Web Resource Classifier

Big Data Computing

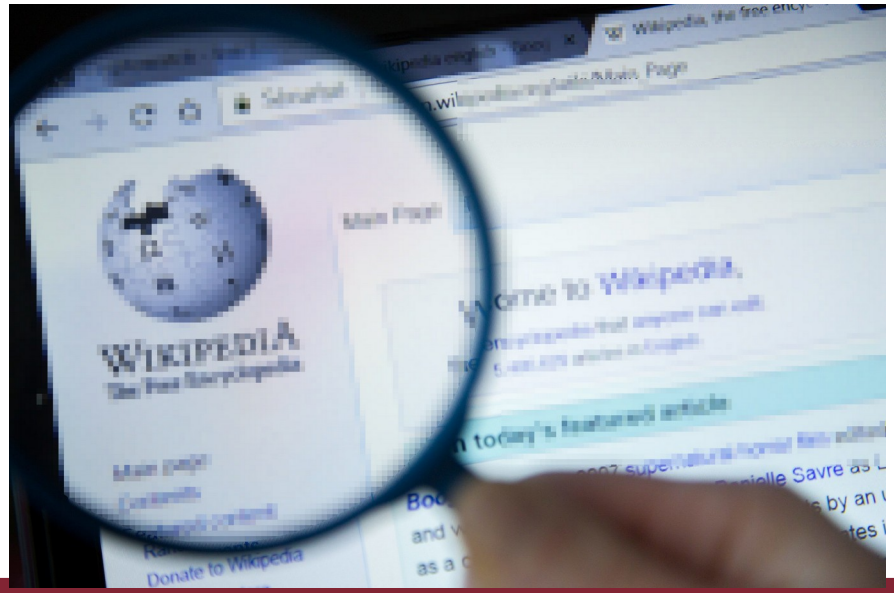
Simone Ruberto 1845772

Ruberto.1845772@studenti.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Introduction

The **goal** of the project is to address a multi-class classification problem performing category prediction based on html title tag and meta description tag of a web resource in real-time.



Dataset

The source dataset is available on kaggle and was created with the aim of predicting the website category from the website **urls**.

The data in the dataset was obtained from DMOZ (Open Directory Project), which is a vast and all-encompassing Internet directory edited and maintained by a group of volunteer editors from around the world. It was actively maintained from 1998 until 2017.

Dataset

The dataset is structured in three columns: id, url and type.

The type column indicates the website category and there are 15 different categories: Recreation, Science, Home, Computers, Sports, Health, Society, Shopping, Reference, Adult, Games, Arts, Kids, Business and News.

The dataset contains 1.500.000 records.

890145	http://www.louisianaschools.net/lde/scs/1938.html	Recreation
890146	https://online.dds.ga.gov/motorcycle/index.aspx	Recreation
890147	http://www.iamvd.com/ods/mre/index.htm	Recreation
890148	http://www.aamtfl.com	Recreation
890149	http://academic.pgcc.edu/transportation/index_files/motorcycle.htm	Recreation
890150	http://www.state.tn.us/safety/mrep.htm	Recreation
890151	http://www.learner.com.au/	Recreation
890152	http://www.toprider.com.au/	Recreation
890153	http://www.dropbears.com/m/mcplus/	Recreation

Dataset transformation

The initial dataset is not adequate for the problem, so I decided to add two more columns (title and description) to each record in the dataset.

To implement this transformation, it was necessary to perform HTTP GET requests for each URL in the dataset, resulting in approximately 1,500,000 requests.

The title and description were then scraped from the response of each request.

Problems

The transformation process encountered several challenges:

- Many URLs were no longer accessible or don't contains a title or a description.
- Many domains had expired.
- Many titles and descriptions were not in English.

Solutions

- In the case of inaccessible URLs or where the title and description could not be retrieved, these records were simply removed from the dataset.

