February 2024

Text Mining and Search Project

AMAZON FINE FOOD REVIEWS

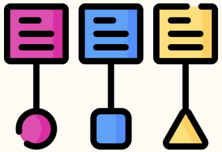
Francesca Corvino
Enrico Mannarino
Christian Persico

Master's Degree in Data Science

Project goals

Text classification

It consists of the assignment of natural language documents to **predefined categories** according to their content.



Text summarization

The process of condensing a large volume of text into a **shorter version**, preserving key information and the overall meaning.

Data exploration

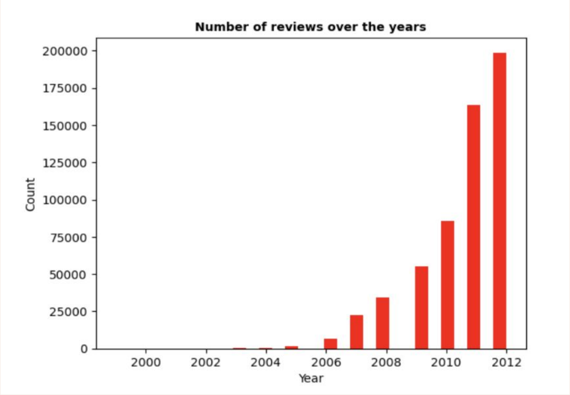
The dataset is composed by **10** variables:

- **ID**: a unique identifier for each review in the dataset
- **ProductId**: the unique identifier of the reviewed product
- **UserId**: the unique identifier of the user who wrote the review
- **ProfileName**: the profile name of the reviewing user
- **HelpfulnessNumerator**: the number of votes indicating a review helpfulness
- **HelpfulnessDenominator**: the total number of votes a review has received for helpfulness
- **Time**: the time when the review was posted in UNIX format
- **Summary**: a brief summary of the evaluation provided by the user
- **Text**: the full text of the review detailing the user's experience with the product
- **Score**: a rating between 1 and 5

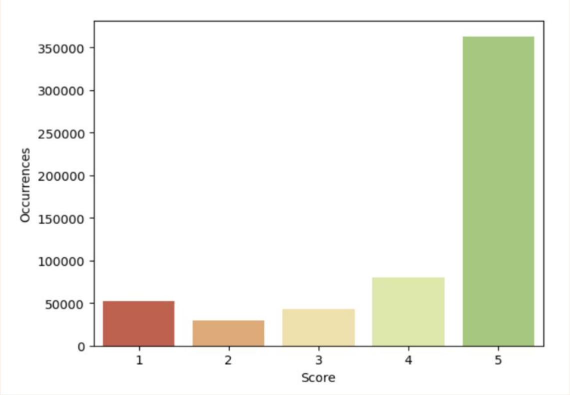
Removing duplicates considering **Score** and **Text**:

568,454 reviews → **393,675 reviews**

REVIEWS COUNT OVER THE YEARS



SCORE DISTRIBUTION



Text processing

This step involves cleaning and transforming unstructured text data preparing it for analysis.

Normalization

- Lowercase conversion
- Link/HTML removal
- Emoji removal
- Accents removal
- Punctuation removal
- Whitespace normalization

Language detection and correction

- Identification of the reviews that were **not in english**
- Translation using the **google translate API**

Decontractions

- Expansion of some contractions to their full form

Stop words removal

- Remotion of common words like “a” or “and” - addition of “not”, “product”, “amazon” and “would”

Tokenization

- Breaking the text into small units

Lemmatization

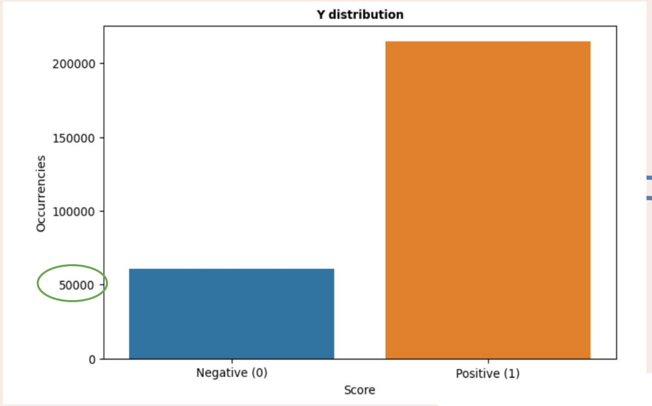
- Reduction of the words to their base form

Text classification

Analyze text and make predictions, categorizing each piece of text into one of two predefined categories based on its content.

