Zorus URL Classification Engine

John G, Anant K, HuanHuan L, Spandana L, Akshay P, Visaga R, Yiliang R

University of Connecticut

Advanced Business Analytics and Project Management - OPIM 5770

Professor Jennifer Eigo

12/10/2019

Table of Contents

1. **Executive Summary**

The internet is one of the biggest sources of information of websites with a variety of content. This content is often categorized and organized for consumption based on requirement. One such application is a custom classifier required by Zorus to provide flexibility to further develop Zorus' product and to align with their current technology stack. A URL classification model was built that can accurately classify URLs.

The URL Classifier has the ability to classify URLs based on the website content data (text and images) and is aimed to help negotiate the existing provider eventually phasing out third party after the classifier gets mature enough. Apart from the basic data pre-processing, deadlinks were excluded as linkrot and a balanced dataset was created for analysis. URL classifier accurately classifies URLs with higher accuracy in predicting "Adult" category since this is a major concern for Zorus.

2. **Problem Statement**

The content classification engine provided by Zorus' third party is a critical part of Zorus offering and is a major risk to its business model. MSPs struggle to provide cost effective and capable solutions. Since the margin of error (misclassification) in classifying URLs could hinder Zorus Product credibility, Zorus needs a URL classification model that can accurately classify given URLs into one of the 15 categories.

3. **Dataset Introduction**

The dataset is related to website classification (DMOZ dataset). It has three columns which are ID, URL and Category. URL is the predictor variable and Category is the target variable. It has

1.5 million URLs with 15 different categories. The dataset is highly imbalanced. Few of the URLs are repeated with different categories. Many of the URLs were inaccessible.

4. <u>Data Exploration</u>

4.1.Distribution of URLs per category

In the given imbalanced dataset as shown in Appendix Figure 1. Distribution of URLs per category, most of the URLs belongs to Arts, Society and Business category. The bottom three categories contributed to less than 5% of the dataset: 0.6% in News category, 1.8% in Home category ,and 2.3% in Adult category.

4.2.Percentage of English words per category

URLs with Kids category had 100% of English words and URLs with Science and Sports category had lowest ~63 % of English words (as shown in Appendix Figure 2. Percentage of English words per category).

4.3.Average number of images per category

As shown in Appendix Figure 3. Average number of images per category, URLs with Adult category have the highest average number of images and URLs with Science category had the lowest number of average images.

5. Methodology

The dataset was highly imbalanced. To build the machine learning model, a balanced dataset was created with equal number of entries per category. Title, description, body was extracted from web-page content for analysis. These were converted into meaningful tokens to predict the category of URLs.

The assumptions made during analysis are as follows: 1. The classification of URL is single-label, subjective (not functional or sentimental). 2. No additional categories for URLs, (except the provided 15 and Linkrot) exists. 3. Efficient processing for over 1.5 million records is possible in Python programming language

5.1 Data Preprocessing

5.1.1. Linkrot exclusion & creation of balanced dataset

Data exploration results showed there were certain number of broken links, which could negatively affect the user experience. As shown in appendix Figure 4. HTTP status code, linkrot checking program was run on entire dataset, which detected all the inaccessible links like 301, 302, 404, 500. There were about 250k broken links and they were tagged as “Linkrot” and were excluded from analysis (except code 202 which was retained as a valid link) thus, retaining 1.3 million records for further analysis.

The dataset was highly imbalanced. Nearly 10k records per category were used to build the balanced dataset. To build a logistic regression model, top 2500 most frequent words per category were used. To build CNN model, 1k images from adult category URLs and 1k images from other categories of URLs were downloaded and used in training the model.

5.1.2. Web scraping and text parsing

Each website content was downloaded by sending a GET request using Python library Urlib. The timeout for GET request was set to 15 seconds and the website was skipped if it did not respond to the GET request. Website response consists of HTML code where necessary text was stored. Text parsing consists of a) Parse text into beautiful soup object for filtering of text and manipulation. b) CSS and JavaScript content were filtered out. c) Using Python library beautiful soup HTML tags were excluded d) Symbols were removed from text. e) Every word was joined to text string by a new line.

5.1.3. Text tokenization

As shown in appendix Figure 5. Illustration of pre-processing, parsed text of website was split up into tokens using regular expression word tokenizer. Regular expression was passed to NLTK python library RegexpTokenizer function and it converted plain website text into tokens. When text was tokenized, each word was converted to lowercase as upper and lowercase have different ASCII codes.

