Context-based Relevancy Classification: A Machine Learning Approach

Liyue Chen, Yi Gong, Allen (Yi Xin) Hu, Yihang Lin, Congyu Pu, Letian Qi, Hiridaya Sanjayb Shinde
Columbia University, M.S. in Business Analytics – Capstone Project
5/1/23

Table of Contents

1. Executive Summary

Project Title: Context-Based Relevancy Classification

Corporate Sponsor: BlackSwan Data

This report presents the development and evaluation of machine learning models to identify consumer sentiments towards non-alcoholic beverages using social media data, specifically tweets. The goal is to gain insights into consumer preferences and behavior for marketing and advertising purposes.

BlackSwan Data provided a dataset containing textual documents grouped into 17 categories. Data preprocessing and feature engineering were performed, including complexity analysis and hashtag exploration. Four baseline models were implemented using scikit-learn library and compared based on accuracy, precision, recall, and F-1 score. Advanced techniques, such as BERT and GPT-J transformers, were also explored.

The BERT model outperformed all other models with an accuracy of 0.77, followed by logistic regression (0.75), SVM (0.74), and Naïve Bayes (0.66). However, it is important to consider factors such as robustness, scalability, and training cost when selecting an appropriate model, as transformer-based models typically require more computational resources.

To better understand the decision-making process behind individual predictions, local interpretable model-agnostic explanations (LIME) and SHapley Additive exPlanations (SHAP) methods were utilized. These methods help identify essential features for classification and provide insights into model behavior.

The project leverages CometML for centralized model experimentation, enhancing communication between data scientists and stakeholders. A Streamlit web interface was developed to showcase the model implementation.

2. Introduction

2.1 Client Information

BlackSwan Data is a company that specializes in using data science and creative technology to help clients gain insights and solutions that can transform the way they approach their businesses. By analyzing consumer behavior, BlackSwan Data creates innovative and cutting-edge technology to deliver better outcomes for its clients.

One of BlackSwan Data's primary products is called Trendscope, a data science tool designed to assist clients with product innovation. Trendscope collects large volumes of publicly available social data, which is then used to map out and score the importance of various social trends that could be relevant to our clients. Our work often involves natural language processing (NLP) and big data, as we analyze vast amounts of information, such as 10% of everything said on Twitter.

2.2 Glossary

- **Topics:** topics of discussions Black Swan tracks within datasets. In Non-Alcoholic Beverages, for instance: Orange Juice, Orange, Artificial Ingredients, etc. Sometimes a topic is referred to as "sublens".

- **InclusionTerm:** Each topic has its own inclusion terms, which are matched against text documents to determine the presence of a given topic. For instance: the topic Orange has orange and oranges as inclusion terms. If a document contains any of these two terms, we assume that the topic "Orange" is discussed within the document.

- **Lens:** categories of topics. Main categories: Products, Brands, Ingredients, Themes, Benefits.

- **Definition of "Relevance"**[i]

- **Definition of "Non-Alcoholic Beverages"**[ii]

2.3 Engagement Methods

- **Weekly Touchpoints with BlackSwan**

- **Extended Interviews with**
    - BlackSwan PMs
    - Representatives from CometML
    - Office Hours with lead data scientists at BlackSwan
- **Team Composition:**
    - 4 team members held responsible for model production
    - 1 team member responsible for feature engineering
    - 1 team member responsible for liaison and project roadmapping

3. Project Description

3.1 Purpose & Objectives

The end-goal of this project is to identify inter-document (tweet) scores for non-alcoholic beverage consumer sentiments. For example, If a tweet is “I used to love coffee, but not anymore”, the current model will give a negative score for ‘coffee’. Upon the baseline models, we are exploring BERT and GPT implementations upon a MLOps model to best predict outcomes and build a pipeline upon which BlackSwan can show this to clients and showcase their models.

3.2 Deliverables

1. Final Written Report
2. Fine-tuned baseline/pre-trained ML model/workflow assets
3. StreamLit web interface as model implementation
4. Stakeholder presentations

4. Data Exploration

4.1 Dataset Overview

Black Swan Data has various datasets, one of which is the Non-Alcoholic Beverages dataset for the US market. In this dataset, they have textual documents from various sources, and several topics we match against these documents, like generic products (i.e.: Orange Juice), specific products of a brand (i.e.: Pepsi Max), beverage ingredients (i.e.: Orange), or other topics of discussion like themes (i.e.: Artificial Ingredients) or benefits (i.e.: Prevents Dementia) attributed to beverages.

4.2 Annotations

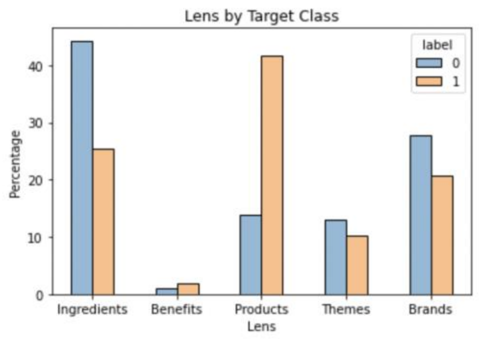
The data scripted using databricks are social media text messages of domain structure mapping data which contains food and beverages content. Our analysis goal is to apply machine learning and algorithms in order to identify if a drink contains alcohol. The category knowledge and lens constant is derived from multiple third party sources (eg. Walmart, TESCO, Kroger,etc). The beverages dataset captures conversations around the consumption of non-alcoholic drinks and their brands.

Entity Name	**Description**
_unit_id	id of the document
message	document text
market, category	constant, *"US Non-Alcoholic Beverages"*
lens	category of the topic, either Products, Brands, Themes, Ingredients, Benefits

topic	target topic being annotated for relevancy in the text
inclusion	inclusion term, matched in the text which indicates the presence of the target topic
label	yes if the target topic is relevant, otherwise no
taxonomies_snacking	topics matched from the domain of snacking, format (domain, lens, topic)
taxonomies_beverages	topics matched from the domain of non-alcoholic beverages, format (domain, lens, topic)
taxonomies_skincare	Same as above

4.3 Data Preprocessing

We have 17 different categories in our taxonomy to group all our products with exceptions of supermarket own brands, baby food, snacking and coffee capsules. The 6 lenses we applied are brands, products, ingredients, themes and benefits, and we will apply these on labeling.



