Final project: Amazon Review Fashion Dataset

Uncovering Consumer Sentiments towards Amazon Fashion Products: A Data Analysis and Sentiment Classification Study

Karen Ardila López

Bootcamp Código Facilito

Project approach

Amazon is one of the largest companies today, making it a fascinating subject to research. In this case, I chose the **Amazon Fashion** data because I enjoy buying clothes online, and it is something that has become more widespread and ingrained in the daily lives of many people over the last few years. That is why, for this project, I decided to conduct a data analysis of this dataset, with the goal of analyzing customer sentiment toward the products they purchased. Furthermore, the goal is to obtain insights as a point of improvement for the company, as well as a computational model to predict the sentiment for the test data from the training data, thereby applying the knowledge gained in the data science bootcamp.

Sentiment analysis is textual contextual mining that identifies and extracts subjective information from source material, assisting businesses in understanding the social sentiment of their brand, product, or service while monitoring online conversations. This project consist in basic sentiment analysis.

The goal of this project is to find a way to **classify the sentiment of the reviews** so that the company can distinguish the contribution that each product makes and what types of products have better or worse sentiments for customers. As a result, they will have a better future experience, as well as increased production of the most positive products, which will increase sales for the company.

Concerning the data: The data is a large crawl of Amazon product reviews, specifically Amazon fashion. The data is in (loose) json format. Initially, no obvious trends are identified due to the large amount of data, so all of the processing learned in the bootcamp is performed, including exploratory data analysis.

Dataset

This dataset contains product reviews and metadata from Amazon, including 233.1 million reviews spanning May 1996 - Oct 2018. The category chosen is Amazon Fashion with 883,636 reviews.

Link: https://jmcauley.ucsd.edu/data/amazon_v2/index.html

Citation

Justifying recommendations using distantly-labeled reviews and fined-grained aspects Jianmo Ni, Jiacheng Li, Julian McAuley Empirical Methods in Natural Language Processing (EMNLP), 2019

Import Dataset

First, import all the necessary libraries.

```
/shared-libs/python3.9/py/lib/python3.9/site-packages/tqdm/auto.py:22: TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See https://ipywic
  from .autonotebook import tqdm as notebook_tqdm
```

Define two functions to read the .json.gz file and get the DataFrame from it.

Import the Amazon Fashion dataset using the previous functions.

Exploratory Data Analysis (EDA)

Visualize the table of the raw data to start the exploratory analysis.

	overall float64	verified bool	reviewTime object	reviewerID object	asin object	reviewerName ob...	reviewText object	summary object
0	5.0	True	10 20, 2014	A1D4G1SNUZWQO	7106116521	Tracy	Exactly what I	perfect

				T			needed.	replacements!!
1	2.0	True	09 28, 2014	A3DDWDH9PX2YX2	7106116521	Sonja Lau	I agree with the other review, the...	I agree with the other review, the.
2	4.0	False	08 25, 2014	A2MWC41EW7XL15	7106116521	Kathleen	Love these... I am going to order...	My New 'Friends'
3	2.0	True	08 24, 2014	A2UH2QQ275NV45	7106116521	Jodi Stoner	too tiny an opening	Two Stars
4	3.0	False	07 27, 2014	A89F3LQADZBS5	7106116521	Alexander D.	Okay	Three Stars
5	5.0	True	07 19, 2014	A29HLOUW0NS0EH	7106116521	Patricia R. Erwin	Exactly what I wanted.	Five Stars
6	4.0	True	05 31, 2014	A7QS961ROI6E0	7106116521	REBECCA S LAYTON	These little plastic backs work great...	Works great!
7	3.0	True	09 22, 2013	A1BB77SEBQT8VX	B00007GDFV	Darrow H Ankrum II	mother - in - law wanted it as a...	bought as a present
8	3.0	True	07 17, 2013	AHWOW7D1ABO9C	B00007GDFV	rosieO	Item is of good quality. Looks...	Buxton heiress collection
9	3.0	True	04 13, 2013	AKS3GULZE0HFC	B00007GDFV	M. Waltman	I had used my last el-cheapo fake...	Top Clasp Broke Within 3 days!