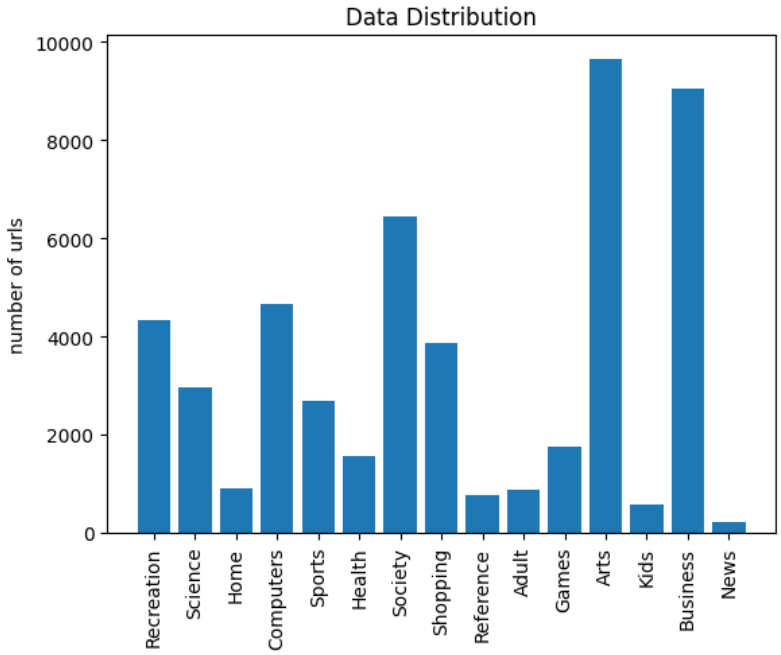
- To remove URLs from expired domains, I simply filtered the records based on the grouping of titles or descriptions where the count was greater than one. This approach was effective because many hosting sites display the same message when a domain expires.
  Additionally, while analyzing the obtained titles and descriptions, I noticed that the phrase "{{ domain name }} is for sale" frequently appeared in the title of expired domains. Therefore, I filtered out these records as well.

Solutions

- To eliminate records with titles and descriptions that are not in English, I employed a pre-trained language detection pipeline from the Spark NLP library. This pipeline is specifically designed for Language Detection & Identification, covering a comprehensive range of 220 languages. It allows for accurate identification and filtering of non-English content within the dataset.

Dataset distribution

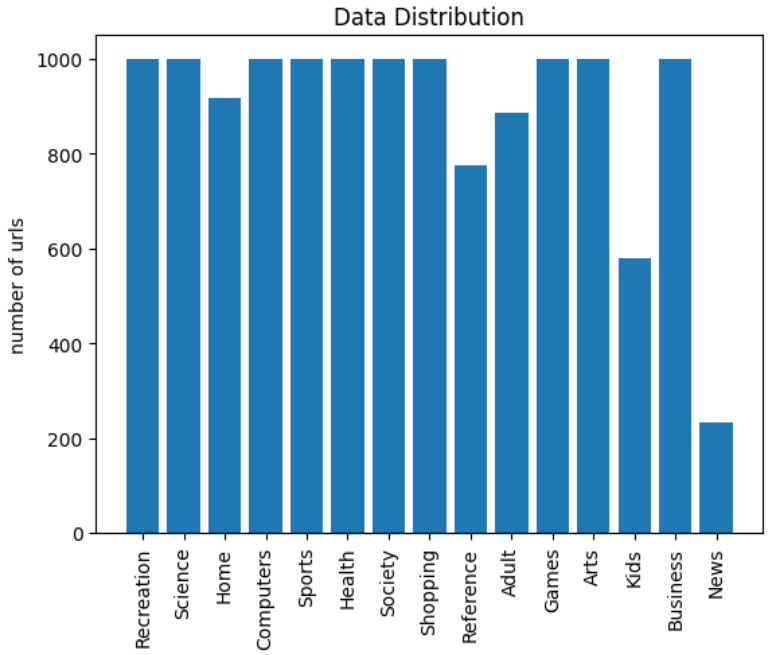
After the transformation and filtering phase, the dataset exhibits a significant imbalance, with a notable disparity among different data classes.



Dataset distribution

As the initial step, I truncated the number of records for each category to a maximum of 1000.

However, as the dataset was still not balanced, I utilized the ChatGPT API to generate synthetic data for categories that had fewer than 1000 records.

