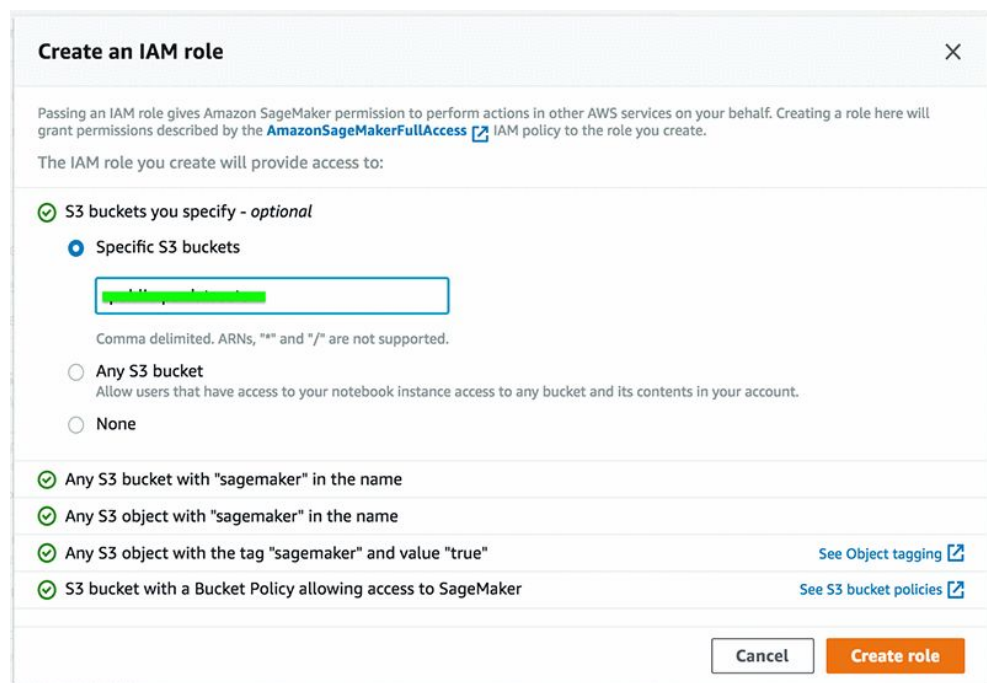
After successfully creating an IAM role, just click on Create notebook instance. After a few minutes, the notebook instance will be ready to use.

- Step 2: Attach Policy
  In the notebook settings, we need to choose IAM role ARN to open the IAM role attached to the notebook instance. After clicking on Attach policies, we just need to search for ComprehendFullAccess to attach it.

- Step 3: Create a notebook
  After successfully launching the notebook instance, we should choose new and choose conda_python3 to create a notebook in Python 3 environment.
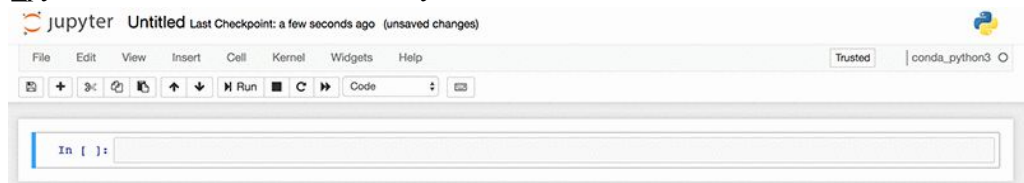
- Step 4: Connect to Amazon Comprehend
  Then we can use the AWS SDK for Python SDK (Boto3) to connect to Amazon Comprehend from your Python code base. Boto is the Amazon Web Services (AWS) SDK for Python. It enables Python developers to create, configure, and manage AWS services, such as S3 and Amazon Comprehend. Using the following command, we import boto3 and connect to Amazon Comprehend in a specified AWS Region using the boto3 client.

```python
import boto3

session = boto3.Session(region_name='us-east-1')
client = session.client('comprehend')
```

  Remember, we need to specify the AWS region in order to connect to Amazon Comprehend. Now we are ready to go.

b. Detect the dominant language:

Amazon Comprehend is able to identify text written in over 100 languages and returns the dominant language with a confidence score to support that a language is dominant. By calling the detect_dominant_language function on the text you want to analyze, Amazon comprehend will output the dominant language and the corresponding confidence score.