Target variable

1 if the **Score is greater than 3,** 0 otherwise.

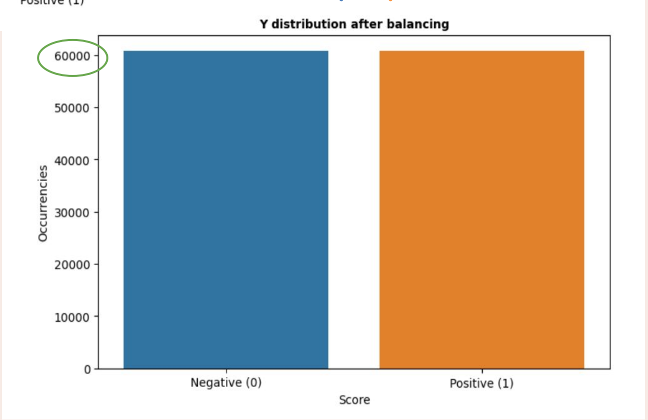


Training and test sets

We divided the dataset into a training and test set with a **70%-30%** proportion.

Class imbalance

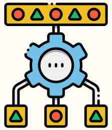
We had to **downsample** the training set as there were much more good reviews (306,814) than bad ones (86,856).



Evaluation metric

The model's performance varied significantly between the two classes. We mainly referred to the Area Under the Curve (**AUC**) score, an accurate and single performance measure across all classification thresholds, indicating the presence of a high discrimination level between the positive and negative classes.

Text classification



Text representations adopted

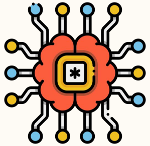
BOW

Creating a "bag" of all the words that appear in a given text corpus, without taking into account their order or context. Each unique word represents a feature, and the text is subsequently represented as a vector, which indicates the presence and **frequency of each word** within the text.

TF-IDF

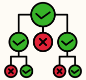
TF-IDF takes into account the scarcity of a word in all documents in the corpus. This approach assigns more weight to unique terms that appear only in a single document, which may indicate their higher importance and **less weight to common words** that appear across many documents.

In both cases, we set a **threshold of 0.001** to determine the minimum document frequency for words to be included in the analysis.



Classification models

Logistic Regression



Random Forest

Support Vector Machine

Text Classification

LOGISTIC REGRESSION

Bag Of Words

	precision	recall	f1-score	support
Negative	0.84	0.61	0.71	35880
Positive	0.85	0.95	0.90	82221
accuracy			0.85	118101
macro avg	0.84	0.78	0.80	118101
weighted avg	0.85	0.85	0.84	118101

AUC = 0.91

TF-IDF

	precision	recall	f1-score	support
Negative	0.86	0.61	0.71	36812
Positive	0.84	0.95	0.89	81289
accuracy			0.84	118101
macro avg	0.85	0.78	0.80	118101
weighted avg	0.85	0.84	0.84	118101

AUC = 0.92

Most informative features

Negative connotation and coeff.

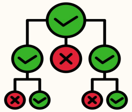
Positive connotation and coeff.

-8.6857	worst	8.1036	delicious
-8.0966	not	7.9041	great
-7.2186	disappoint	7.4417	perfect
-6.7003	disappointment	7.3915	best
-6.6353	unfortunately	7.0188	highly
-6.4977	terrible	6.7160	love
-6.1690	ok	6.3081	excellent
-6.1156	hop	5.7912	wonderful
-6.0587	awful	5.5137	pleasantly
-5.7617	horrible	5.3446	amaze

The overall accuracies of 0.85 and 0.84 indicate that most predictions were correct. However, while is proficient in correctly identifying positive reviews (higher **recall score** for the 'Positive' class), it's not as effective in identifying negative reviews.

This highlights the importance of considering other evaluation metrics. We referred to the **ROC curve** for this purpose, which presented AUC scores of 0.91 and 0.92.

We found that **words** with high positive or negative coefficients were strongly associated with their **respective connotation** and also the same for the two representations.