Verify the dimension of the data.

```
(883636, 12)
```

Implement the `.info()` function to get a concise summary of the dataframe.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 883636 entries, 0 to 883635
Data columns (total 12 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   overall         883636 non-null  float64
 1   verified        883636 non-null  bool   
 2   reviewTime      883636 non-null  object 
 3   reviewerID      883636 non-null  object 
 4   asin            883636 non-null  object 
 5   reviewerName    883544 non-null  object 
 6   reviewText      882403 non-null  object 
 7   summary         883103 non-null  object 
 8   unixReviewTime  883636 non-null  int64  
 9   vote            79900 non-null   object 
 10  style           304569 non-null  object 
 11  image           28807 non-null   object 
dtypes: bool(1), float64(1), int64(1), object(9)
memory usage: 81.7+ MB
```

Generate descriptive statistics with the `.describe()` function. Descriptive statistics include those that summarize the central tendency, dispersion and shape of the dataset distribution, excluding NaN values.

	overall float64	unixReviewTime f...	
count	883636.0	883636.0	
mean	3.9069401880412298	1456751249.1749997	
std	1.4182795015745813	44306912.83448761	
min	1.0	1036972800.0	
25%	3.0	1434240000.0	
50%	5.0	1462233600.0	
75%	5.0	1484265600.0	
max	5.0	1538352000.0	

Verifity that we will still have a sizable data to train an algorithm after removing the missing values.

```
878557
```

Remove missing or NA values from the dataset with the function `.dropna()`.

Examine the counts of unique values of the review scoring from 1 to 5.

```
5.0    3054
4.0     872
1.0     560
3.0     355
2.0     238
Name: overall, dtype: int64
```

Here we can see that there is a remarkable difference between the scores of the reviews, with 5.0 being the one with the highest number of ratings and 2.0 curiously having less than the others. In third place is 1.0, meaning that there are some reviews that were quite bad but despite this the 5.0 still presents a noticeable difference and surpassing it in quantity.

It wasn't possible to perform the Raw Review Dataset Analysis with the function `ProfileReport`, because the kernel process runs out of RAM with the raw data. Hence, the profile report is performed with the preprocessed dataset.

```
/shared-libs/python3.9/py/lib/python3.9/site-packages/pandas/core/frame.py:4441: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().rename(
Summarize dataset:  68%|      | 13/19 [00:05<00:03,  1.97it/s, Calculate auto correlation]/shared-libs/python3.9/py/lib/python3.9/site-packages/scipy/stats/
  warnings.warn("The input array could not be properly "
Summarize dataset: 100%|| 25/25 [00:11<00:00,  2.13it/s, Completed]
Generate report structure: 100%|| 1/1 [00:07<00:00,  7.19s/it]
Render HTML: 100%|| 1/1 [00:01<00:00,  1.11s/it]
Export report to file: 100%|| 1/1 [00:00<00:00, 124.92it/s]
```

In this report we were able to observe the statistics of the dataset, types of variables, interactions between variables, correlations, missing values, and duplicate rows. This is the first step to an exploratory data analysis (EDA) and to see the data in a much faster and visually pleasing way.

High cardinality refers to a database that has a large number of distinct values such as ID numbers, user names or email addresses. Having high cardinality data isn't a bad thing, and in this case it doesn't affect the performance or stability issues in your database, then it's not worth trying to lower the cardinality.

Drop columns style and image, these columns are in an unsupported type and they don't affect the analysis.

Implement functions `.sort_values` according to the productID and `.drop_duplicates` which returns the DataFrame with duplicate rows removed, considering certain columns ('reviewerID', 'reviewerName', 'reviewTime', 'summary', 'reviewText') for identifying duplicates.

Then, examine the datatype of each row's review data. Any row with data of a type different from a string will have that data converted to a string.

	overall float64 1.0 - 5.0	verified bool True 90.7% False 9.3%	reviewTime object 07 18, 2016 0.3% 11 2, 2015 0.3% 1227 others 99.4%	reviewerID object A2CERG67M... 0.1% A1RWF29TT... 0.1% 4826 others 99.9%	asin object B00ZW3SCF0 2.4% B019Q577XC 2.4% 1645 others 95.2%	reviewerName ob... Amazon Cu... 10.8% Kindle Custo... 0.2% 3984 others 88.9%	reviewText object Love it 0.1% I love it 0.1% 4892 others 99.8%	summary object Five Stars 6.4 Four Stars 0.9 4111 others 92.8
155	5.0	True	12 15, 2015	AFCKJ61O6600Y	B00008JVTT	Todd Gilligan	Great shirt. It popped as...	Stand-out shirt
397660	3.0	True	12 31, 2016	A6GWE7A06U480	B0002Z1JNK	Amazon Customer	I bought these for Halloween. They...	Cute and fun
397685	4.0	True	04 20, 2017	A10QE3HYX0TN7X	B00061RG3M	Andrea Cain	Pretty, dainty, simple. It's hard t...	Pretty, dainty, simple.
287	5.0	True	01 3, 2011	AHLJA9KTLXE54	B00061RG3M	Nicole	Exactly what I was hoping for. The...	Gorgeous Marqui for such a low...
513	1.0	True	01 13, 2015	AQ4WYO1VUHKI0	B00062NHH0	Caoxie Zhang	There are two holes after...	Really bad quality after washing
950	5.0	True	07 18, 2016	A1IVVE7YD1A2WS	B00063VWSA	Fred Davis	A nice hatband to dress up my...	Nice Hatband
1057	5.0	True	10 10, 2016	AUXRL3F3DGDU0	B00066G516	Devany Wills		

							Very cute socks!	Exactly what I wa
1055	3.0	True	10 17, 2016	A1DNF3F3298V10	B00066G516	Alyssa schmidt	I'm conflicted. I wrote a review bu...	Great for Halloween
1836	4.0	True	07 25, 2015	A2ESGTC8OVQCCA	B00080L00Q	Yudhi Wiyono	Purchased this on January 2015, an...	Classic sophisticated dop
1972	5.0	False	09 13, 2016	AEUP3BYE86XOY	B0008F6WMM	amy	The kids love it a year later the...	They loooove it

It is crucial to communicate to the model what is a **positive** and what is a **negative sentiment** while we perform sentiment analysis. We have ratings ranging from 1 to 5 in our rating column. We can classify reviews 1 and 2 as `negative`, and reviews 4 and 5 as `positive`. 3 is in the center, it is neutral. Therefore, we eliminate all threes.

Write a `sentiment` function that returns 1 if the rating is 4 or higher and 0 otherwise. **Positive sentiments: 1** and **negative sentiments: 0**.

Apply the sentiment function and add a new column at the end with the positive and negative sentiment represented as 1 or 0.