3

```
import json

session = boto3.Session(region_name='us-east-1')
client = session.client('comprehend')
text = "It is raining today in Seattle"

print('Calling DetectDominantLanguage')
print(json.dumps(client.detect_dominant_language(Text = text), sort_keys=True, indent=4))
print("End of DetectDominantLanguage\n")
```

```
Calling DetectDominantLanguage
{
    "Languages": [
        {
            "LanguageCode": "en",
            "Score": 0.9963054656982422
        }
    ],
    "ResponseMetadata": {
        "HTTPHeaders": {
            "content-length": "64",
            "content-type": "application/x-amz-json-1.1",
            "date": "Mon, 17 Feb 2020 04:59:52 GMT",
            "x-amzn-requestid": "32e1e074-1e5b-4ee0-9dfa-2505c403adb4"
        },
        "HTTPStatusCode": 200,
        "RequestId": "32e1e074-1e5b-4ee0-9dfa-2505c403adb4",
        "RetryAttempts": 0
    }
}
End of DetectDominantLanguage
```

c. Entity Recognition:

By calling the Amazon Comprehend detect_entities function, it will return the named entities ("People," "Places," "Locations," etc.) that are automatically categorized based on the provided text, as well as a confidence score and count.

```
import boto3
import json

session = boto3.Session(region_name='us-east-1')
client = session.client('comprehend')
text = "I'm an avid photographer, and I'm primarily found shooting with my DSLR \
or my instant film camera that I carry around for casual use. While nothing beats \
my DSLR in power and convenience, there's something magical about my instant film \
camera. Perhaps it's that you're shooting on actual film, or maybe it's that every \
shot you take is a unique physical artifact (which is special in today's world of \
Instagram and Facebook, where photos are a dime a dozen). All I know for sure is \
that they are incredibly fun to use and peoples' eyes light up when you pull one of these out at a party."

print('Calling DetectEntities')
print(json.dumps(client.detect_entities(Text=text, LanguageCode='en'), sort_keys=True, indent=4))
print('End of DetectEntities\n')
```

Result Table:

Entity	Category	Count	Confidence
Amazon.com, Inc.	Organization	1	0.96
Seattle, WA	Location	1	0.96
July 5th, 1994	Date	1	0.99
Jeff Bezos	Person	1	0.99
Seattle	Location	2	0.98
Portland	Location	1	0.99
Vancouver, BC	Location	1	0.97
Starbucks	Organization	1	0.91
Boeing	Organization	1	0.99

d. Key Phrase Extraction:

By calling the Amazon Comprehend detect_entities function, it will return the key phrases or talking points and a confidence score to support that this is a key phrase.

```
import boto3
import json

session = boto3.Session(region_name='us-east-1')
client = session.client('comprehend')
text = "I'm an avid photographer, and I'm primarily found shooting with my DSLR \
or my instant film camera that I carry around for casual use. While nothing beats \
my DSLR in power and convenience, there's something magical about my instant film \
camera. Perhaps it's that you're shooting on actual film, or maybe it's that every \
shot you take is a unique physical artifact (which is special in today's world of \
Instagram and Facebook, where photos are a dime a dozen). All I know for sure is \
that they are incredibly fun to use and peoples' eyes light up when you pull one of these out at a party."

print('Calling DetectEntities')
print(json.dumps(client.detect_entities(Text=text, LanguageCode='en'), sort_keys=True, indent=4))
print('End of DetectEntities\n')
```

Keyphrase	Count	Confidence
an avid photographer	1	0.99
my DSLR	2	0.97
my instant film camera	2	0.99
casual use	1	0.99
power and convenience	1	0.94
actual film	1	0.99
every shot	1	0.92
a unique physical artifact	1	0.99
today	1	0.91
world	1	0.99
Instagram and Facebook	1	0.99

e. Sentiment Analysis:

And last but not least, Amazon Comprehend is able to detect the sentiment of our input text. While using, we need to specify the language that we want to analyze. And the output will be the sentiment (positive, negative, mixed and neutral) and its confidence score.

```
# Try some examples
sentiment = client.detect_sentiment(
    Text="Works awesome for apt size 110 dryer - Works awesome for apt \
    size 110 dryer. Handles load from apt size washer just fine. It does take \
    longer to dry. Electric cost savings over a full size 220 is worth the time. \
    Does not add much humidity unless lint filter is full.",
    LanguageCode='en'
)
sentiment['Sentiment'], sentiment['SentimentScore']
```

```
('POSITIVE',
 {'Positive': 0.9983564019203186,
  'Negative': 3.536563235684298e-05,
  'Neutral': 0.0015746206045150757,
  'Mixed': 3.36409175361041e-05})
```

```
# Try some examples
sentiment = client.detect_sentiment(
    Text="才刚买的，用了两天就坏了，说换货一直没换，这么大的店，早干嘛呢！",
    LanguageCode='zh'
)
sentiment['Sentiment'], sentiment['SentimentScore']
```

```
('NEGATIVE',
 {'Positive': 0.003523502266034484,
  'Negative': 0.9938187003135681,
  'Neutral': 0.00265671918168664,
  'Mixed': 1.0957356835206156e-06})
```