Text Classification

RANDOM FOREST

Bag Of Words

	precision	recall	f1-score	support
Negative	0.78	0.50	0.61	40463
Positive	0.78	0.93	0.85	77638
accuracy			0.78	118101
macro avg	0.78	0.72	0.73	118101
weighted avg	0.78	0.78	0.77	118101

AUC = 0.86

TF-IDF

	precision	recall	f1-score	support
Negative	0.79	0.49	0.61	41856
Positive	0.77	0.93	0.84	76245
accuracy			0.77	118101
macro avg	0.78	0.71	0.72	118101
weighted avg	0.78	0.77	0.76	118101

AUC = 0.86

Most informative features

```
Top 10 Feature Importance:

not: 0.09051019653468385
love: 0.05258313065098667
great: 0.05217696615727457
bad: 0.034820663626306474
best: 0.03133535235785007
disappoint: 0.028501298838598907
perfect: 0.02653736523091155
delicious: 0.0251775373976598
ok: 0.01916613791625092
think: 0.016940613107366883
```

Positive connotation

Negative connotation

Overall **accuracy** got worse and there is a noticeable difference in recall between the two classes indicating a **tendency to predict** positive outcomes more accurately than negative ones. The **AUC** value also decreased to 0.86.

Words like "**not**" and "**love**" have the highest importance scores, suggesting they are key indicators of sentiment in the reviews.

Text Classification

SUPPORT VECTOR MACHINE

Bag Of Words

	precision	recall	f1-score	support
Negative	0.28	0.22	0.24	33512
Positive	0.72	0.78	0.75	84589
accuracy			0.62	118101
macro avg	0.50	0.50	0.50	118101
weighted avg	0.59	0.62	0.60	118101

TF-IDF

	precision	recall	f1-score	support
Negative	0.55	0.30	0.39	47815
Positive	0.64	0.83	0.72	70286
accuracy			0.62	118101
macro avg	0.59	0.57	0.55	118101
weighted avg	0.60	0.62	0.59	118101

The model showed **suboptimal performance** with both text representation adopted.

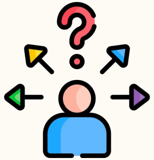
Precision, recall and consequently F1-score values for the 'Negative' class were low.

This suggests that the model had difficulty in correctly identifying negative reviews.

Model selection

We prioritized using the **AUC as our decisive metric**, taking into account the trade-off between a true positive rate and a false positive rate.

After evaluating the models based on AUC value, the **logistic** one with the **TF-IDF** text representation technique **performed the best.**



Recognizing the most common words in misclassified reviews can help identify patterns or similarities in prediction errors. This will help refining our text representation or adjusting the model to enhance its predictive accuracy.

What are the **most frequent tokens found in misclassified reviews**?

not	29658
like	11888
taste	10130
get	7406
good	7371

It might be **misunderstanding the context** in which some words are used, like in "Definitely not good!" potentially expected as a positive review. We can try to **integrate n-grams**, which let the model consider n contiguous tokens!

Text Classification

LOGISTIC REGRESSION & N-grams 👑

TF-IDF Vectorizer with a specified **n-gram** range to capture both **unigrams and bigrams** in our texts.

After training with this updated set of features, we observed a **significant improvement** in our model's performance metrics.

	precision	recall	f1-score	support
Negative	0.86	0.61	0.71	36812
Positive	0.84	0.95	0.89	81289
accuracy			0.84	118101
macro avg	0.85	0.78	0.80	118101
weighted avg	0.85	0.84	0.84	118101

	precision	recall	f1-score	support
Negative	0.87	0.64	0.74	35048
Positive	0.86	0.96	0.91	83053
accuracy			0.86	118101
macro avg	0.87	0.80	0.82	118101
weighted avg	0.87	0.86	0.86	118101

$AUC_{logit} = 0.92$

$AUC_{n\text{-}logit} = 0.94$

Text Summarization

Extractive Summarization

Abstractive Summarization

LSA

The $\mathbf{A=U\Sigma V^T}$ decomposition enables the selection of sentences, that best capture the representation of the concept in the document.

BERT

Convert sentences into vector representations for similarity comparison. Apply a clustering algorithm to the vector matrix. Identify sentences nearest to the cluster centroids.

BART

Seq2seq transformer with bidirectional encoder and autoregressive decoder, trained by corrupting the input with an arbitrary noising function and, only afterwards, learning to reconstruct the original text.
We used the model pre-trained on CNN/Daily Mail dataset, without fine-tuning.