```
/tmp/ipykernel_35/4171760341.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  products['sentiment'] = products['overall'].apply(sentiment)
```

	overall float64 1.0 – 5.0	verified bool True 90.5% False 9.5%	reviewTime object 06 13, 2016 0.3% 07 18, 2016 0.3% 1203 others 99.4%	reviewerID object A2CERG67M... 0.1% A1RWF29TT... 0.1% 4486 others 99.8%	asin object B00ZW3SCF0 2.5% B019Q577XC 2.5% 1560 others 94.9%	reviewerName ob... Amazon Cu... 10.7% Kindle Custo... 0.2% 3723 others 89.1%	reviewText object Love it 0.2% Perfect! 0.1% 4549 others 99.8%	summary object Five Stars 6.9% Four Stars 0.9% 3794 others 92.2%
155	5.0	True	12 15, 2015	AFCKJ61O6600Y	B00008JVTT	Todd Gilligan	Great shirt. It popped as...	Stand-out shirt
397685	4.0	True	04 20, 2017	A10QE3HYX0TN7X	B00061RG3M	Andrea Cain	Pretty, dainty, simple. It's hard t...	Pretty, dainty, simple.
287	5.0	True	01 3, 2011	AHLJA9KTLXE54	B00061RG3M	Nicole	Exactly what I was hoping for. The...	Gorgeous Marquise for such a low...
513	1.0	True	01 13, 2015	AQ4WYO1VUHKI0	B00062NHH0	Caoxie Zhang	There are two holes after...	Really bad quality after washing
950	5.0	True	07 18, 2016	A1IVVE7YD1A2WS	B00063VWSA	Fred Davis	A nice hatband to dress up my...	Nice Hatband
1057	5.0	True	10 10, 2016	AUXRL3F3DGDU0	B00066G516	Devany Wills	Very cute socks! Fit my size 10 fee...	Exactly what I was looking for
1836	4.0	True	07 25, 2015	A2ESGTC8OVQCCA	B00080L00Q	Yudhi Wiyono	Purchased this on January 2015, an...	Classic sophisticated dop...
1972	5.0	False	09 13, 2016	AEUP3BYE86XOY	B0008F6WMM	amy	The kids love it a year later the...	They loooove it
2066	5.0	True	11 24, 2014	A362GLSIIBEP5Q	B0008F6WMM	usernamehidden	This was a big hit on Halloween. My...	Got us on the front page!
1998	5.0	True	12 22, 2015	A1CQ2IR7P9VV0Q	B0008F6WMM	Lulu	My son used this for his Halloween...	Perfect For Costume!

Create a single column by combining the "name" and "review" columns to get the training features ready by creating a function `combined_features`.

Apply the function and add a new column called `review_features` with the strings from the name and review columns in it.

```
/tmp/ipykernel_35/2663865909.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  products['review_features'] = products.apply(combined_features, axis=1)
```

Use the `pd.to_datetime()` method, to parse the date strings that have mixed formats to datetime with a standard format (default is YYYY-MM-DD).

```
/tmp/ipykernel_35/3744044002.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  products['reviewTime'] = pd.to_datetime(products['reviewTime'])
```

`strftime()` method takes `datetime` format and returns a string representing the specific format. Extract the year from the pandas DataFrame using %Y and save it in a new column called **year**.