3. **Amazon Reviews Data**

We then used Amazon Customer Reviews as our dataset to try out Amazon Comprehend as a big data analytic tool. Amazon Customer Reviews (a.k.a. Product Reviews) is one of Amazon's iconic products. In a period of over two decades since the first review in 1995, millions of Amazon customers have contributed over a hundred million reviews to express opinions and describe their experiences regarding products on the Amazon.com website. Over 130+ million customer reviews are available to researchers as part of this dataset.

a. Features:

|	marketplace	customer_id	review_id	product_id	product_parent	product_title	product_category	star_rating	helpful_votes	total_votes	vine	verified_purchase	review_headline	review_body	review_date
0	US	15199106	R203HPW78Z7N4K	B0067WNSZY	633038551	FGGF3032MW Gallery Series 30" Wide Freestandin...	Major Appliances	5	0	0	N	Y	If you need a new stove, this is a winner.	What a great stove. What a wonderful replacem...	2015-08-31
1	US	16374080	R2EAIGVLEALSP3	B002QSXK60	811786871	Best Hand Clothes Wringer	Major Appliances	5	1	1	N	Y	Five Stars	worked great	2015-08-31
2	US	15322085	R1K1CD73HHLILA	B00EC452R8	345582728	Supco SET184 Thermal Cutoff Kit	Major Appliances	5	0	0	N	Y	Fast Shipping	Part exactly what I needed. Saved by purchasi...	2015-08-31
3	US	32004835	R2KZBMOFRMYOPO	B00MVVIF2G	563052763	Midea WHS-160RB1 Compact Single Reversible Doo...	Major Appliances	5	1	1	N	Y	Five Stars	Love my refrigerator! ! Keeps everything cold.	2015-08-31
4	US	25414497	R6BIZOZY6UD01	B00IY7BNUW	874236579	Avalon Bay Portable Ice Maker	Major Appliances	5	0	0	N	Y	Five Stars	No more running to the store for ice! Works p...	2015-08-31

Most of the features are intuitive. Among these, 'vine' means vine review. Basically, Amazon Vine invites the most trusted reviewers on Amazon to post opinions about new and pre-release items to help their fellow customers make informed purchase decisions. Amazon invites customers to become Vine Voices based on their reviewer rank, which is a reflection of the quality and helpfulness of their reviews as judged by other Amazon customers.

b. Data Cleaning:

We filtered out all reviews that are under 10 characters and over 5000 characters, the Amazon Comprehend max character length. And we only kept the purchases happening in the US market for convenience.

4. Sentiment Irregularities

With everything setup, we can now take a look at the average ratings by category and sentiment. Firstly, we compiled the review sentiments by star rating for the Major Appliances category, as shown below:
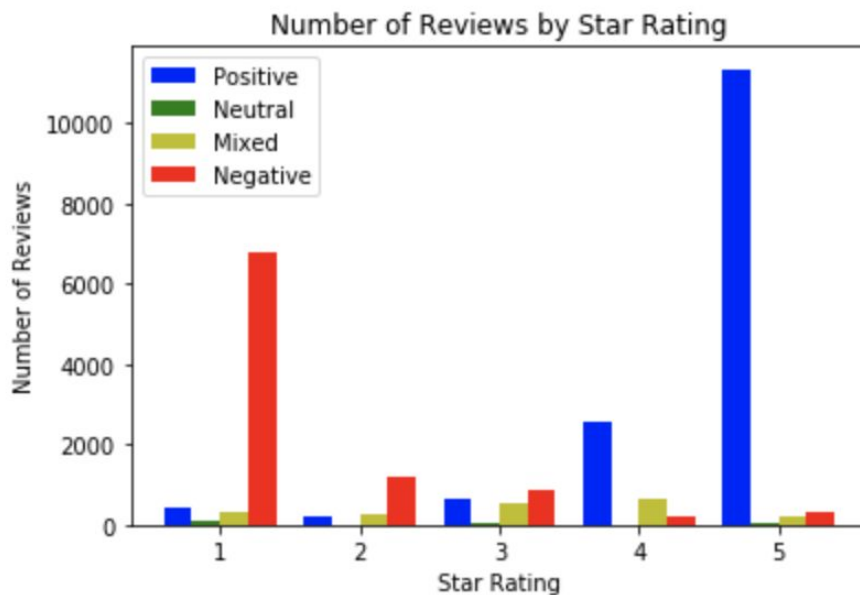
From visual inspection, we can immediately see that inflection point between positive and negative review occur at 3-star reviews, which makes sense. What doesn't make sense is that there are positive reviews with very low star ratings and negative reviews with very high star ratings. In addition, mixed reviews, which by definition should be around 2-4 in