5.1.4. Translation

There were many texts with words in Chinese and other languages. Using Google API translator, all the non-English words were converted to corresponding English words

5.1.5. Removing stop words and normalizing text

Tokens of website text consists of a lot of words and not all of them are useful for building an ML model. Tokens could consist of stop words, random words, singular and plural words. Such words consumes a high volume of data so to reduce this volume, text normalization operations such as removing stop words and word lemmatization were implemented. English language had 180 stop words. For analysis, stopwords.words('english') function which consists of 179 commonly used English language stop words were used. Words with length less than 3 is also considered as stop words because there are not many words that contains meaningful information in English language. The symbols which are in range of ASCII table of: [32:64],[91:96],[123:127] are considered as stop words. Integer numbers from range gets generated and they were converted to char symbols which are stored in list. When stop words list is generated then it is possible to filter out stop words from token.

Words lemmatization was done using English words dictionary of singular and plural forms. NLTK python library WordNetLemmatizer function was used which takes a word as an argument and returns a singular form of word. If input words was in a singular form then output would be an input word in a singular form and if the input words was in a plural form then output would be an input words in singular form.

5.1.6. Removing word stemming

There were a lot of words which have the same meaning but had various forms. Such inflectional forms of words were reduced to a common base or root word. For example, consult, consultant, consulting, consultation to consult.

5.2 Models

5.2.1. Words frequency model

Word frequency model would generate the most frequent words which will be used for feature generation of Machine learning model. This model created a list of top 2500 frequent words for each category. All URLs were classified into 15 categories. Each category's feature can be defined by the words. Humans could recognize intuitively by the set of words, for example, a set of words ['health', 'treatment', 'cancer', …] implies that it more likely belong to "Health" category.

We had the word list of tokens generated from all websites content through web-scraping and pre-processing. Frequency of each word per category was calculated and were later sorted in ascending order. The frequency of words were calculated by using nltk.FreqDis package.
Since this process is generating a list of most frequent words, only the words within 2500 most frequent words are extracted and saved in the list. For example, as shown in appendix Figure 6. Most frequent words per category, the top 5 most frequent words in "Science" category is ['research', 'science', 'new', 'information', 'contact'].

5.2.2. Machine learning model (Logistic Regression)

Machine learning algorithms are trained and tested by using the top 2500 most frequent words list per category. The main goal was to create a model to predict a website category without human interaction. This is done by processing 2 steps:

1. Features and labels creation
2. Machine Learning model training

Machine Learning requires a specific form of data as input. The data is already set up properly by creating the most frequency words list. Features are the shared attributes among the tokens of website content. If a word in the website token list is in the most frequent list for the specific category, the position of this word is changed 1, otherwise, it is 0. Labels are the target output categories of URLs. Since the dataset contains 15 categories so the labels are used to track them properly then the machine learning models would recognize the target categories

The input data set of 10k per category is partitioned into 70% as training data and 30% as test data by using sklearn.model_selection function. The training data is used to train the machine learning models, and the test data is an indicator of how well the models are performing.

Logistic Regression, Decision Tree and SVM(Support Vector Machine) are the commonly used built-in machine learning model in scikit-learn library for Python. After comparing the results of these three models, Logistic Regression is the best model because it had the highest F1 score of 85% on Adult category and overall F1 score of 82%. It is a linear classifier methods to make a classification decision based on the values of a linear combination of the input features.

There are several scores to evaluate the accuracy of machine learning models :

1. Precision indicates how many positives predictions have been predicted of total predictions. In this project, it is the ratio of number of URLs correctly categorized to the total number of URLs categorized.
2. Recall score is the number of true positives divided by the number of true positives plus the number of false negatives. In this project, it is the ratio of number of URLs correctly categorized to the actual number of URLs.
3. F1 score is a weighted average of the precision and recall of a model, which is commonly used to evaluate the performance of classification models.

As shown in Appendix Figure 10. Performance matrix for logistic regression model, Logistic regression model performed best and predict on 82% accuracy (overall F1-score). The best results are performed on “Home” category where model predict of 99% accuracy (F1-score). The least accurate category is on “News” where model predict of 77% accuracy (F1-score). The model predict of 85% accuracy (F1-score) on “Adult” category.

5.2.3. Deep learning model (CNN)

Since the margin of error in classifying URLs would hinder Zorus product credibility, the misclassification rate should be reduced as much as possible. Image classification algorithm, powered by Convolutional Neural Network is a core research subject for many industries to make up the shortcomings of traditional algorithms. CNN model recognizes the patterns within pre-labeled input images and learns which category the test images belong to.

Web scraping have extracted images from the websites. Based on 1000 adult images and 1000 non-adult images, we build a CNN model to improve the accuracy of “Adult” category. The images are partitioned into 70% training and 30% validation. The output of model is a particular image belongs to “Adult” category or not. According to the results on test dataset, The CNN performed an 80% accuracy on image classification.