Our team’s approach to basic data cleaning include the following steps (detailed methodology may vary dependent upon mode types):

Action	**Standard Operating Procedures**
Unicode Character Removal	a. Encode the input text using the 'ascii' encoding and ignore any Unicode characters. b. Decode the resulting encoded text to obtain a clean version without Unicode characters.
Link Removal	Remove any HTTP or HTTPS links from the text using regular expressions.
Emoji Removal	a. Create a regular expression pattern to match emojis. b. Remove any emojis found in the text using the created pattern
Text Cleaning	a. Remove any contractions using regular expressions. b. Remove any alphanumeric strings (strings containing both alphabets and numbers) using regular expressions. c. Remove any occurrences of two or more consecutive whitespace characters with a single whitespace character using regular expressions. d. Remove any occurrences of a single non-word, non-whitespace character surrounded by whitespace using regular expressions.

Lowercase Conversion	Convert all text to lowercase
Punctuation Removal	Remove any punctuation marks from the text
Tokenization	Split the input text into individual words or tokens using the `nltk` library's `word_tokenize` function.
Stopword Removal	Remove any stopwords (commonly used words that are unlikely to be informative in identifying named entities) from the tokens
Lemmatization	Lemmatize the tokens (reduce words to their base form) using the `nltk` library's `lemmatizer`

By applying these preprocessing steps, we aim to obtain a cleaned and standardized text that is more suitable for further analysis, feature extraction, and model implementation. And indeed, from empirical model training, we discovered that applying data cleaning approaches as followed above has a **5 - 10% incremental value on accuracy improvement**.

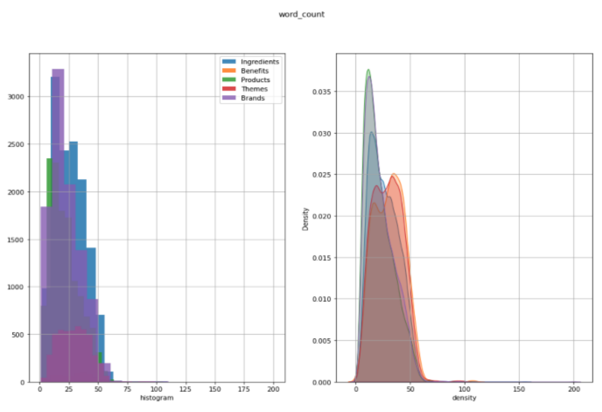
5. Feature Engineering

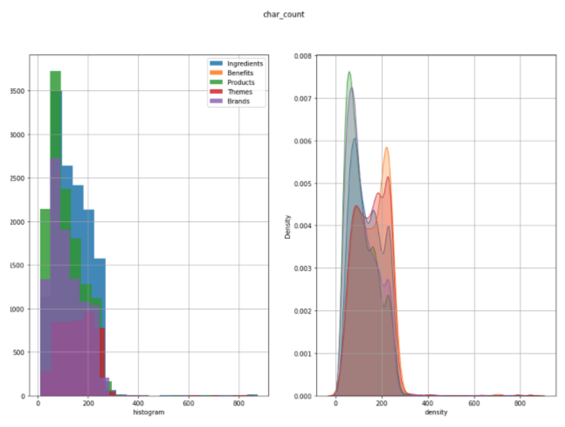
5.1 Complexity Analysis

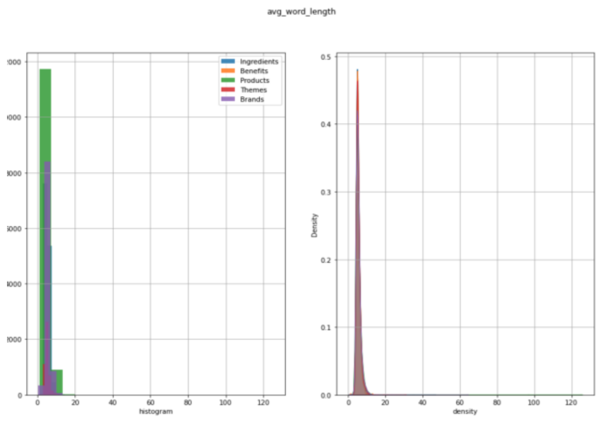
The complexity analysis aimed to identify patterns and potential biases in the data by measuring the word and character count of each sentence.

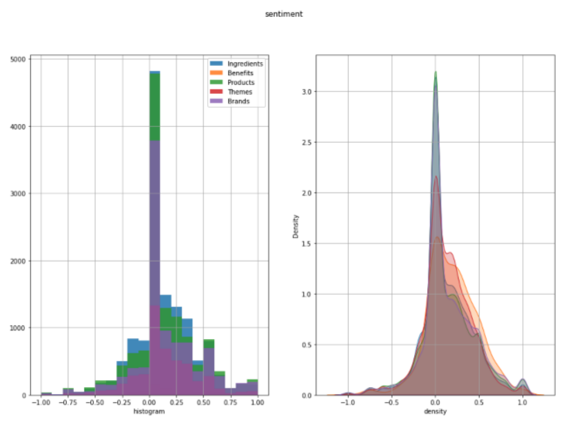
To conduct the analysis, two attributes were created: "word_count" and "character_count". These attributes were used to calculate the average word length of each sentence, with the goal of identifying whether certain categories had longer average word lengths than others.Upon examining the summary numbers, it was found that each category had similar levels of word and character count. To gain a deeper understanding of the distribution within each category, density plots were used. The results showed that each category had a similar distribution.Based on these findings, it was concluded that the different complexity of each text within each category is unlikely to introduce any significant bias in the classification task.

In addition to the complexity analysis, a simple sentiment analysis was also conducted using the same technique. The findings suggested that all categories had a relatively neutral sentiment. Overall, this study demonstrates the importance of conducting complexity analysis in NLP classification tasks to identify potential biases and gain a deeper understanding of the data.