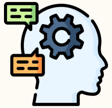
Evaluation

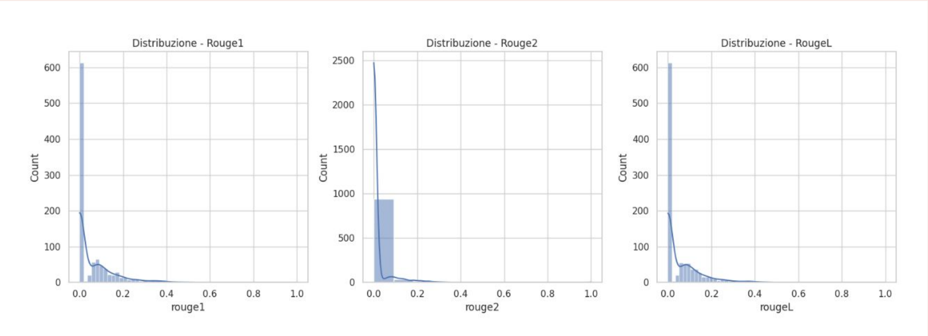
To carry out an **automatic inspection** of the predictions, summaries made by the model are compared with those made by professionals.
Rouge is the score used to compute the similarity between the two.

Human factor is not negligible when we want to perform an effective inspection of the predictions. Finding interesting elements can enable scientists to understand the errors made by the models and improve their capability.

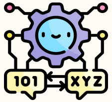
Automatic Evaluation

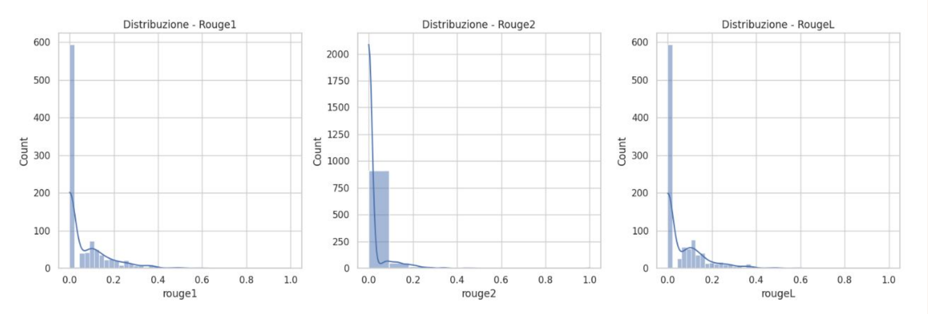
LSA





BERT





Performance are poor for all the three scoring metrics

The **summary box** is not used properly by the users.

Moreover, even if the summary box was used correctly, a human user doesn't repeat words within the summary and the corpus.

On this premises, it's not possible for extractive systems to perform well on metrics based on co-occurring n-grams.

Human Evaluation

Original text
Great taffy at a great price. There was a wide assortment of yummy taffy. Delivery was very quick. If your a taffy lover, this is a deal.

Summary wrote by the user
Great taffy

Summarized text by LSA
There was a wide assortment of yummy taffy.

Summarized text by BERT Summarizer
There was a wide assortment of yummy taffy.

Summarized text by BART Large CNN
Great taffy at

Original text
I am very satisfied with my Twizzler purchase. I shared these with others and we have all enjoyed them. I will definitely be ordering more.

Summary wrote by the user
Love it!

Summarized text by LSA
I am very satisfied with my Twizzler purchase.

Summarized text by BERT Summarizer
I am very satisfied with my Twizzler purchase.

Summarized text by BART Large CNN
I am very satisfied with

Original text
I love this candy. After weight watchers I had to cut back but still have a craving for it.

Summary wrote by the user
Twizzlers

Summarized text by LSA
After weight watchers I had to cut back but still have a craving for it.

Summarized text by BERT Summarizer
After weight watchers I had to cut back but still have a craving for it.

Summarized text by BART Large CNN
I love this candy.

Original text
Oh, I love these chips! And they're so hard to find where I am, and when I do find them, they're usually $1-2 more per bag for less. Great that these are so cheap here at Amazon.

Summary wrote by the user
Delicious as always!

Summarized text by LSA
Great that these are so cheap here at Amazon.

Summarized text by BERT Summarizer
And they're so hard to find where I am, and when I do find them, they're usually $1-2 more per bag for less.

Summarized text by BART Large CNN
These chips are hard to