ratings, are sporadically all over the place. If we break apart the above graph into histograms by each rating, this effect is seen even more clearly:
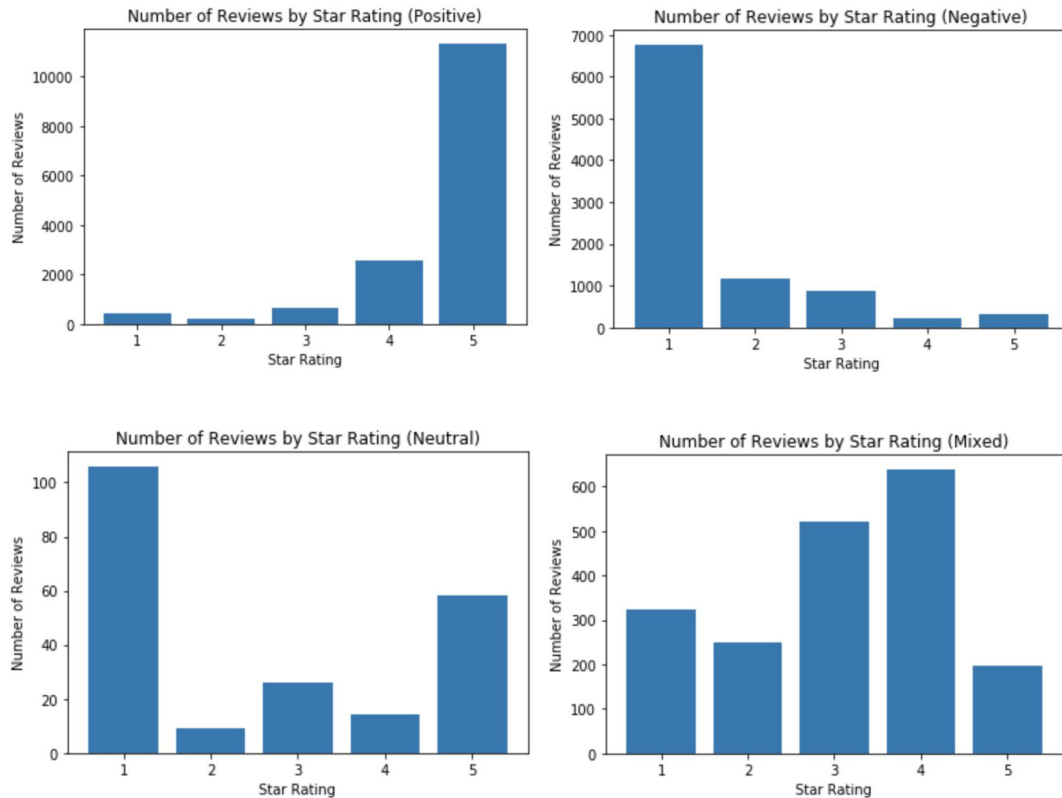
With the average star rating at 3.8853, it seems like the average scores of each sentiment fall within expected norms.

```
sentiment
MIXED      3.432326
NEGATIVE   1.575680
NEUTRAL    3.840580
POSITIVE   4.733060
```

Upon closer inspection, the reasoning behind illogical sentiment-rating pairings become evident. Reviews are written by humans, and humans make mistakes. Take this review below for an example. With an extremely positive comment, the one-star rating can only be explained by human error. In fact, it becomes apparent that this is the one factor that data has a hard time encapsulating.
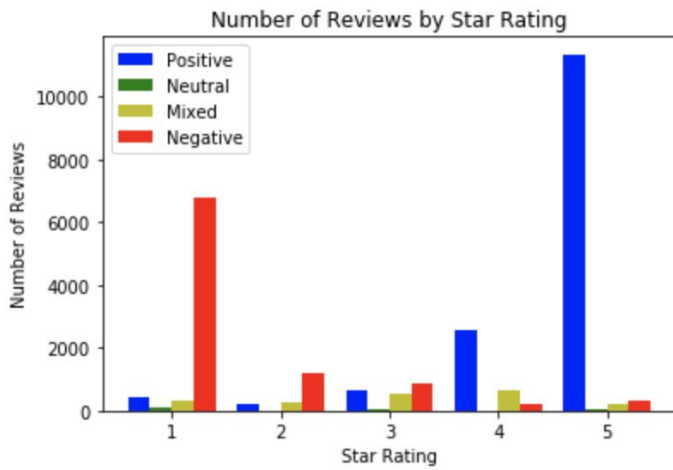
One Star	All these items I purchased are excellent.	2015-08-29	POSITIVE