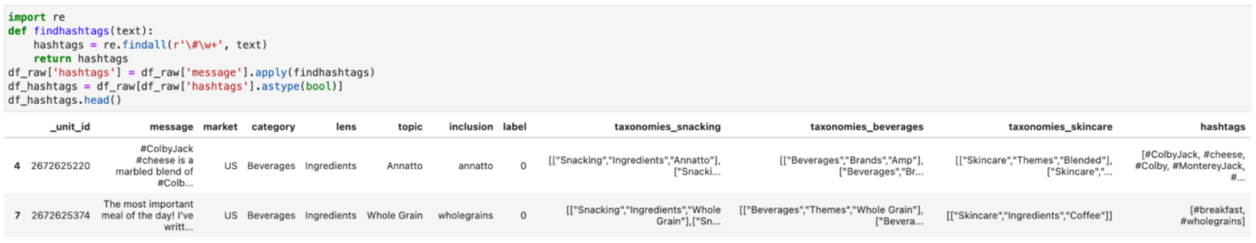






5.2 Hashtags Exploration

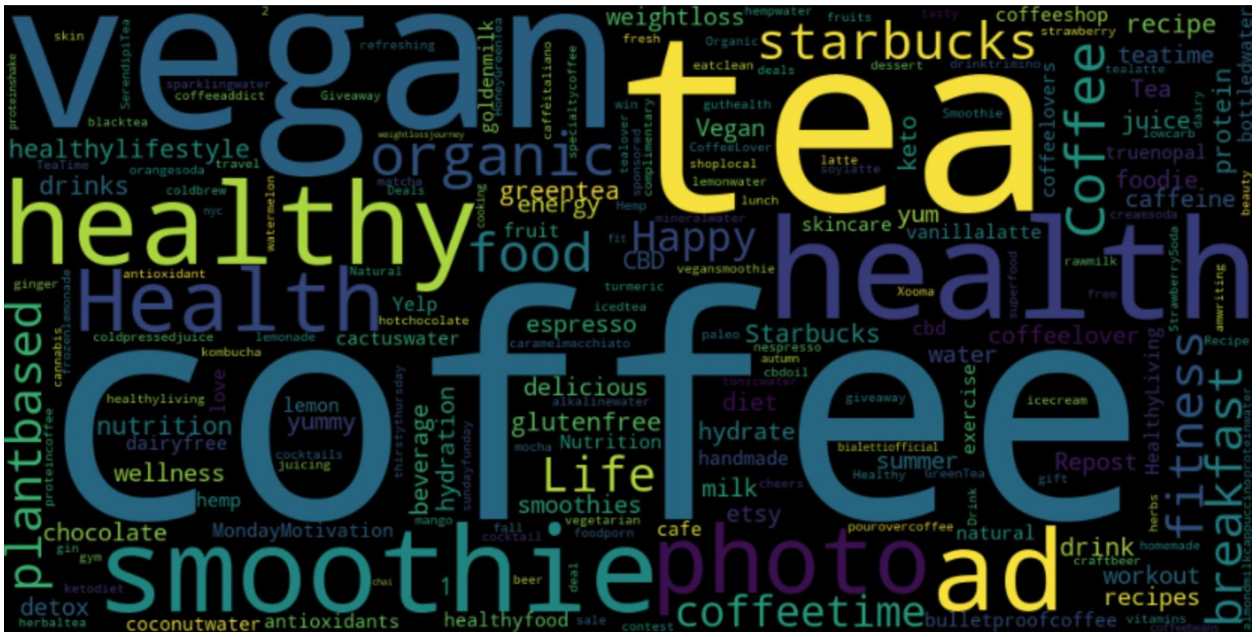
In this study, potential patterns within a dataset of messages were further explored by examining the presence of the word "URL" in tweets, as well as the use of hashtags. Given that hashtags are often used by users to convey additional meaning or context, it was hypothesized that the use of specific hashtags could be indicative of certain topics or labels.To investigate this hypothesis, each sentence containing a hashtag was split, and a list of hashtag words was created for each message. These lists were then compared to the true label or topic of the message.The results showed that certain hashtags were indeed associated with specific topics or labels. For example, messages containing the hashtag #Tea was more likely to be classified as relevant.



```python
import re
def findhashtags(text):
    hashtags = re.findall(r'\#\w+', text)
    return hashtags
df_raw['hashtags'] = df_raw['message'].apply(findhashtags)
df_hashtags = df_raw[df_raw['hashtags'].astype(bool)]
df_hashtags.head()
```

	_unit_id	message	market	category	lens	topic	inclusion	label	taxonomies_snacking	taxonomies_beverages	taxonomies_skincare	hashtags
4	2672625220	#ColbyJack #cheese is a marbled blend of #Colb...	US	Beverages	Ingredients	Annatto	annatto	0	[["Snacking","Ingredients","Annatto"], ["Snacki...	[["Beverages","Brands","Amp"], ["Beverages","Br...	[["Skincare","Themes","Blended"], ["Skincare","...	[#ColbyJack, #cheese, #Colby, #MontereyJack, #...
7	2672625374	The most important meal of the day! I've writt...	US	Beverages	Ingredients	Whole Grain	wholegrains	0	[["Snacking","Ingredients","Whole Grain"],["Sn...	[["Beverages","Themes","Whole Grain"], ["Bevera...	[["Skincare","Ingredients","Coffee"]]	[#breakfast, #wholegrains]

After collecting the hashtags of tweets and a word cloud was generated based on the frequency of the words in the hashtags. The purpose of this was to provide a more direct visualization of the most popular words and identify potential focus areas for the problem being addressed.



Based on the observations from the word cloud, a popular word list was generated. This list could be incorporated as a new feature in the existing model. There are two potential approaches to this:

- First, the popular words could be treated as binary columns and added as a second input to the model. Second, during the training process, more weight could be assigned to the popular words to give them greater importance in the classification task.