5.2.4. Model applications & performance

Logistics regression model was the best model and CNN was well trained to reduce the misclassification rate. As shown in Appendix Figure 7. Proposed implementation flow chart of model, the sequence of operations taking place with the model are as follows:-

1. The input URLs are stored in Zorus database after classification.
2. For new URLs which are not existing in Zorus database, web scraper extracts the content and image from the websites. The website content is preprocessed to tokens.

3. Logistic regression classifies the URLs to 15 categories based on the tokens.
4. CNN model process all the URLs which are not predict as “Adults” and classify them as “Adult” or “Non-adult” categories. “Adult” URLs are sent to database and “Non-adult” URLs retains the category which was predicted predicting by Logistic regression in previous step and it is sent to the database.

As shown in appendix Figure 8. Performance of Models, Logistic regression has 85% accuracy on “Adult” category. The misclassification rate is 15% composed of 10.30% False Negative and 4.70% False Positive. 10.30% of False Negative, which means the URLs are predicted as “Non-adult” but they are actually “Adult”, are processing through CNN model. With 80% accuracy of image classification, False Negative decreases by 8.24% (10.3% * 80% = 8.24%) , therefore, misclassification rate decreases by 8.24% correspondingly. In other words, after applying CNN model, the misclassification rate for “Adult” category decreases from 15% to 6.76%.

6. Conclusions

In conclusion, an ensemble model consisting of the logistic regression along with the CNN for image classification provides an improved accuracy for the Adult category to 93%. Since this was a major concern for Zorus and their client; so as to not let an Adult website pass the firewall. With a run time of 4 minutes for the logistic regression and 2 minutes for the CNN model on the test data; the model will have more correct predictions in the future when compared to incorrect.

Increase in the classified URLs will further build model’s memory, thereby providing higher accuracy results in the future. Not only the model is cost effective but it also provides an added filter for the Adult category with the CNN model to prevent any Adult URLs from being classified as non-Adult.

7. Recommendations

Below are proposed two implementation recommendations:

1. URL prediction as a service: The two models used in the project, namely Logistic Regression and CNN can be encapsulated or wrapped in a web-service as an easy consumption service. Using a familiar python library scikit-learn the model can be trained. The encapsulation would get initiated by storing the trained model in a serialized format as a pickle file with extension *.pkl. An interface to the trained model would need to built using a call .prediction() on the model. Furthermore, turning the .prediction() interface into a web service using the python micro-web server framework Flask. Flask's support for REST API web services should allow for minimal configuration.

2. High Level implementation approach: In order to facilitate the usability of the model, a simple flowchart depicting AWS components that would allow for a scalable implementation while taking advantage of AWS elastic services and compute. Refer Figure 9. Implementation of model flowchart through AWS. The entry point for URLs would be using Fargate as a Web Server to query a REDIS elastic search. REDIS would be a cached repository of classified URLs. Once a result of the model is generated, control can be passed to the Zorus URL exception handling to override the prediction if need be. The override transactions should be stored and used to continuously train the model so future overwritten URLs follow the rules setup by Zorus and not those of the classification model. RDBMS could be used for storing the data, and as the data gets larger in size, AWS allows for various big data services, like EMR or Redshift.

8. References

Badr, W. (2019, 02 02). *Having an Imbalanced Dataset? Here Is How You Can Fix It.* From https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb

Bonner, A. (2019, 2 02). *The Complete Beginner's Guide to Deep Learning: Convolutional Neural Networks and Image Classification*. From https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb

Chauhan, N. S. (2019, 3 11). *Real world implementation of Logistic Regression*. From https://towardsdatascience.com/real-world-implementation-of-logistic-regression-5136cefb8125

Dave, R. (2018, 07 30). *Industrial Classification of Websites by Machine Learning with hands-on Python*. From https://towardsdatascience.com/industrial-classification-of-websites-by-machine-learning-with-hands-on-python-3761b1b530f1

Kho, J. (2018, 9 26). *How to Web Scrape with Python in 4 Minutes*. From https://towardsdatascience.com/how-to-web-scrape-with-python-in-4-minutes-bc49186a8460

Meyers, D. P. (2011, 5 31). *An SEO's Guide to HTTP Status Codes (An Infographic)*. From https://moz.com/blog/an-seos-guide-to-http-status-codes

Raza, N. (2019). *Cyber Security for the Channel*. From https://zorustech.com/

Shung, K. P. (2018, 3 15). *Accuracy, Precision, Recall or F1?* From https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