```
/tmp/ipykernel_35/2095522489.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  products['year'] = products['reviewTime'].dt.strftime('%Y')
```

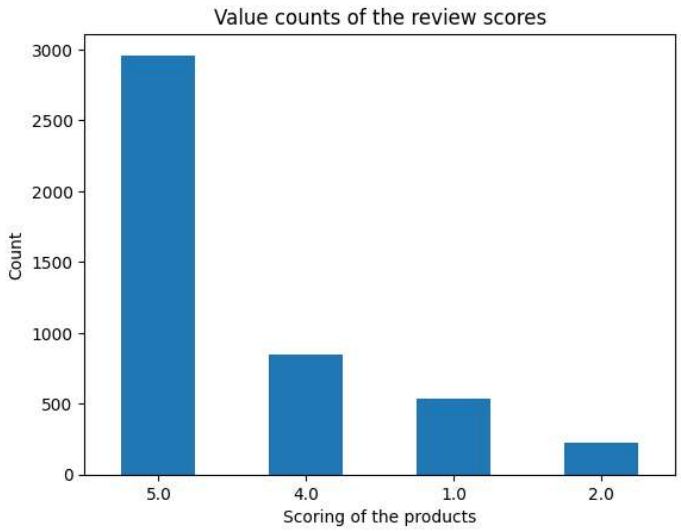
Visualization of the Data

	overall float64 1.0 – 5.0	verified bool True 90.5% False 9.5%	reviewTime dateti... 2010-07-23 00:00:...	reviewerID object A2CERG67M... 0.1% A1RWF29TT... 0.1% 4486 others 99.8%	asin object B00ZW3SCF0 2.5% B019Q577XC 2.5% 1560 others 94.9%	reviewerName ob... Amazon Cu... 10.7% Kindle Custo... 0.2% 3723 others 89.1%	reviewText object Love it 0.2% Perfect! 0.1% 4549 others 99.8%	summary object Five Stars 6.9% Four Stars 0.9% 3794 others 92.2%
155	5.0	True	2015-12-15 00:00:00	AFCKJ61O6600Y	B00008JVTT	Todd Gilligan	Great shirt. It popped as...	Stand-out shirt
397685	4.0	True	2017-04-20 00:00:00	A10QE3HYX0TN7X	B00061RG3M	Andrea Cain	Pretty, dainty, simple. It's hard t...	Pretty, dainty, simple.
287	5.0	True	2011-01-03 00:00:00	AHLJA9KTLXE54	B00061RG3M	Nicole	Exactly what I was hoping for. The...	Gorgeous Marquise for such a low...
513	1.0	True	2015-01-13 00:00:00	AQ4WYO1VUHKI0	B00062NHH0	Caoxie Zhang	There are two holes after...	Really bad quality after washing
950	5.0	True	2016-07-18 00:00:00	A1IVVE7YD1A2WS	B00063VWSA	Fred Davis	A nice hatband to dress up my...	Nice Hatband
1057	5.0	True	2016-10-10 00:00:00	AUXRL3F3DGDU0	B00066G516	Devany Wills	Very cute socks! Fit my size 10 fee...	Exactly what I was looking for
1836	4.0	True	2015-07-25 00:00:00	A2ESGTC8OVQCCA	B00080L00Q	Yudhi Wiyono	Purchased this on January 2015, an...	Classic sophisticated dop...
1972	5.0	False	2016-09-13 00:00:00	AEUP3BYE86XOY	B0008F6WMM	amy	The kids love it a year later the...	They loooove it