- The incorporation of the popular word list as a new feature has the potential to improve the accuracy and efficiency of the model. By identifying the most important words in the dataset, the model can focus on these areas and improve its ability to classify tweets accurately.

In conclusion, incorporating these features into existing models can lead to improved performance and more accurate classification results. What's more, the use of hashtags could provide additional insights into user behavior and preferences, which could be valuable for marketing and advertising purposes.

6. Model Implementation

6.1 Baseline Models

Four baseline models, logistic regression, support vector machines (SVM), Naive Bayes, and SGD (Stochastic Gradient Descent) classifiers were implemented using a similar pipeline. [iii]All models leverage the scikit-learn library for building the pipelines, which consist of a series of data preprocessing and modeling steps.

6.1.1 Logistic Regression

The logistic regression model implementation begins by splitting the dataset into training and testing sets, with a 70:30 ratio. A pipeline is then defined for the logistic regression model, which includes a `CountVectorizer` for converting the text data into numerical features. The logistic regression model is instantiated using the `LogisticRegression()` class from scikit-learn.

A parameter grid for hyperparameter tuning is specified, which includes the maximum number of features considered by the `CountVectorizer`, the inverse regularization strength `C`, the regularization type (`l1, l2, elasticnet, or none`), and the optimization algorithm used by the logistic regression model (`lbfgs`, `sag`, or `saga`).

The hyperparameters are tuned using a grid search with 5-fold cross-validation, and the best hyperparameters are selected based on their performance. The logistic regression model is then trained with the best hyperparameters, and its accuracy is computed on the test set.

6.1.2 Support Vector Machines (SVM)

For the SVM model implementation, the same training and testing sets are used. A pipeline is defined for the SVM model, which incorporates a `CountVectorizer` with a maximum of 5,000 features and an `ngram_range` of (1, 2). In addition to the `CountVectorizer`, a `TfidfTransformer` is applied to the features, weighting them using the Term Frequency-Inverse Document Frequency (TF-IDF) method.

6.1.3 Naïve Bayes and Stochastic Gradient Descent (SGD) Model

Two additional models, Multinomial Naïve Bayes and Stochastic Gradient Descent (SGD) with hinge loss (i.e., linear SVM), are implemented using a pipeline similar to the ones used for logistic regression and SVM. Both models employ the `CountVectorizer` and `TfidfTransformer` for text preprocessing.

The Multinomial Naïve Bayes model is instantiated using the `MultinomialNB()` class, while the SGD model with hinge loss is instantiated using the `SGDClassifier()` class. The SGD model is configured with an `L2` penalty, a learning rate (`alpha`) of 1e-3, and a maximum of 5 iterations.

6.1.4 Conclusion

In conclusion, four different models, namely logistic regression, support vector machines, Multinomial Naïve Bayes, and Stochastic Gradient Descent with hinge loss, were implemented and evaluated on the same dataset. These models employed scikit-learn pipelines with similar preprocessing steps and were compared based on their performance on the test set. This approach provides a comprehensive analysis of various models to identify the most suitable one for the given task.

6.2 BERT Methods

Based on Transformers and the pre-trained BERT model, we implemented a neural network with a Bert layer that generates word embeddings, and an underlying structure that combines BERT embeddings and categorical features engineered from the data, to output a binary class, where 1 indicates that the tweet is relevant to non-alcoholic beverages, and 0 otherwise.

Besides tokenizing and deleting irrelevant symbols, we also created features based on certain keywords in the text. For example, we created 10 columns including ‘`isalcohol`’, ‘`istea`’, ‘`isjuice`’, and ‘`iscoffee`,’ to indicate that the text contains words that are clearly relevant to these products. For `‘isalcohol,’` if the tweet contains keywords such as ‘vodka’, ‘whisky’, and ‘cocktail,’ it’s clearly relevant to alcohol. This helps the model focus on key information to determine the relevancy.

For the modeling part, we implemented both mean and max pooling strategies to turn word embeddings obtained from BERT into sentence embeddings, and concatenate the results. In this way, the model can not only capture strong features due to the effect of max pooling, but also smooth out noise with the help of mean pooling. After that, a fully connected hidden layer of size 256 is used with a dropout ratio of 0.5, to prevent overfitting and enhance the model's ability to generalize on new text data. Then, we concatenate the input categorical features with the hidden states from the previous layer, and add another fully connected hidden layer to process the combined information. Lastly, we define the output layer with sigmoid as its activation function, to output probabilities.

To enhance the model's performance, after a lot of trials, we implemented layer-wise optimizers/learning rates. BERT has already been pre-trained on a lot of text data, so we want to use a smaller learning rate for the BERT layer, to make sure that the learning process is consistent in the whole model. The architecture is shown below:

```
Model: "model"

______________________________________________________________________________________

-------
 Layer (type)                Output Shape         Param #     Connected to
=====================================================================================

============

 input_ids (InputLayer)      [(None, 180)]        0           []

 attention_mask (InputLayer)  [(None, 180)]       0           []

 tf_bert_model (TFBertModel)  TFBaseModelOutputWi  333579264  ['input_ids[0][0]',
                              thPoolingAndCrossAt
'attention_mask[0][0]']
                              tentions(last_hidde
                              n_state=(None, 180,
                               1024),
                               pooler_output=(Non
                              e, 1024),
                               past_key_values=No
```

```
                                  ne, attentions=None,
                                  cross_attentions=No
                                  ne)

 global_average_pooling1d (Glob  (None, 1024)        0
['tf_bert_model[0][25]']
 alAveragePooling1D)

 global_max_pooling1d (GlobalMa  (None, 1024)        0
['tf_bert_model[0][25]']
 xPooling1D)

 concatenate (Concatenate)      (None, 2048)        0
['global_average_pooling1d[0][0]'
                                                                ,
'global_max_pooling1d[0][0]']

 dense (Dense)                  (None, 256)         524544      ['concatenate[0][0]']

 dropout_73 (Dropout)           (None, 256)         0
['dense[0][0]']
 input_meta (InputLayer)        [(None, 10)]        0           []

 concatenate_1 (Concatenate)    (None, 266)         0           ['dropout_73[0][0]',
                                                                 'input_meta[0][0]']

 dense_1 (Dense)                (None, 128)         34176
['concatenate_1[0][0]']

 dense_2 (Dense)                (None, 1)           129         ['dense_1[0][0]']

==============================================================================
Total params: 334,138,113
Trainable params: 334,138,113
Non-trainable params: 0
______________________________________________________________________________
```

GPT (AutoML)

In an effort to explore alternatives of the BERT model, we employed another language model GPT and focused on its text classification function with the GPT-J transformer. This study aims to investigate the applicability of the GPT-J transformer, an open-source alternative to OpenAI's GPT-3, for text classification tasks. The objective is to identify alcoholic beverage-related tweets utilizing the Hugging Face pipeline and the pretrained weights of the GPT-J transformer model. To achieve this, we employ the `AutoModelForSequenceClassification` class, which extends the base pretrained model by adding a classification layer.