9. Appendix

Figure 1. Distribution of URLs per category

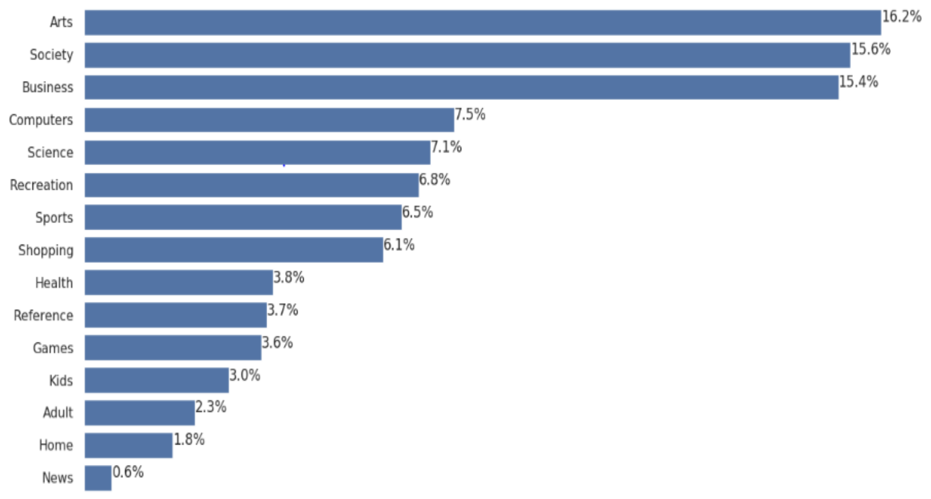


Figure 2. Percentage of English words per category