As we will see when we examine another category, other factors also come into play that makes gives Amazon Comprehend a lot of space for improvement.

5. Cross-Category Analysis

Let's take a look at how ratings by sentiment differ by category:

Major Appliances :

Video Games:

Right off the bat, we can see that sentiment analysis for Video Games is completely screwed up. Positive review forms a concave parabola from 1-5 star ratings. Rating Expectations greatly vary depending on the product category. This could be due to various factors, such as:

- availability of choice (20,000 racing games vs 5 washing machines)
- ease of review (finding scratches on product vs visual glitch in-game)
- tolerance level (you can restart a game but not a product)

Here are some snippet from our presentation that further identify this problem. In summary, due to the change in digital products after release, reviews scores and sentiment scores are unreliable for digital products.

Average Star Rating Comparison:

3.8853 - Major Appliances

4.2039 - Video Games

Star Rating becomes unreliable for products that change over time:

Video Games:

sentiment	
MIXED	4.225060
NEGATIVE	4.204478
NEUTRAL	4.128000
POSITIVE	4.209306

Major Appliances:

sentiment	
MIXED	3.432326
NEGATIVE	1.575680
NEUTRAL	3.840580
POSITIVE	4.733060

Digital Release Items

Seller follows up following bad rating with remedy action, causing review/rating to alter. User may forget to change either/both.

Example:

Rating	Comment	Date	Sentiment
1	Charge last fit days with daily use of Xbox on...	2015-08-31	POSITIVE

Alternatively, some products (e.g. digital releases) can change over time.

Result: Unreliable ratings that require further analysis

Further Analysis

Cross-Category comparison will become very interesting when done at scale

- Reviews can be adjusted based per-category to better reflect reality
- Analyse review and sentiments by customer profiling (Jie)

We are constrained by our budget on this project:

Every ~10,000 lines costed roughly 6-7 USD for Amazon Comprehend to process

```
US REVIEWS DATASET:
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Wireless_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Watches_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Video_Games_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Video_DVD_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Video_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Toys_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Tools_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Sports_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Software_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Shoes_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Pet_Products_v1_00.tsv.gz
https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Personal_Care_Appliances_v1_00.tsv.gz
```

6. **User Analysis**
   a. **Data Used**

In this Amazon Reviews Database, there are 46 US reviews datasets, ranging from wireless products to apparels. In addition, there are 5 multilingual reviews datasets from Amazon's US, UK, Japan, France, and Germany websites. A complete list of these datasets can be found at https://s3.amazonaws.com/amazon-reviews-pds/tsv/index.txt.

For user analysis, we choose 4 US reviews datasets, including Major Appliances, Furniture, Watches, and Musical Instruments. After being concatenated, there are 2,752,353 rows in total. For the purpose of analyzing user behavior, we only keep customers who have more than 30 reviews in these product categories. This leaves us with 27,106 reviews and 569 unique customers.

b. Data Description

After dataset concatenation, we append sentiment results and sentiment statistics. Sentiment statistics consist of positive sentiment probability, negative sentiment probability, neutral sentiment probability, mixed sentiment probability. These statistics measure the probability of the review is one of these sentiment results and should add up to 1 for each review. We will focus primarily on star-rating and positive-probability in this analysis.

	customer_id	star_rating	prob_positive	prob_negative	prob_neutral	prob_mixed
count	2.710600e+04	27106.000000	27106.000000	27106.000000	27106.000000	27106.000000
mean	3.017988e+07	4.438538	0.809649	0.091011	0.033859	0.065471
std	1.551739e+07	1.001274	0.349027	0.256465	0.124053	0.223598
min	9.604500e+04	1.000000	0.000000	0.000000	0.000000	0.000000
25%	1.629649e+07	4.000000	0.870200	0.000000	0.000300	0.000000
50%	3.087312e+07	5.000000	0.996500	0.000200	0.001000	0.000000
75%	4.439716e+07	5.000000	0.999500	0.004200	0.008200	0.000100
max	5.309583e+07	5.000000	0.999900	0.999900	0.999800	0.999900

