# Rodrigo Rodriguez

07/16/2024

Applied Data Science

DSC 680 (SUMMER)

**Project Whitepaper:**

**Automating Online Cookies Classification to Mitigate Legal Risk**

# Business Problem­

The legal team requires that all online web cookies be classified as Essential, Functional, Performance, or Targeting. Thousands of online cookies are currently live across our various domains. Categorizing each one manually one by one is a significant time-intensive effort. Given the potential risk of leaving the cookies uncategorized, the Legal team is looking for a solution that can be deployed as soon as possible. It has taken an analyst roughly about two weeks to categorize about 100 cookies. With this time frame it is unfeasible to manually complete this task within the allocated two-month window. Leveraging a ML model will speed up the process of categorizing these cookies and in doing so, will mitigate WM’s exposure to legal risk.

# Background and History

Online cookies are used to track certain meta-data left behind when a website visitor takes an action on a web page. These actions can be as simple as loading a page or click a link, or as complex as triggering API calls to check login eligibility. Most cookies are essential for the website to work, like the before mentioned login example, however other types of cookies are used to collect different types of data points. Cookies that may cause the website to break if turned off are categorized as “Essential” and must remain on. Cookies that track performance metrics such as how many times a certain page is viewed are called “Performance/Analytics” and cookies related to non-critical website functionality are classified as “Functional”. Lastly, the most sensitive type of cookies are the ones that track user behavior and follow users around are classified as “Targeting” and are often used for advertising purposes (Komnenic 2022).

The company­ is Fortune 200 company that predominantly receives new business orders offline through the call center. However, the company continues to push the percentage of total orders towards the online channel. With the recent laws passed regarding consumer data privacy, such as CCPA, the legal and cyber-security team now heavily focuses on identifying and correctly categorizing online cookies.

# Data Explanation

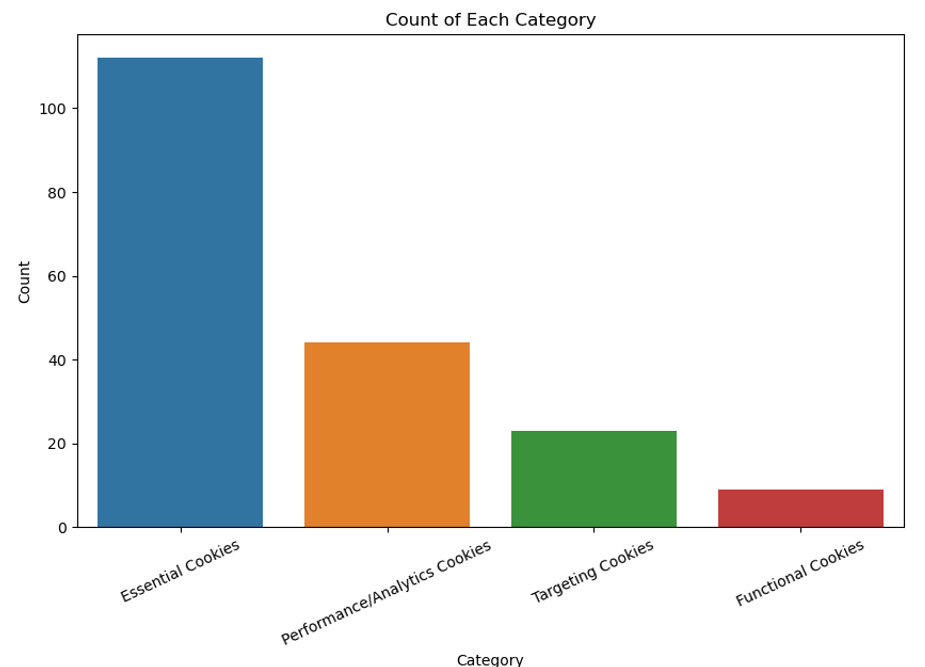
The data is unstructured Natural Language Text presented in a structured way. The data enables us to conduct supervised training for modeling. The feature variable 'Description' can be used to predict the target variable 'CategoryID.' The dataset was obtained internally from the company. Cookie category IDs were manually assigned by an analyst based on cookie description. Training data will be the current 100+ cookies that the analyst has already categorized.