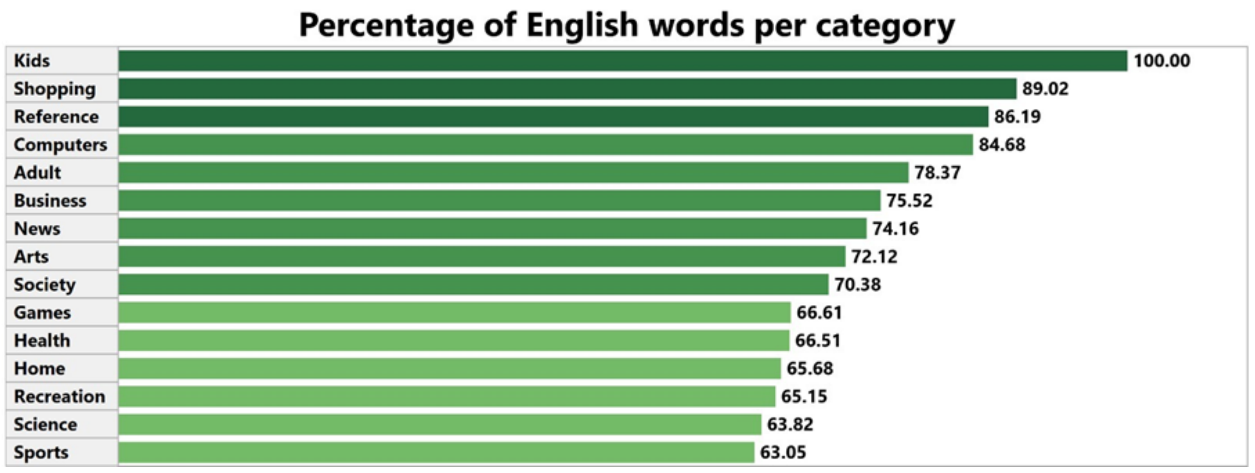


Figure 3. Average number of images per category

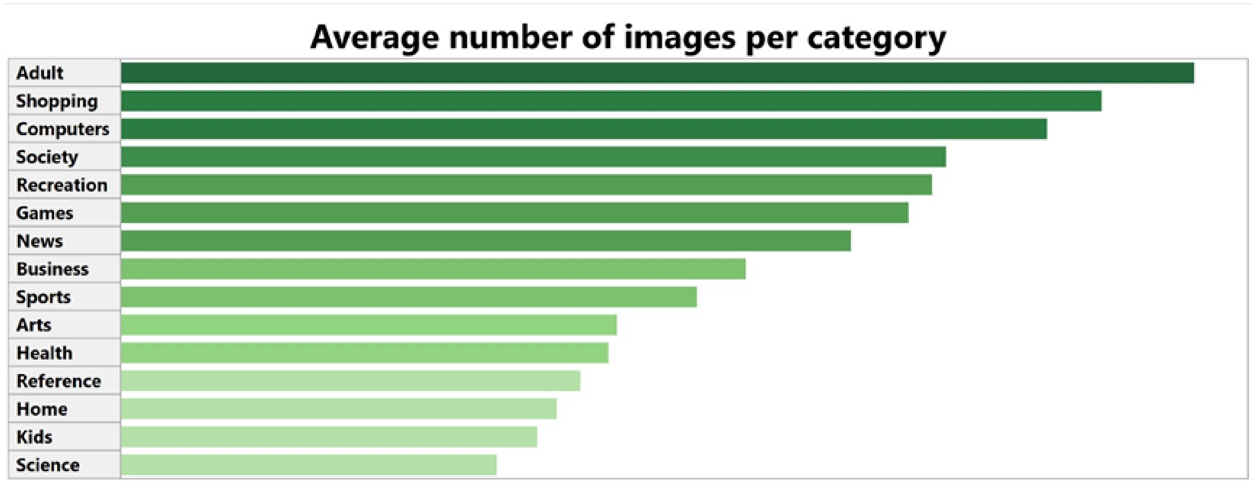