2066	5.0	True	2014-11-24 00:00:00	A362GLSIIBEP5Q	B0008F6WMM	usernamehidden	This was a big hit on Halloween. My...	Got us on the front page!
1998	5.0	True	2015-12-22 00:00:00	A1CQ2IR7P9VV0Q	B0008F6WMM	Lulu	My son used this for his Halloween...	Perfect For Costume!

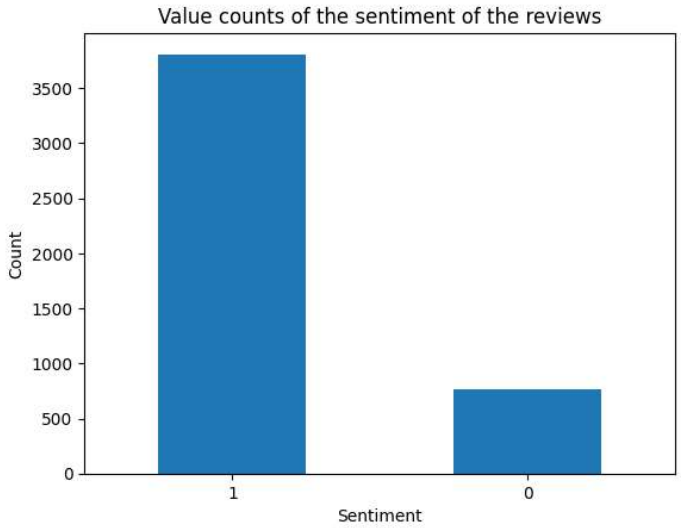
We can now see a small visualization of the distribution for each variable due to the data's dimension reduction. For example, we can see that 2016 has 47% more review data than any other year. We can also see that positive sentiment outnumbers negative sentiment in general.

First, an univariate analysis. Use `.value_counts()` to understand how many values of a given variable are in thedataset.

The variable `overall` represents the score from 1 to 5 of each product. Therefore, it's relevant to plot and see how balanced is the dataset in terms of review scores.



The next variable I want to examine is the sentiment of the reviews with a plot.



Then, I want to know the specific quantities for each sentiment.

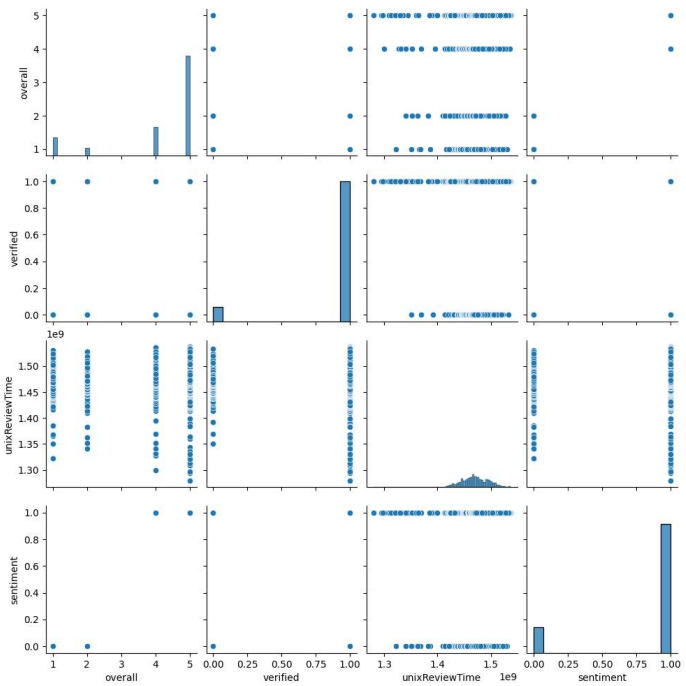
```
1    3808
0     762
Name: sentiment, dtype: int64
```

The goal is to discover intriguing relationships that demonstrate the influence of one variable on another, preferably on the target: `sentiment`. This information is valuable and provides the ability to act strategically.

In a scatterplot, `pairplot` plots all variables against each other. It is extremely helpful for capturing the most important relationships without doing every single possible combination.

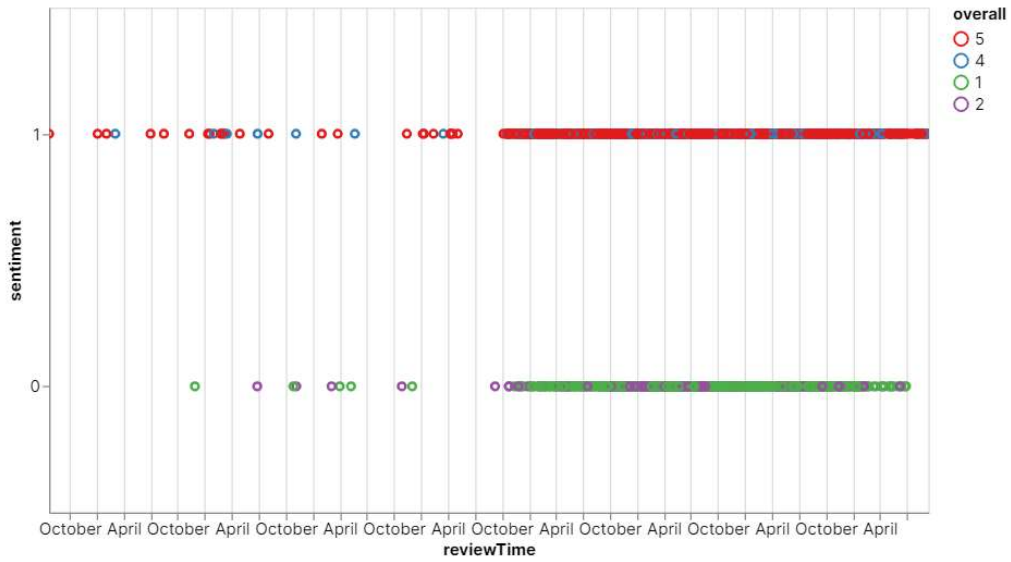
```
<__array_function__ internals>:180: RuntimeWarning: Converting input from bool to <class 'numpy.uint8'> for compatibility.
```