Initially, the dataset is upsampled to balance the classes and converted into the required format using the Dataset class. Subsequently, it is divided into training, validation, and test sets. The tokenizer from the transformers library is employed to convert the text data into numerical representations, which are then processed in batches. The tokenized datasets are further refined by renaming the "label" column to "labels" to meet the Trainer class's requirements.

To train the model and make predictions, the GPT-J transformer model is imported with its pretrained weights using the `from_pretrained()` method of the `AutoModelForSequenceClassification` class. The model's embedding size is resized to match the new tokenizer. For evaluating the model performance, we employ the accuracy metric, which is defined within a `compute_metrics()` function.

The `TrainingArguments` class is used to define training arguments, such as output directory, evaluation strategy, training, and evaluation batch sizes. The Trainer class, in conjunction with the training arguments, tokenized datasets, model, and `compute_metrics` function, facilitates model training and evaluation.

7. Model Evaluation

Evaluation Metrics	**Models**	**Baseline**				**Transformer-Based**	
		Naïve Bayes	Logistic	SVM	SGD	BERT	GPT
Accuracy (Leading Metric)		0.66	0.75	0.74	0.61	0.77	0.65
Precision		0.72	0.74	0.73	0.68	0.8	-
Recall		0.66	0.75	0.74	0.61	0.82	-
F-1 Score		0.59	0.74	0.73	0.46	0.81	-

The accuracy scores reveal that the BERT model outperforms all other models with an accuracy of 0.77. The logistic regression model achieves the second-highest accuracy of 0.75, closely followed by SVM (0.74) and Naïve Bayes (0.66). The baseline SGD model has the lowest accuracy of 0.61.

While the table provides a clear comparison of the models' performances in terms of accuracy, precision, recall, and F-1 score, it is essential to consider additional factors such as robustness, scalability, and training cost when selecting an appropriate model for the text classification task.

The BERT model demonstrates superior performance in terms of the key evaluation metrics. However, it is important to acknowledge that transformer-based models like BERT and GPT tend to have higher computational requirements compared to traditional machine learning models. This might lead to longer training times and increased hardware costs, which could be a concern for large-scale deployments or when working with limited resources.

On the other hand, traditional models like logistic regression and Naïve Bayes are relatively simpler and computationally less demanding. They can be trained and deployed more quickly, making them suitable for projects with limited computational resources or tight deadlines. Additionally, these models can be more easily interpreted, which can be advantageous when the goal is to understand the underlying relationships between features and the target variable.

SVM and SGD models offer unique benefits. SVMs are known for their robustness in handling high-dimensional data and can be effective in cases where the dataset has a complex decision boundary. The SGD model, although having the lowest performance in terms of the F-1 score, is highly scalable and can be effectively used for large-scale, online learning scenarios due to its incremental learning capability.

8. Model Interpretability

8.1 Method 1: Local Interpretable Model-Agnostic Explanations (LIME)[iv]

LIME is useful in learning about local interpretability of the model, i.e. helps in understanding the decision-making behind a single prediction. In the examples above, we can see which words in the message contribute most towards the predicted class. We can see that words like “tea” and “detox” are essential in predicting the message as Relevant to a Non-Alcoholic Beverage and words like “gin” are important when predicting a message as Not Relevant. In the last two cases we see how the presence of words like “milk”/ “beer” lead to misclassification.

8.2 Method 2: SHapley Additive exPlanations (SHAP)[v]

SHAP is a unified measure of feature importance, grounded in cooperative game theory. It assigns each feature an importance value by considering all possible feature combinations and contributions to the model's output. SHAP values provide fair and consistent attributions, ensuring that the sum of feature contributions equals the difference between the model's prediction and its expected value.

In our team’s pipeline, a logistic regression model is trained and evaluated using a dataset partitioned into training and test sets. After assessing the model's performance, SHAP values are calculated to interpret its predictions. To expedite the computation, both the training and test sets are sampled using the shap.sample function. A KernelExplainer object is then created, which leverages the logistic regression model and the sampled training data. The object computes SHAP values for the sampled test data using its shap_values method.

The pipeline concludes with two visualizations: a summary plot and a force plot. The summary plot offers a comprehensive view of feature importances, illustrating the average impact of each feature on the model's output. We can see that words like "coffee", "water", "juice", and "tea" contribute the most to feature importance, indicating their high frequency among the text.