Figure 4. HTTPs status code

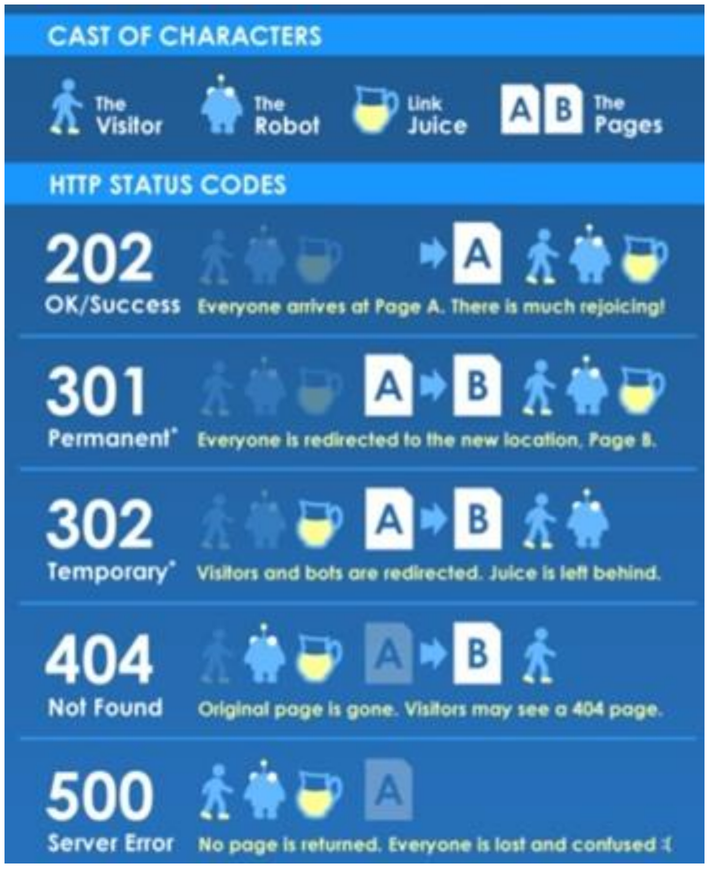
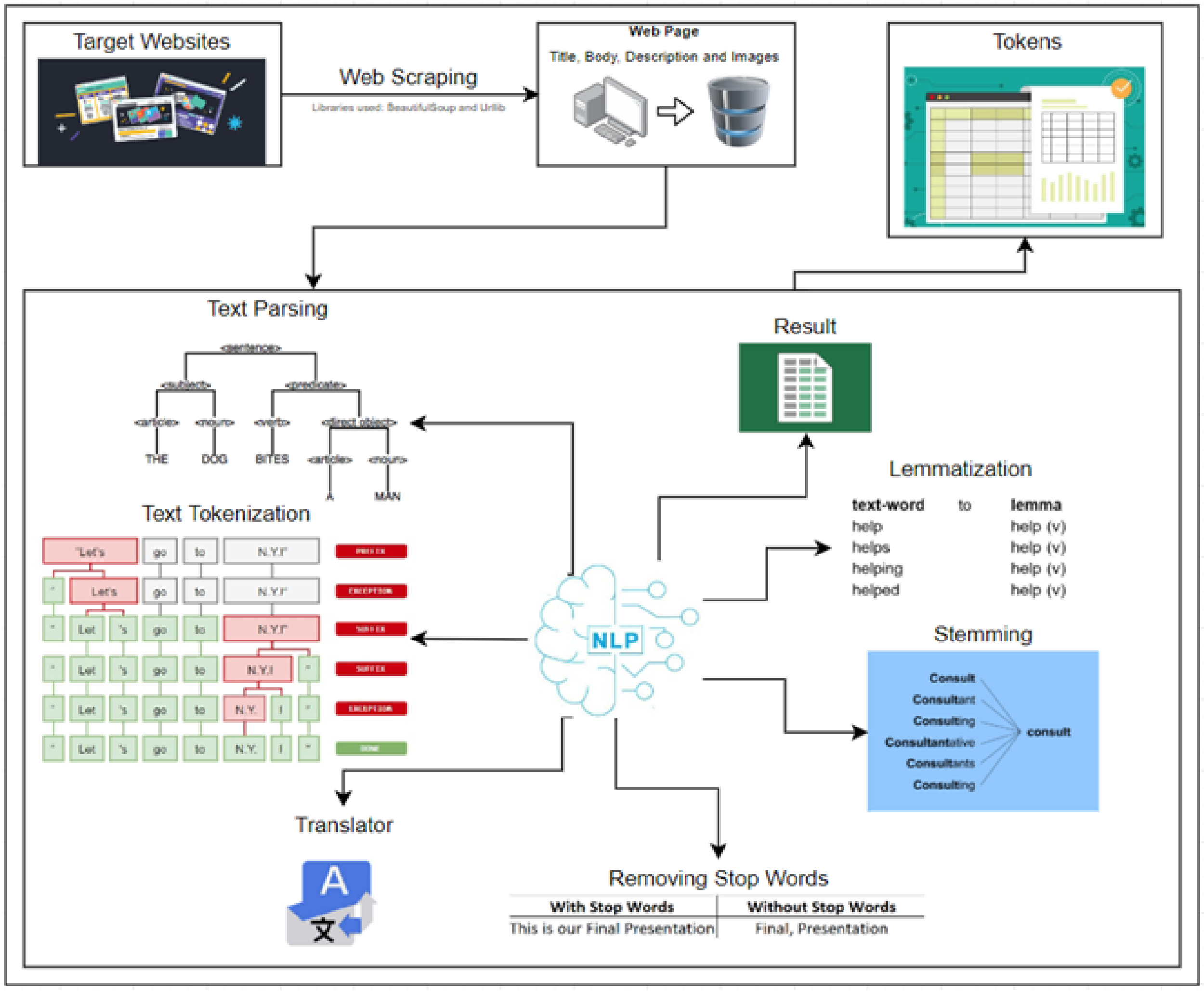