```
<seaborn.axisgrid.PairGrid at 0x7fb9d7b91a30>
```

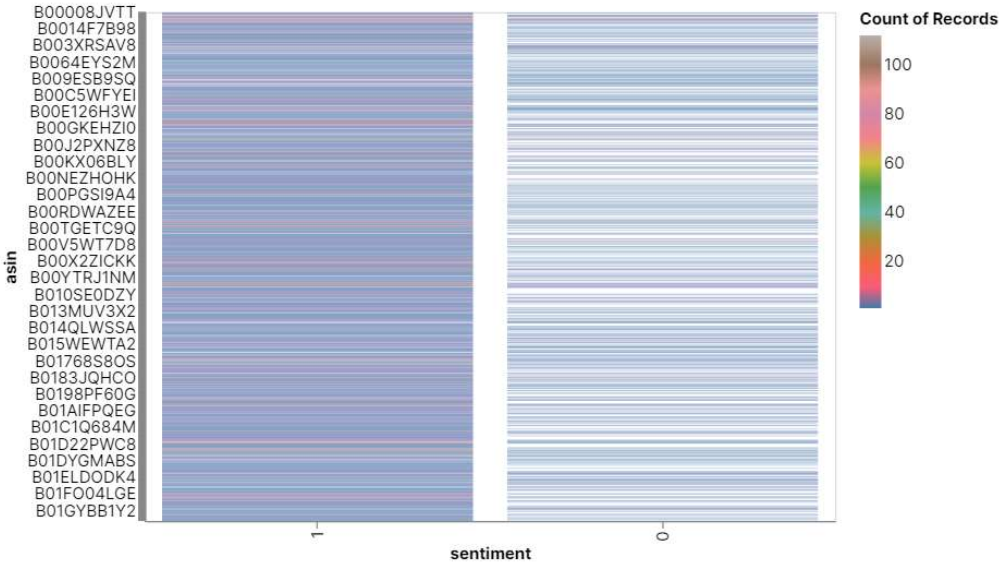


Using the Altair library, I wanted to plot the `productID` and `review score` from 1 to 5, aiming to broadly observing and analyzing this data, before going into a machine learning model.

This next visualization is intended to analyze the sentiment (positive or negative) based on the reviewTime in years, with a color differentiation for each review score, from 1 to 5 (without taking 3 into account because it is neutral). We can see that from 2011 to a little over 2014, there were more positive sentiment reviews than negative, with mostly 5.0 scores; since 2015, both positive and negative records have had many more records, but there is still a higher number of positive sentiments with the predominant scores being 1.0 and 5.0.

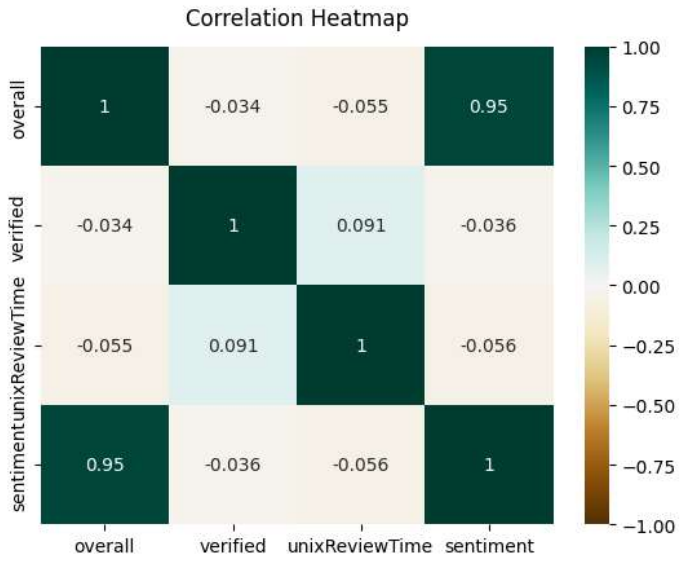


The following visualization is intended to show the overall sentiment based on the productID (asin) and the number of records with positive and negative sentiment. We can see that the quantity of positive sentiment varies significantly compared to the negative one; for example, we have 43 records of positive reviews for the product B000PHANNM, 44 records of positive reviews for the product B00I0VHS10, and 12 records of negative reviews for the product B000KPIHQ4. Most of the number of records for the negative reviews is around 1.



Correlation heatmap

Presents the same correlation matrices data but in a more visually appealing format. The heatmap is useful because it allows us to quickly determine which variables are highly correlated with one another. Furthermore, they indicate at a glance which variables are correlated, to what extent, and in which direction, as well as potential multicollinearity issues.



Word Cloud Graphic

A word cloud, also known as a tag cloud, is a visual representation of the words that comprise a text, with larger words appearing more frequently.

Negative sentiment: 0

The following keywords were found in the analysis of the negative sentiment reviewText: disappointed, different, size, picture, and received. This reveals that some of the most common problems with bad customer experiences, and thus negative sentiment in the reviews, could be related to receiving a different size or fit than expected, the clothes being different from the picture, and the feeling of disappointment they now associate with the garment. This is something that needs to be improved by the supplier in order to improve the customer satisfaction.

Positive sentiment: 1

Some of the keywords found in the analysis of the negative sentiment reviewText were: love, fit, perfect, great, price, and size. This reveals that the most common positive sentiment associated with the review was probably because the clothes fit great, they loved the product and thought it was great and perfect for what they ordered, and the quality-price relationship. This demonstrates the strong points in the sales, so it should be keep that way and use it when promoting the products.

Sentiment Classifier Model

Import all the necessary libraries for the modeling.

First, it is necessary to define the input variable `X` and the output variable `y`.

The next step is to divide the dataset into two parts: training and testing. The scikit-learn library's `train_test_split` function is useful. The training dataset will be used to **train** the model (75%), and the **test** dataset will be used to test its performance (25%).

Use the scikit-learn library's `CountVectorizer`. It creates a vector that contains all of the words in the string (converts a set of text documents into a token count matrix), then we fit the train and test data with it.

One of the most fundamental and often used algorithms for solving a classification problem is logistic regression. Since its underlying methodology is very similar to that of linear regression, it is known as "logistic regression." The **Logit function**, which is utilized in this categorization method, is where the term "Logistic" originates.

Logistic Regression uses the Sigmoid function and if it comes across an outlier, it will take care of it.

Because this is a binary classification, we will use `LogisticRegression`. Import the training data and fit it into the model.