As is shown above, for these four product categories, the average star-rating is 4.44 stars, and the average probability of positive sentiment is 80.96%. The *star_rating* variable ranges from 1 to 5, while the *prob_positive* covers a variety of 0.00% to 99.99%. Most of the reviews have 5 stars and a probability of positive sentiment over 99%. Histograms for *star_rating* and *prob_positive* also confirm this finding.
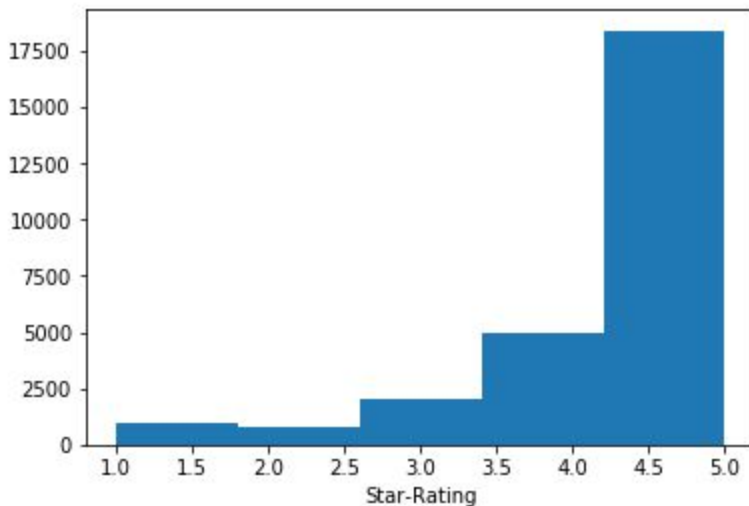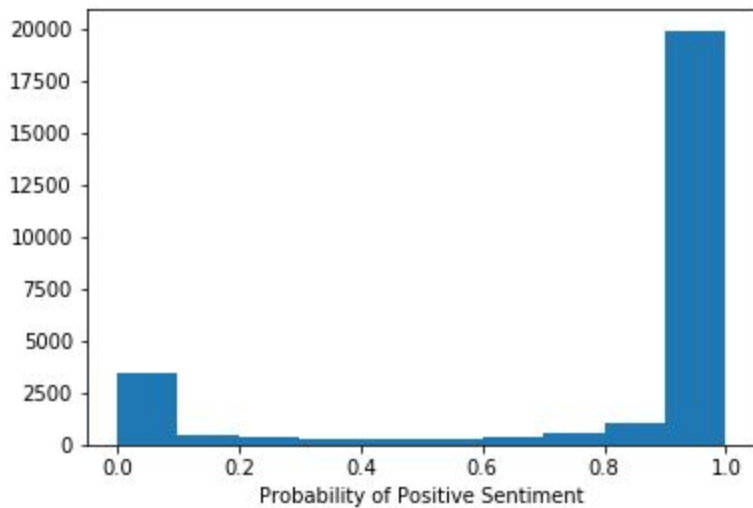
c. User Behaviors

After grouping data by *customer_id*, we have the following interesting findings.

i) Highest Star-Rating Givers

customer_id	star_rating	prob_positive
20631789	5.0	0.960306
37033189	5.0	0.999400
28961183	5.0	0.985140
29830790	5.0	0.310852
32726343	5.0	0.999228

Among all 569 customers with more than 30 reviews, 51 customers always give 5 stars, and most of them have also very high average positive sentiment probability. However, some customers have low positive sentiment probability even though they give 5 stars frequently. In other words, they rate 5-star even when they are not that satisfied. A good example here is customer 29830790, who has an average *star_rating* of 5.0 but with an average *prob_positive* of only 31.09%.

ii) Lowest Star-Rating Givers

customer_id	star_rating	prob_positive
24254971	1.000000	0.000009
35926111	1.000000	0.000337
37141039	1.000000	0.022365
12048856	1.435897	0.147344
18691646	1.619565	0.181318

There are 3 customers always giving 1-star. Among these three, customer 24254971 and 35926111 only reviewed watches, while customer 37141039 only reviewed mattresses in the furniture product category. All their reviews are without Verified Purchase. Since it makes no sense for a customer to buy more than 30 mattresses from one single brand, we conclude these reviews are made on purpose by competitors.