CAST OF CHARACTERS
The Visitor
The Robot
Link Juice
The Pages
HTTP STATUS CODES
202
OK/Success
Everyone arrives at Page A. There is much rejoicing!
301
Permanent*
Everyone is redirected to the new location, Page B.
302
Temporary*
Visitors and bots are redirected. Juice is left behind.
404
Not Found
Original page is gone. Visitors may see a 404 page.
500
Server Error
No page is returned. Everyone is lost and confused :(

Figure 5. Illustration of pre-processing



text-word	to	lemma
help		help (v)
helps		help (v)
helping		help (v)
helped		help (v)

With Stop Words	Without Stop Words
This is our Final Presentation	Final, Presentation

Figure 6. Most frequent words per category

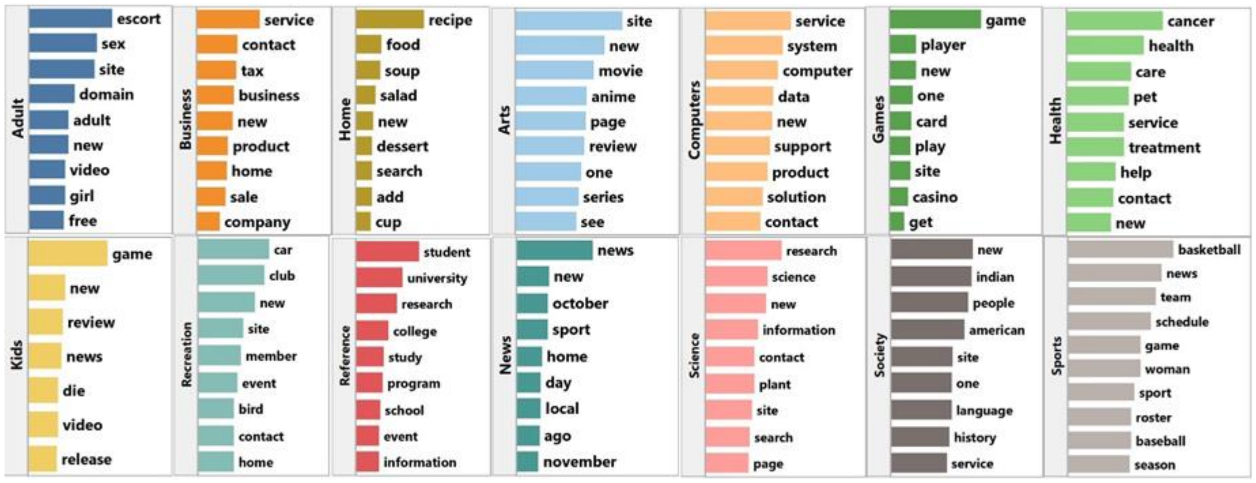


Figure 7. Proposed implementation flow chart of model

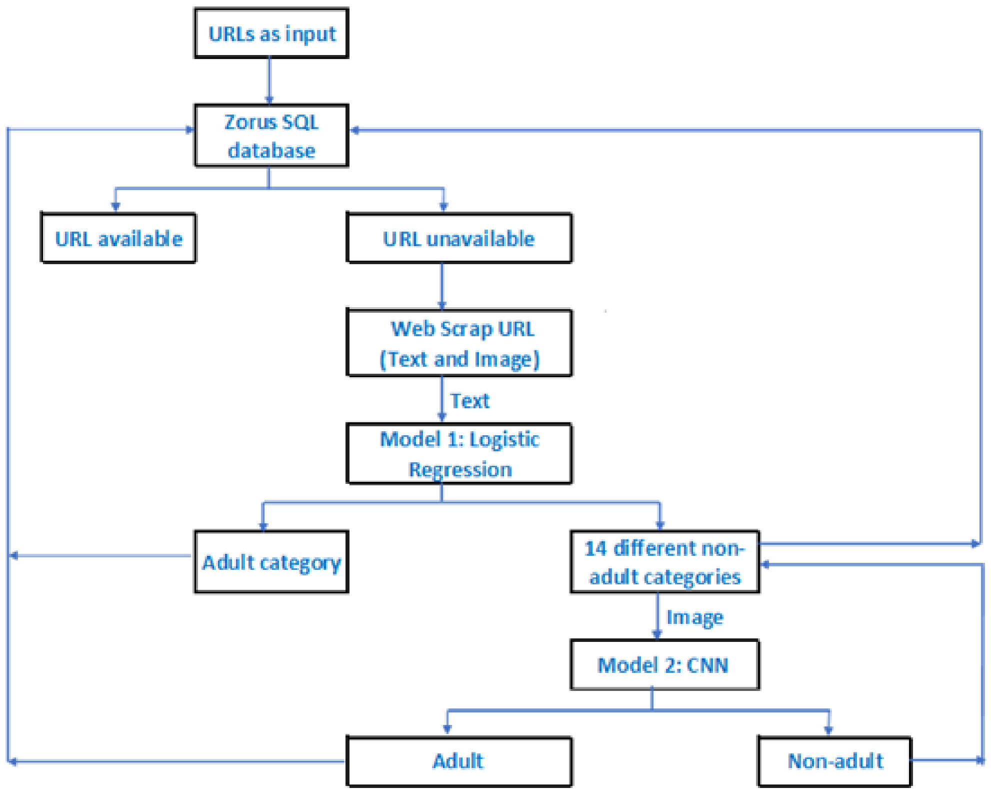
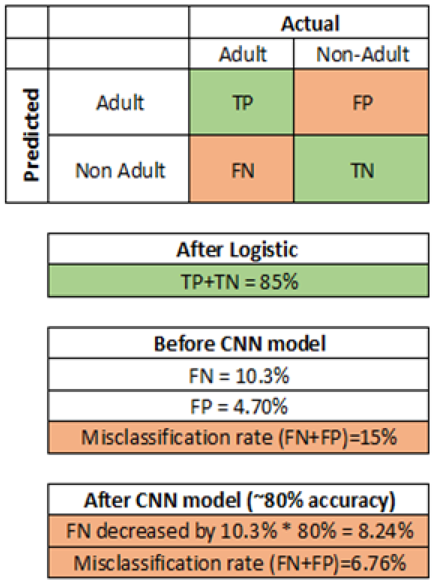


Figure 8. Performance of Models



		Actual	
		Adult	Non-Adult
Predicted	Adult	TP	FP
	Non Adult	FN	TN

After Logistic
TP+TN = 85%

Before CNN model
FN = 10.3%
FP = 4.70%
Misclassification rate (FN+FP)=15%

After CNN model (~80% accuracy)
FN decreased by 10.3% * 80% = 8.24%
Misclassification rate (FN+FP)=6.76%