```
/shared-libs/python3.9/py/lib/python3.9/site-packages/sklearn/linear_model/_logistic.py:444: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression()
```

Model Results

Predict the sentiments for the test data using the trained model described above. If the test features are managed to pass, it will predict the output `Y`, which is the sentiment data.

Output the test data. `1` is for good reviews and `0` for bad reviews.

```
array([1, 1, 1, ..., 0, 0, 1])
```

Accuracy: is defined as the proportion of correct predictions to total predictions. It is one of the most basic model measures. For our model, we must strive for high accuracy. If a model has a high accuracy, we can conclude that it makes correct predictions the majority of the time.

Precision: is the ratio of the correct positive predictions to the total number of positive predictions.

Recall: is calculated as the ratio of predicted positives to total positive labels.

F1 Score: depends on both the Recall and Precision, it is the harmonic mean of both the values.

While accuracy by itself cannot tell whether a model is good or terrible, accuracy when combined with precision, recall, and F1 Score can provide a good indication of the model's performance.

```
Accuracy: 0.9282589676290464
Precision score: 0.9457286432160804
Recall score: 0.9711042311661506
F1 score: 0.9582484725050916
```

The accuracy of the model is 92.8%

The precision score of the model is 94.6%

The recall score of the model is 97.1%

The F1 score of the model is 95.8%

Classification Report

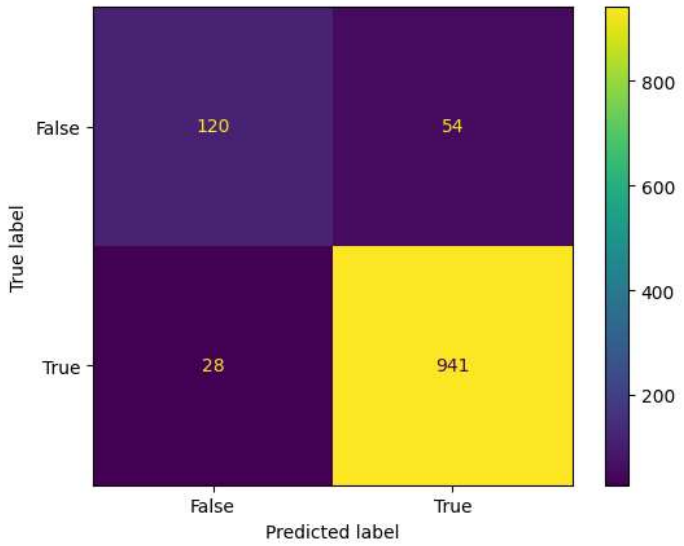
Build a text report showing the main classification metrics.