9. Model Deployment and Client Presentation

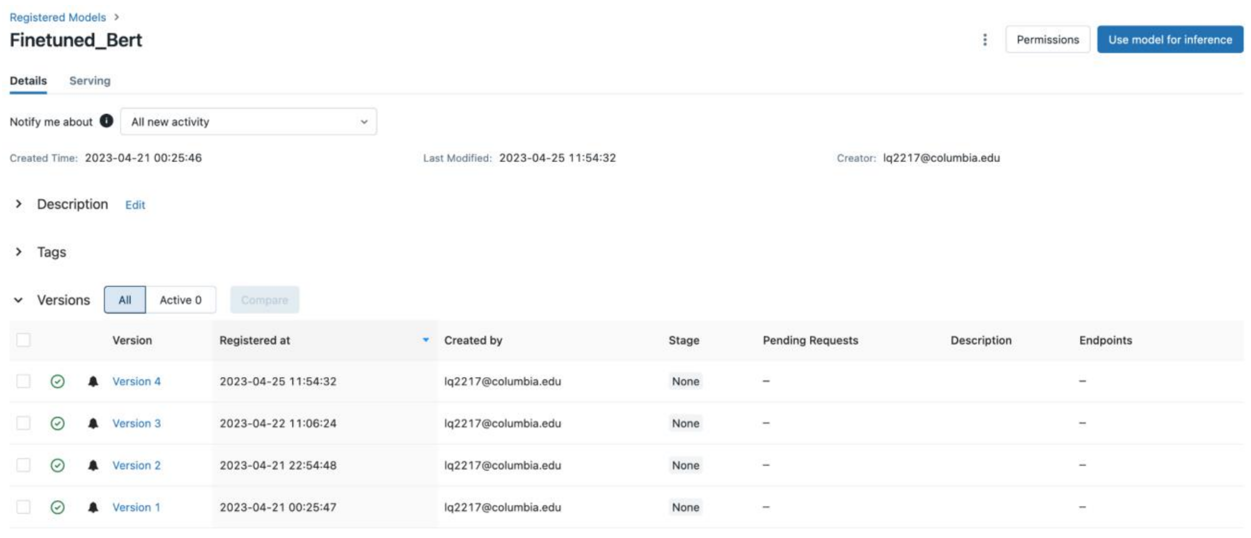
9.1 CometML Model Experimentation "Light MLOps" Implementation

Using CometML, we are able to collect all of the locally run experiments into a central hub where we are able to compare and contrast the various models that we tested. This is meant to consolidate all of our efforts into a cohesive environment and increase clarity of communication between data scientists and stakeholders.

The key advantage of CometML automation is that it is low-code and highly adaptable to new experiments. Data scientists need less than 10 lines of code to fully implement CometML and record their key model artifacts and results into a centralized visualization database. We worked with BlackSwan PMs to identify best practices for data science project implementation and liaised with CometML representatives to assist us in CometML's implementation.

Together with Streamlit, this allowed us to construct a lightweight MLOps pipeline, separating, model dev (local), model stage (CometML), and model production (GitHub+Streamlit) environments; mimicking real-world machine learning engineering projects by productionizing the our models by exposing an interface with the outside world.

Using mlflow, we deployed the finetuned BERT model on Databricks, and a demo with streamlit, so that we can enter any text to test the model:



Registered Models >

Finetuned_Bert

Notify me about: All new activity

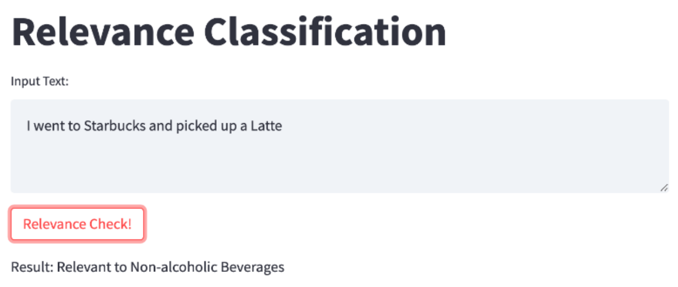
Created Time: 2023-04-21 00:25:46 Last Modified: 2023-04-25 11:54:32 Creator: lq2217@columbia.edu

> Description Edit

> Tags

Versions All Active 0 Compare

Version	Registered at	Created by	Stage	Pending Requests	Description	Endpoints
Version 4	2023-04-25 11:54:32	lq2217@columbia.edu	None	–		–
Version 3	2023-04-22 11:06:24	lq2217@columbia.edu	None	–		–
Version 2	2023-04-21 22:54:48	lq2217@columbia.edu	None	–		–
Version 1	2023-04-21 00:25:47	lq2217@columbia.edu	None	–		–



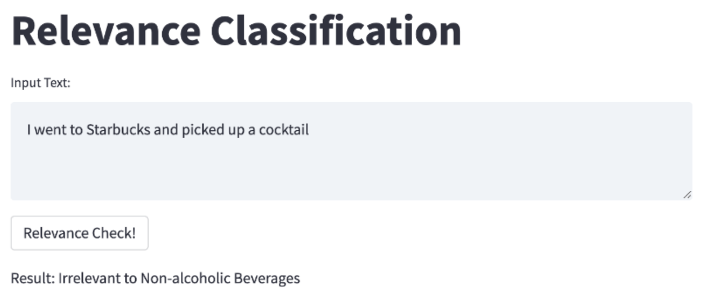
Relevance Classification

Input Text:

I went to Starbucks and picked up a Latte

Relevance Check!

Result: Relevant to Non-alcoholic Beverages



Relevance Classification

Input Text:

I went to Starbucks and picked up a cocktail

Relevance Check!

Result: Irrelevant to Non-alcoholic Beverages

The models can then be presented to future clients of BlackSwan Data and other non-technical stakeholders with ease. Users can play around with this web-based user interface to approach the key applications of our modeling process.

10. Appendix

[i] **Definition of Relevance:** We consider matches for topics in Appendix 2 to be relevant for this specific dataset when the topic in the given context describes the consumption of a non-alcoholic beverage.