Figure 9. Implementation of model flowchart through AWS

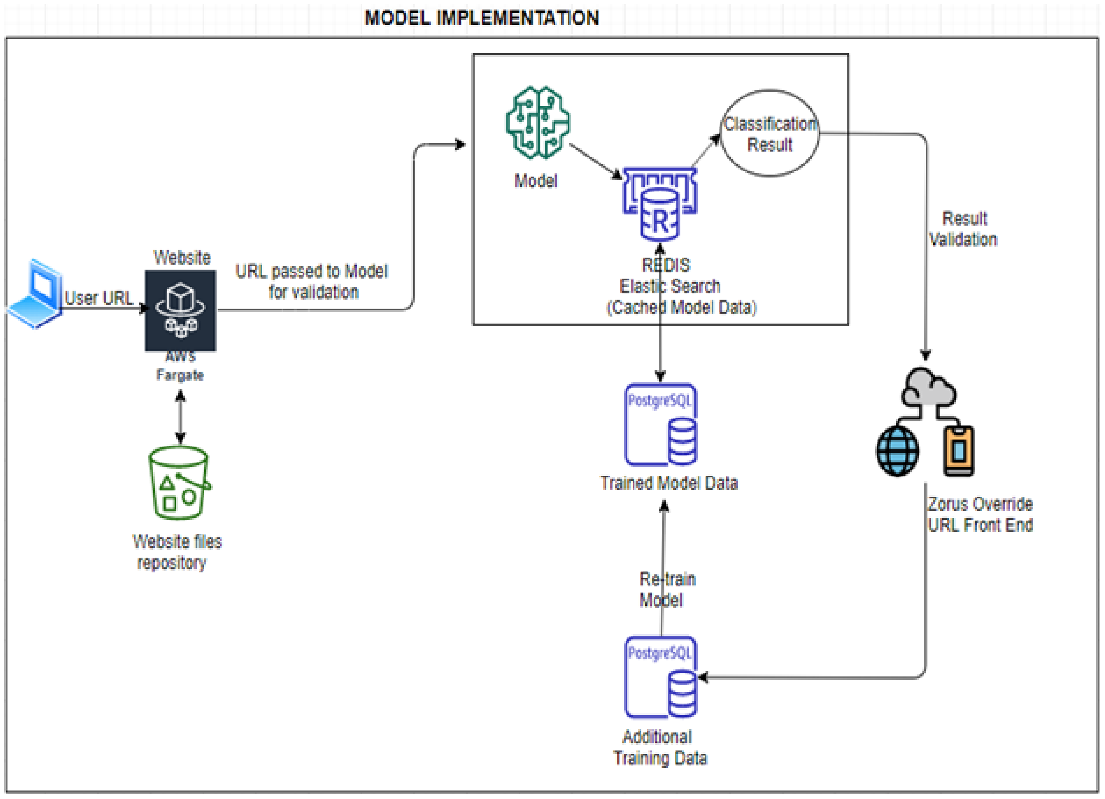
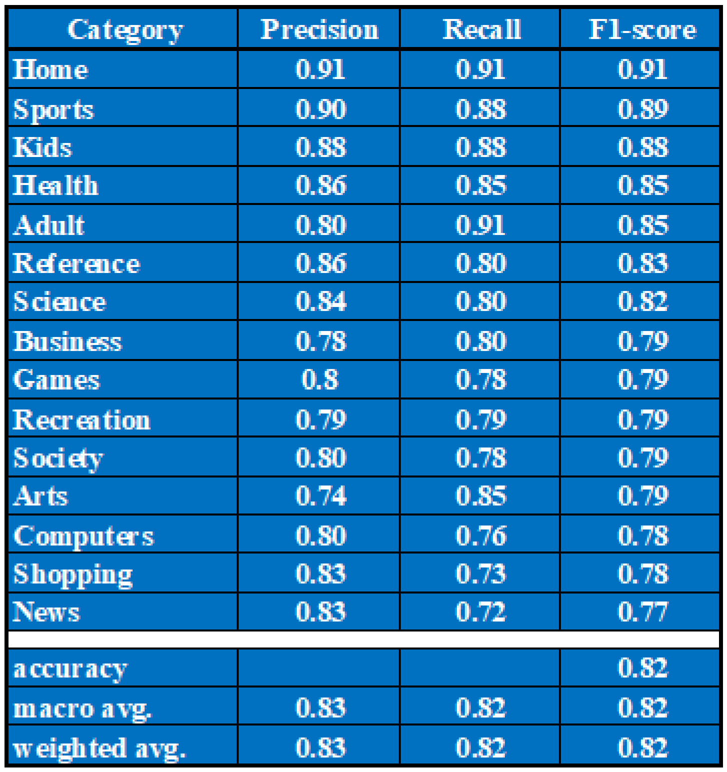


Figure 10. Performance matrix for logistic regression model



Category	Precision	Recall	F1-score
Home	0.91	0.91	0.91
Sports	0.90	0.88	0.89
Kids	0.88	0.88	0.88
Health	0.86	0.85	0.85
Adult	0.80	0.91	0.85
Reference	0.86	0.80	0.83
Science	0.84	0.80	0.82
Business	0.78	0.80	0.79
Games	0.8	0.78	0.79
Recreation	0.79	0.79	0.79
Society	0.80	0.78	0.79
Arts	0.74	0.85	0.79
Computers	0.80	0.76	0.78
Shopping	0.83	0.73	0.78
News	0.83	0.72	0.77
accuracy			0.82
macro avg.	0.83	0.82	0.82
weighted avg.	0.83	0.82	0.82