The dataset contains several categorical variables with the target variable being “CategoryID”, this feature indicates what category the cookie will be placed in. Most variables do not add predictive value to the model and will have to be removed during data preparation. Once the non-predictor variables are removed, we are left with only three variables: “CategoryID”, “C\_Source”, and “Description”.

# Methods

During EDA the distribution of different key variables were plotted. The primary finding was the presence of class imbalances in the “Category” column as seen in Figure (1). To ensure adequate representation in training and test sets, stratified sampling was implemented.

This includes dropping any non-predictor variables to remain with the three key variables listed above under Data Explanation. Then dummy variables will need to be created from the “C\_Source” categorical variable. Lastly, the natural text data in the “Description” variable will need to be processed and the dataset will have to undergo text vectorization.

**** *Figure 1. Frequency of Cookie Categories*

After EDA the data must be prepped before processing. Data preparation consisted mainly of typical steps associated with NLP classification problems. Beginning with first dropping any variables that do not add any predictive value to the model. For example, any agnostic contextual data variables or identifiers are not relevant for our purposes since it does not affect cookie description or classification. Figure 2 includes a code snippet of the variables that were dropped and the remaining variables in the dataset.

*A screenshot of a computer

Description automatically generated  
Figure 2. Dropped Variables, and Remaining Variables*

As seen in Figure (2), the ‘C\_Source’ holds predictive value and is a classification variable that identifies the source of the cookie. This variable has string values that must be converted to numerical. I do this by using one-hot encoding to create dummy variables. After the dummy variables are created, we can proceed with Natural Language Processing steps.

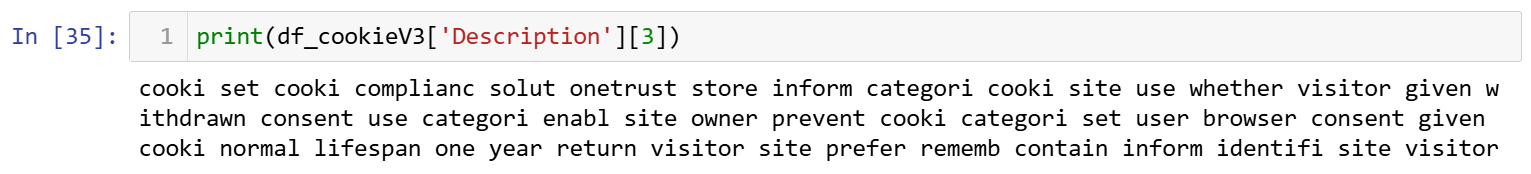
The steps required to process natural language text are as follows:

1. Remove HTML (if applicable).
2. Convert all text to lower case.
3. Remove punctuation and special characters.
4. Remove stop words.
5. Apply stemmer.

Figure (3) shows an example of a cookie description before applying any of the steps listed above. For comparison, figure (4) shows the same example of cookie description but now after applying the five steps listed above.

*A screenshot of a computer

Description automatically generated  
Figure 3. Unprocessed Cookie Description*

*  
Figure 4. Processed Cookie Description*

The last processing step associated with NLP is transforming the data using a vectorizer. However, this step is done in the modeling phase after splitting the data into training and test sets to avoid data leakage and ensure the model generalizes well to new unseen data.

# Analysis

The modeling phase also has a set of steps that must be followed. While different projects and problems will vary in steps, the steps I conducted ensure that the models are properly prepared and evaluated in a timely manner.

During the modeling phase we perform the following steps to obtain a model’s performance.

1. Split data into training and test sets.
2. Transform using TF-IDF.
3. Concatenate one-hot encoded features.
4. Train and evaluate models.

Given the imbalance in the dataset, specifically with a high volume of 'CategoryID' 1 values and low volume of 'CategoryID' 2 & 3 values, it's important to ensure that each category is adequately represented in both the training and test sets. To achieve this, stratified sampling was applied. This will ensure that the data is divided in a way that the proportion of each category is maintained in both training and test sets.