	customer_id	review_id	product_title	product_category	star_rating	verified_purchase	review_headline	review_body
26681	37141039	R2OD2BS1QUNW8L	Strobel Mattress 37000SPBR Organic Bio-Rest Ma...	Furniture	1	N	Strobel Does Not Honor Warranty	While I am not specifically reviewing this mat...
26682	37141039	R1P9ELV00JAVTP	Strobel Organic Supple-Pedic Lever-Bed 750 Kin...	Furniture	1	N	Strobel Does Not Honor Warranty	While I am not specifically reviewing this mat...
26683	37141039	R1160IXM5WU9XV	Strobel Organic Supple-Latex 3000 Queen	Furniture	1	N	Strobel Does Not Honor Warranty	While I am not specifically reviewing this mat...
26684	37141039	RHSG578XX4YNQ	Strobel Organic Complete Softside Waterbed Spe...	Furniture	1	N	Strobel Does Not Honor Warranty	While I am not specifically reviewing this mat...
26685	37141039	R3PAH159XRBF2A	rganic Waterbed Mattress Hydro-Support 1800 Si...	Furniture	1	N	Strobel Does Not Honor Warranty	While I am not specifically reviewing this mat...

iii) Customers Reviewed in Multiple Categories

In order to analyze customers' reviews across multiple categories, we further limit the data to reviews by customers who reviewed in 2 or more categories. This leaves us with 285 customers and 14,442 reviews. We then plot a Star-Sentiment plot using these reviews.
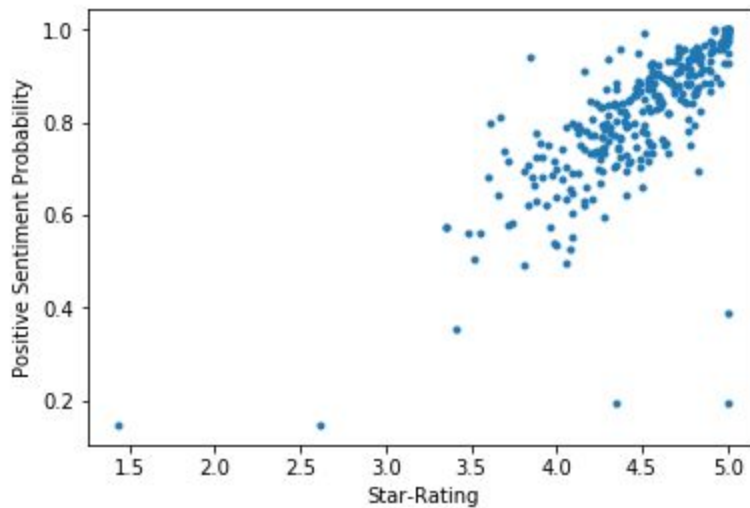
iv) Customers are separated into the following groups.
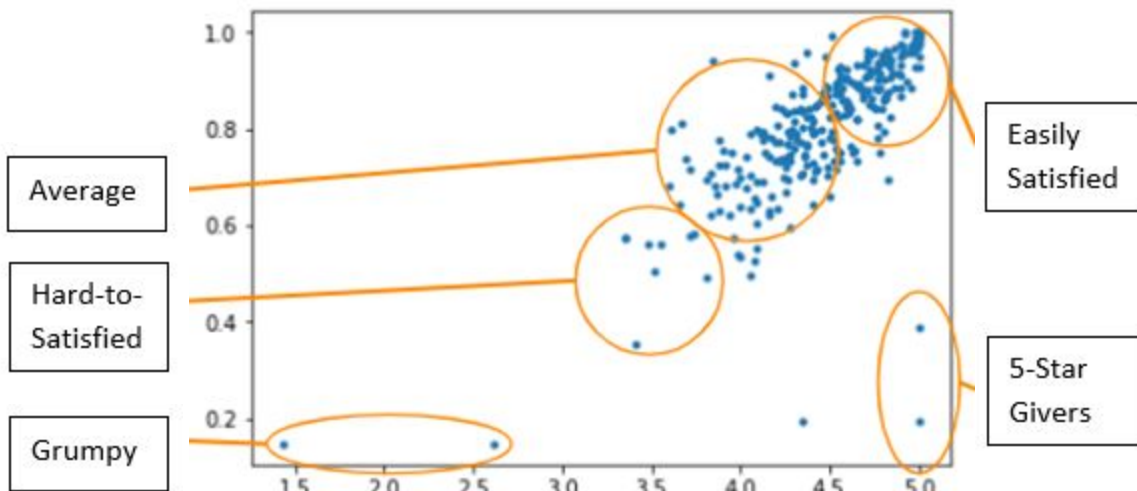
7. **TextBlob Introduction**

Aside from Amazon Comprehend, there are other NLP tools like:

· NLTK

· Spacy

· Stanford Core NLP

· TextBlob

They have their own advantages and disadvantages and can be utilized on different cases, such as what kinds of text analysis you want to perform and what your data looks like.