[ii] **Definition of non-alcoholic beverages:** A non-alcoholic drink is defined as any drink which does not contain alcohol, such as water, sodas, coffees, teas, etc.
Additionally, a non-alcoholic drink should not be associated with consuming alcohol, therefore we exclude alcohol-free versions of drinks which are originally made with alcohol such as mocktails or beers, or any mixers intended for use with an alcoholic beverage.
The Beverages dataset captures conversations around the consumption of non-alcoholic drinks and non-alcoholic drink brands. Since the last time our categories were revised, we have now 17 different categories in our taxonomy to group all our products.
Cocoa Drinks (cocoa-based drinks or chocolate drinks), **Coffee**, **Concentrated drinks** (those which are designed to be diluted), **CSD** (Carbonated flavored beverages), **Dairy Alternative, Drinks** (Beverages used as dairy milk replacement), **Drink Enhancers** (Flavor additives designed to be added to water), **Energy & Sport Drinks**, **Fermented Drinks**, **Flavored Drinks**, **Functional Drinks** (Designed to fulfill a certain function), **Juices**, **Milk Based Drinks**, **Other Beverages** (Unique drinks that don't fit in other categories), **Protein Shakes**, **Smoothies**, **Tea**, **Water**

What is not included in the Non-alcoholic drinks dataset?
Supermarket own brands Products such as 'Carrefour Cola' or 'ASDA Extra Special Coffee' are not included but the conversations are captured by the product lens.
Baby Food and Formulas E.g. Baby Milk.
Snacking E.g. Yoghurt.
Coffee Capsules As "capsules" are only a different way of presenting a product and not a different one itself, this would be part of the Themes list

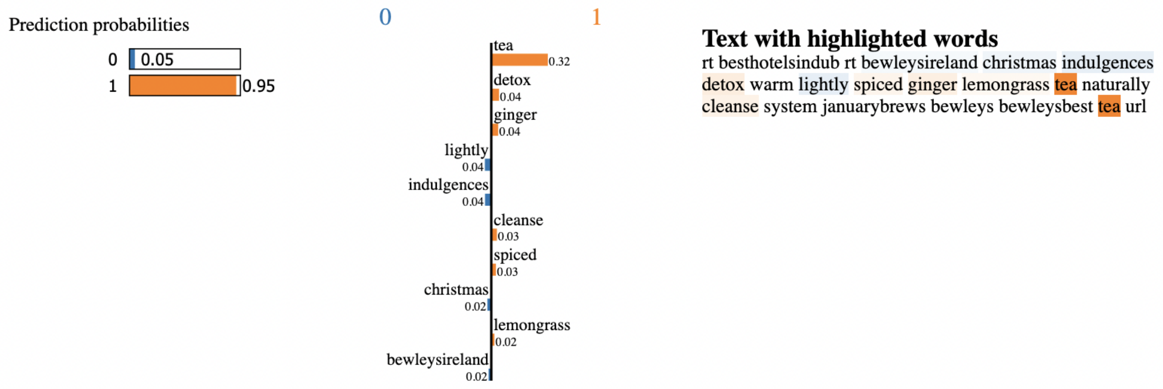
iii Model Pipeline:

```
# Define a pipeline for the logistic regression model
pipeline_clf = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('clf', LogisticRegression())
])

# Define a parameter grid for hyperparameter tuning
param_grid = {
    'vectorizer__max_features': [500, 1000, 5000],
    'clf__C': [0.1, 1, 10],
    'clf__penalty': ['l1','l2','elasticnet',None],
    'clf__solver': ['lbfgs','sag','saga']
}

# Perform grid search to find the best hyperparameters
grid_search_clf = GridSearchCV(pipeline_clf, param_grid, cv=5,
verbose=1, n_jobs=-1)
grid_search_clf.fit(X_train, y_train)

# Print the best hyperparameters and the corresponding accuracy
print('Best parameters:', grid_search_clf.best_params_)
y_pred_clf = grid_search_clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred_clf)
```

[iv] Model Interpretability with LIME and SHAP (*Credit: Hridaya Sanjayb Shinde*)

Correct Prediction:

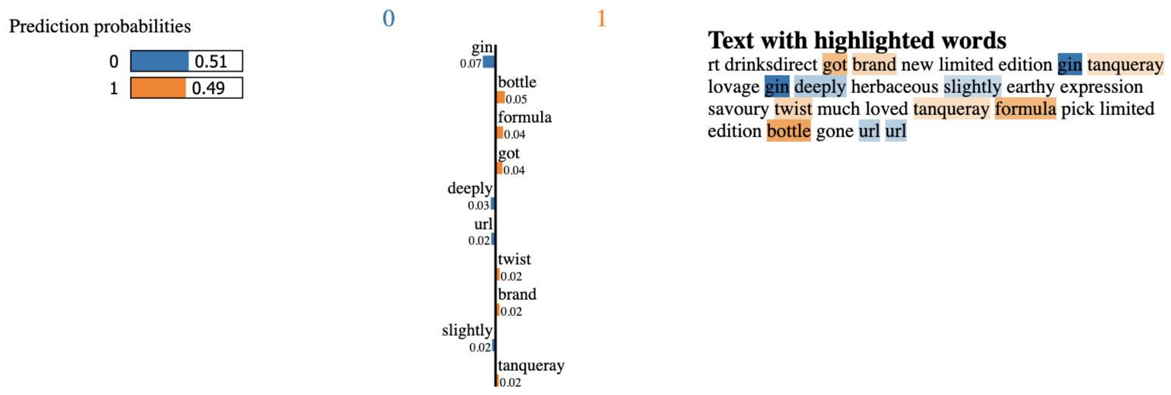
```
Document id: 1
True class: Relevant to Non-Alcoholic Beverage
Predicted class: Relevant to Non-Alcoholic Beverage
```



Text with highlighted words

rt besthotelsindub rt bewleysireland christmas indulgences detox warm lightly spiced ginger lemongrass tea naturally cleanse system januarybrews bewleys bewleysbest tea url

```
Document id: 5
True class: Not Relevant
Predicted class: Not Relevant
```