The modeling phase can begin after the steps listed above are completed. We decided to test three separate classification models that are common and simple to implement. The models to test were picked based on models that gives the best results without having to adjust hyper parameters. The three models used were:

* Logistic Regression (OvR)
* Random Forrest Classifier
* Naïve Bayes Multinomial

After training and testing each model a classification report and confusion matrix were used to evaluate performance. The classification reports and confusion matrix were very similar for all models. Figure (x) shows a report and confusion matrix for the Logistic Regression. The class imbalances can be seen in the low f1-score for class 2 and class 4.

A screenshot of a computer screen

Description automatically generated  
*Figure 5. Logistic Regression Classification Report*

# Conclusion

All models exhibited the same relationship with macro F1 and weighted F1. The macro F1 was lower at about .72 and the weighted F1 was higher at about .84. The performance drastically drops for the minority classes (Class 2, and 4). The results of this project look promising. An accuracy above 85% and high F1 scores for classes with adequate representation in the data prove that the model is capable of correctly classifying an online cookie based on its description. However, no model should be deployed as is as performance can be improved by addressing class imbalances.

# Assumptions

This project assumes that the descriptions for each cookie are adequate and true. Initially, the descriptions were collected manually by researching the function and use of each cookie. Moving forward, a cookie database API will be used to retrieve cookie descriptions based on cookie names. Since the model leverages the cookie description for classification, we must assume that the descriptions are accurate explanations of the cookie’s purpose and function.

# Data Limitations

This data is limited in the number of records available for training and testing. A manual effort was made to collect the supervised learning dataset used for this project. However, the model could improve performance by having access to a more robust dataset; preferably one not affected by class imbalances.

# Challenges

The main challenges for this project were the limited dataset, class imbalances, and short project time period. As listed under Data Limitations, the class imbalances negatively impacted the model’s performance. Additionally, if there was more time to work on the model hyper parameter tuning could be done to further improve model performance.

# Future Use and Recommendations

This model can be scaled and used for all web cookies across all of WM owned domains. Alternatively, this project can be leveraged by other companies looking to classify their online web cookies. My recommendation is to move forward with optimizing the Logistic Regression model over Naïve Bayes and Random Forest. The Logistic Regression model outperformed Random Forest, and can be further improved by tuning hyper parameters, while the Naïve Bayes is limited in this aspect (Yiu 2019).

Additional optimization options are oversampling the minority classes or apply class weighting. These are common techniques to apply when the data exhibits significant class imbalances, as this dataset does. Although these are my primary recommendations other options could include providing a more balanced dataset to the model (i.e. manually collected a more balanced data set to feed the model.)

# Implementation Plan

After addressing the challenges and implementing recommendations, the model can be implemented by feeding new data from the cookie database API. The data will be fed to the model for classification and a random sampling can be taken from the model’s results to validate manually. If manual validation is not passed, then the model will have to be further optimized.

# Ethical Assessment

We are relying on the cookie descriptions provided by API to be accurate and true to properly assign cookie categories. Continuous cookie management will have to take place to monitor and ensure that cookie functions are not changed by the cookie hosts and that cookies are correctly classified.

# Sources

Yiu, T. (2019, October 24). *Understanding The Naive Bayes Classifier*. Towards Data Science. Retrieved May 24, 2024, from https://towardsdatascience.com/understanding-the-naive-bayes-classifier

Pramoditha, R. (2023, November 29). *Logistic Regression for Multiclass Classification — 3 Strategies You Need to Know*. Medium. Retrieved May 24, 2024, from https://rukshanpramoditha.medium.com/logistic-regression-for-multiclass-classification-3-strategies-you-need-to-know-0a3e74574b96

Komnenic, M. (2022, January 14). *The Different Types of Internet Cookies Explained*. Termly.io. Retrieved May 24, 2024, from https://termly.io/resources/articles/types-of-internet-cookies/