However, in general, TextBlob is easier to use because it has a nicer user interface and its documentation is thoroughly explained. There, it is more new-user friendly and we will emphasize on this new and popular NLP tool, TextBlob.

First of all, it 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.

Secondly, TextBlob is built on the shoulders of NLTK, google translator etc. to serve various features like:

· Part-of-speech tagging

· Noun phrase extraction

· Sentiment analysis

· Classification (by using Naive Bayes, Decision Tree)

· Language translation and detection powered by Google Translate

· Spelling correction

· Tokenization (splitting text into words and sentences)

· Word and phrase frequencies

· Parsing

· n-grams

· Word inflection (pluralization and singularization) and lemmatization

· Add new models or languages through extensions

· WordNet integration

And we will use sentiment analysis from TextBlob in Pyspark environment to analysis our Amazon review data and compare the results with the two.

Sentiment Analysis Using TextBlob:

The general pipeline of sentiment analysis is: data collection, text preparation, sentiment detection, sentiment classification and result presentation. And today we will go over those steps using PySpark and TextBlob.

· Step 1: Data Collection

After setting up Spark context and SparkSession, we can load the same Amazon review data as we used in Amazon Comprehend.

· Step 2: Text Preparation

       o remove non-ASCII characters

o fixed abbreviation

o remove irrelevant features:

Specifically, here we need to convert all text to lowercase, remove hyperlinks, remove @mentions, remove punctuation, remove numeric 'words' and remove non a-z 0-9 characters and words shorter than 1 characters.

· Step 3: Sentiment Detection

After text preparation, we then can get a cleaner text data for sentiment detection. In TextBlob, the textblob.sentiments module contains two sentiment analysis implementations. The first one is PatternAnalyzer (based on the pattern library) and the second one is NaiveBayesAnalyzer (an NLTK classifier trained on a movie reviews corpus).

Here, we will use PatternAnalyzer to detect the sentiment. It will return results as a named tuple of the form: Sentiment (polarity, subjectivity, [assessments]). Specifically, polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Subjectivity, ranging from 0 to 1, generally refer to personal opinion, emotion or judgment whereas objective refers to factual information.

After defining polarity as our sentient score, we can get the 10 rows of results.

· Sentiment Classification:

Instead of using the NaiveBayesAnalyzer implemented in TextBlob (reason: 1: it only has two categories: negative and positive, and here, it's better we also have the neutral category; 2: it is trained based on movie review data, and it's different from the Amazon review data(mostly are about various products) , and using it as the training data can give some bias; 3: the naïve bayes has an assumption: features are independent from each other, and it's very likely not true in our case.), we can just classify sentiment categories based on their sentiment score.

·

Compare TextBlob with Amazon Comprehend :

	Amazon Comprehend	TextBlob
Positive	**6696**	**6911**
Neutral	**69**	**2331**
Negative	**2306**	**758**
Mixed	**749**	

After getting the count of each sentiment category, we can see that TextBlob gives about the same amount of positive detections as Amazon Comprehend. It gives more Neutral results and fewer negative results. Since TextBlob can't be used to detect mixed feelings, it gives 0 result for mixed sentiment detections.

Review Text	Amazon Comprehend	TextBlob
Love my refrigerator! ! Keeps everything cold. will recommend!	Positive	Neutral
AS advertised	Positive	Neutral
It's not worth 22 dollars, I've heard it became of some value just not that high.	Negative	Neutral
Did the job but didn't match the original gray wheels.	Mixed	Positive
Cheap knock-off. Don't waste your time	Negative	Positive

However, simple count summary value can't help us determine which tool is better. We need to dive down to the review text itself. After randomly choosing some reviews, we find out that:

Amazon Comprehend has a better sentiment detection accuracy. The possible reasons are that it's based on Machine Learning while TextBlob(PatternAnalyzer) is based on dictionary) and Amazon comprehend has a custom set of entities

or text classification models that are tailored uniquely to text data. Further, since Amazon Comprehend has the level "Mixed", it can better classify reviews. By comparison, the only advantage of TextBlob we can find is that it is free.