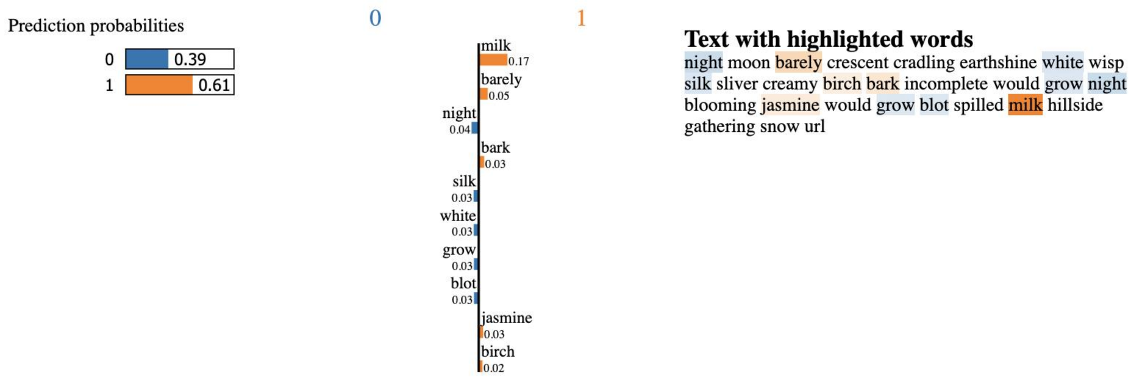


Text with highlighted words

rt drinksdirect got brand new limited edition gin tanqueray lovage gin deeply herbaceous slightly earthy expression savoury twist much loved tanqueray formula pick limited edition bottle gone url url

Incorrect Predictions:

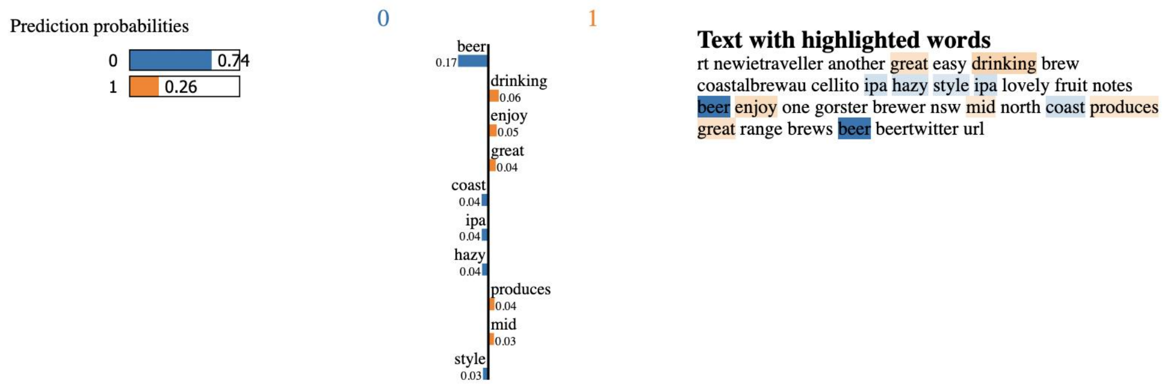
```
Document id: 19
True class: Not Relevant
Predicted class: Relevant to Non-Alcoholic Beverage
```



Text with highlighted words

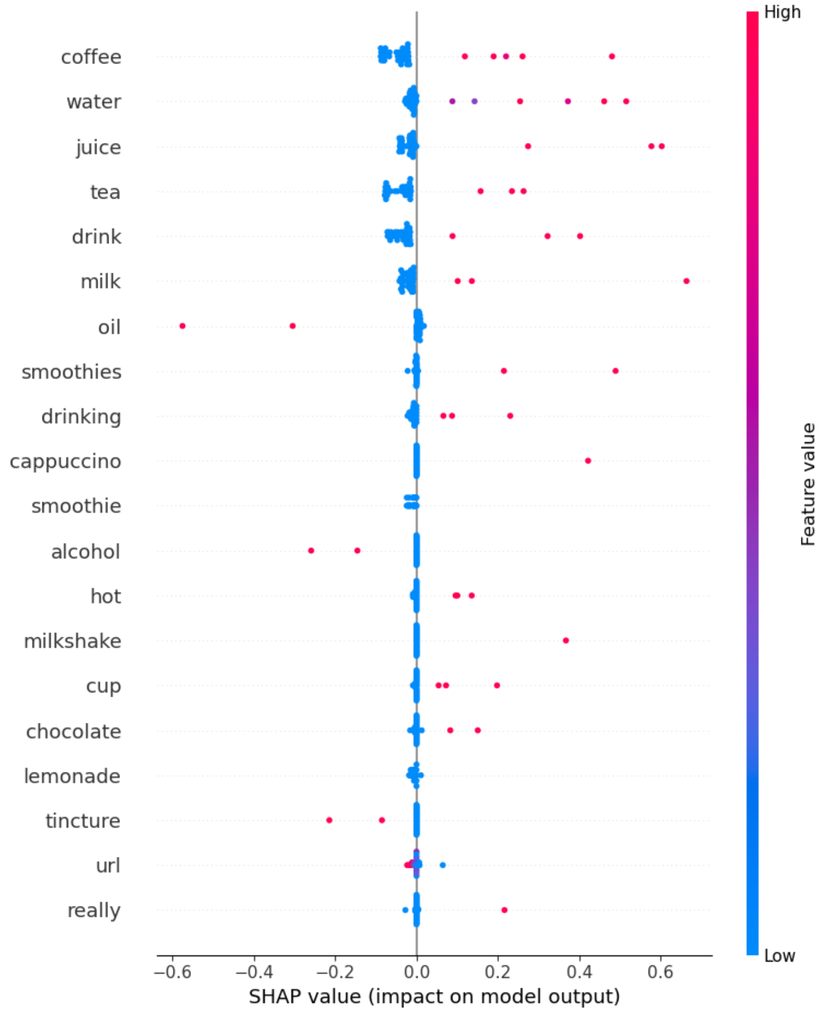
night moon barely crescent cradling earthshine white wisp silk sliver creamy birch bark incomplete would grow night blooming jasmine would grow blot spilled milk hillside gathering snow url

```
Document id: 51
True class: Relevant to Non-Alcoholic Beverage
Predicted class: Not Relevant
```



Text with highlighted words
rt newietraveller another great easy drinking brew coastalbrewau cellito ipa hazy style ipa lovely fruit notes beer enjoy one gorster brewer nsw mid north coast produces great range brews beer beertwitter url

ᵛ SHAP


coffee
water
juice
tea
drink
milk
oil
smoothies
drinking
cappuccino
smoothie
alcohol
hot
milkshake
cup
chocolate
lemonade
tincture
url
really
SHAP value (impact on model output)
High
Feature value
Low