```
              precision    recall  f1-score   support

           0       0.81      0.69      0.75       174
           1       0.95      0.97      0.96       969
```

```
              precision    recall  f1-score   support

    accuracy                         0.93      1143
   macro avg       0.88      0.83      0.85      1143
weighted avg       0.93      0.93      0.93      1143
```

Confusion Matrix

A confusion matrix is a matrix to represent the number of True Positives, False Positives, True Negatives, and False Negatives.



The true positives have a value of 941 in this confusion matrix, followed by the true negatives, which have a value of 120, then 54 false positives, and 28 false negatives. Allowing us to infer and confirm that the model's accuracy is good, as calculated previously along with other measurements of the model's performance, with great results and a significantly lower percentage of false negatives and false positives.

Receiver Operating Characteristic (ROC) and Detection Error Tradeoff (DET)

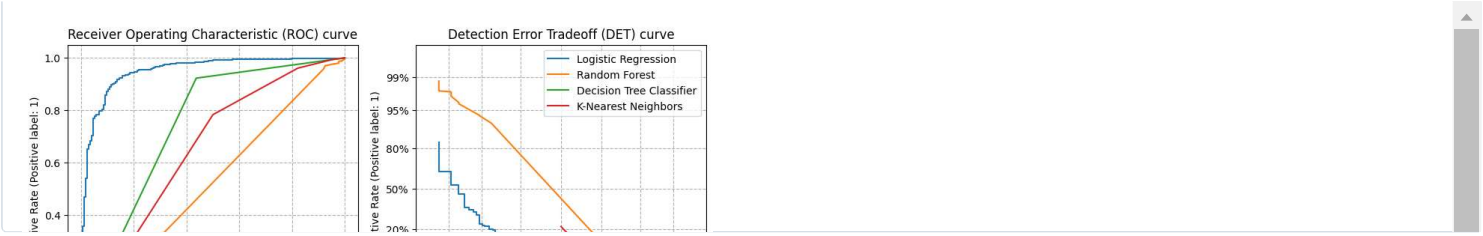
The Y axis of **ROC** curves contains the true positive rate (TPR) and the X axis contains the false positive rate (FPR). This means that the "ideal" point is in the plot's upper left corner, with an FPR of zero and a TPR of one.

DET curves are ROC curves with the False Negative Rate (FNR) plotted on the y-axis instead of the TPR. The "ideal" point in this case is the origin (bottom left corner).

Visualy compare classifiers and their statistical performance across thresholds using the ROC and DET curves.

Plot ROC and DET curves. DET curves are typically plotted in standard deviate scale. To accomplish this, the DET display uses `scipy.stats.norm` to transform the error rates returned by the det curve and the axis scale.

```
/shared-libs/python3.9/py/lib/python3.9/site-packages/sklearn/linear_model/_logistic.py:444: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```



Comparing the four models with the ROC and DET curves, the best performance is indeed with the Logistic Regression with and AUC (Area under the ROC Curve) of 0.95. Therefore, is the model we keep for the metrics results and analysis of insights for this project.

Insights

1. The data shows that there is a significant difference between the scores of the reviews, with 5.0 having the highest number of ratings and 2.0 having less than the others.
2. There is a noticeable difference in review data between years, with 2016 having 47% more data than any other year.
3. Positive sentiment outnumbers negative sentiment in general.
4. Analysis of sentiment by review time shows that from 2011 to a little over 2014, there were more positive sentiment reviews than negative.
5. The overall sentiment based on the product ID shows that the quantity of positive sentiment varies significantly compared to the negative one.
6. Negative sentiment reviewText keywords found include "disappointed," "different", "size", "picture", and "received", indicating that issues with size and product not matching the picture are common complaints.
7. Positive sentiment reviewText keywords found include "love", "fit", "perfect", "great", "price", and "size", indicating these are strong points in the sales, so it should be keep that way and use it when promoting the products.
8. The accuracy of the model was 92.8%, being able to classify the test data appropiately and giving a starting point to then use new data. The other metrics also showed a good performance of the model.

References

References used for the development of this project

1. https://pandas.pydata.org/docs/index.html
2. https://nijianmo.github.io/amazon/index.html
3. https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model
4. https://towardsdatascience.com/understanding-logistic-regression-9b02c2aec102
5. https://towardsdatascience.com/top-10-binary-classification-algorithms-a-beginners-guide-feeacbd7a3e2
6. https://towardsdatascience.com/exploratory-data-analysis-in-python-a-step-by-step-process-d0dfa6bf94ee
7. https://scikit-learn.org/stable/auto_examples/model_selection/plot_det.html#sphx-glr-auto-examples-model-selection-plot-det-py
8. https://medium.com/@szabo.bibor/how-to-create-a-seaborn-correlation-heatmap-in-python-834c0686b88e
9. https://huggingface.co/blog/sentiment-analysis-python
10. https://medium.com/analytics-vidhya/data-preparation-and-text-preprocessing-on-amazon-fine-food-reviews-7b7a2665c3f4
11. https://towardsdatascience.com/a-complete-sentiment-analysis-algorithm-in-python-with-amazon-product-review-data-step-by-step-2680d2e2c23b
12. https://youtu.be/ckLy603HQGI