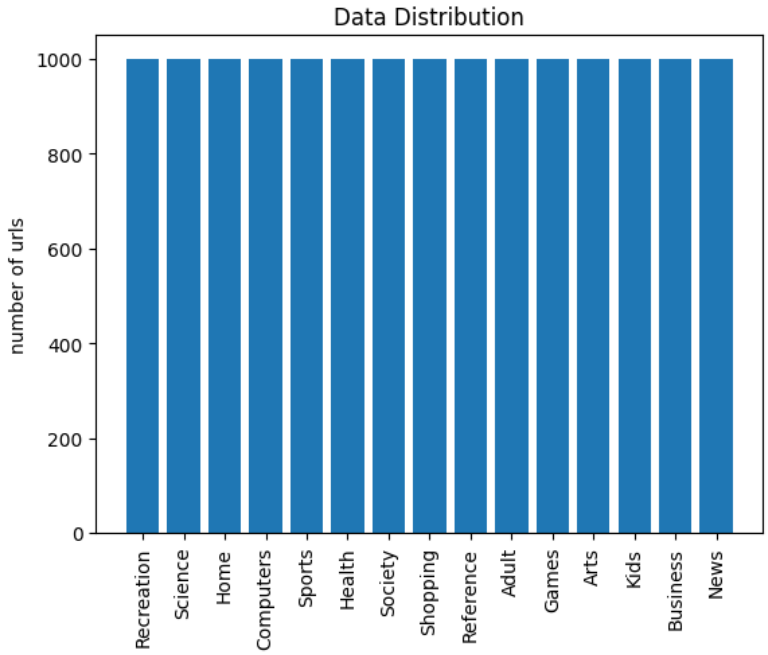


Generate synthetic data

```
response = openai.ChatCompletion.create(
    model=MODEL,
    messages=[
        {
            "role": "user",
            "content": f"generate {count} csv records relating to a title and a meta description \
                tag of a {category} category website, to delimit the record field use the semicolon \
                character, without the csv header and without the record number"
        }
    ],
    stop=None,
    temperature=choice(temperatures),
)
```

Dataset distribution

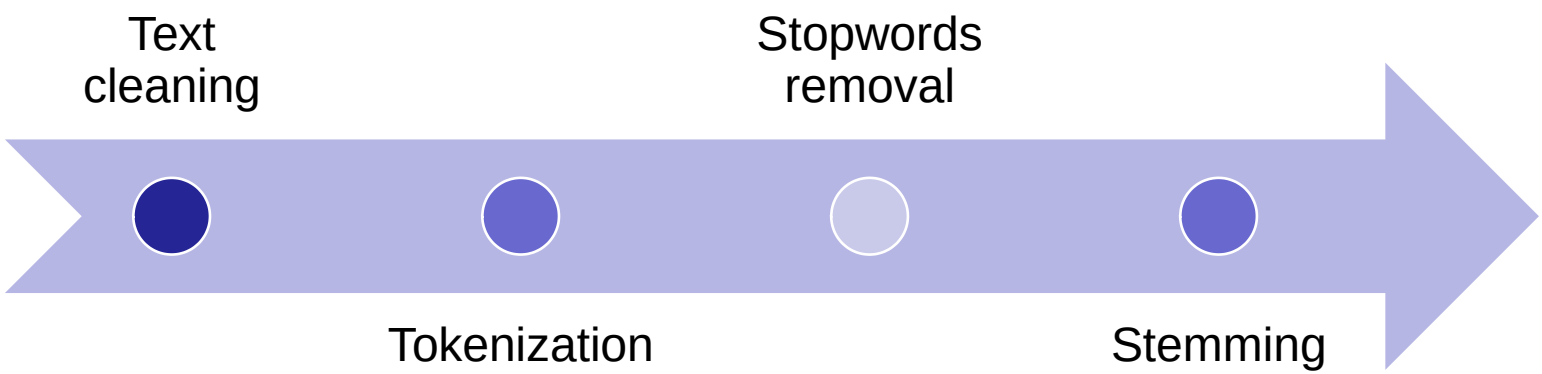
Thanks to these two approaches, I was able to obtain a more balanced and representative dataset.



Data visualization

Data pre-processing

The pre-processing pipeline consists of the following phases:



Data pre-processing

- CountVectorizer
- IDF (Inverse Document Frequency)
- **VectorAssembler**
- **StringIndexer**

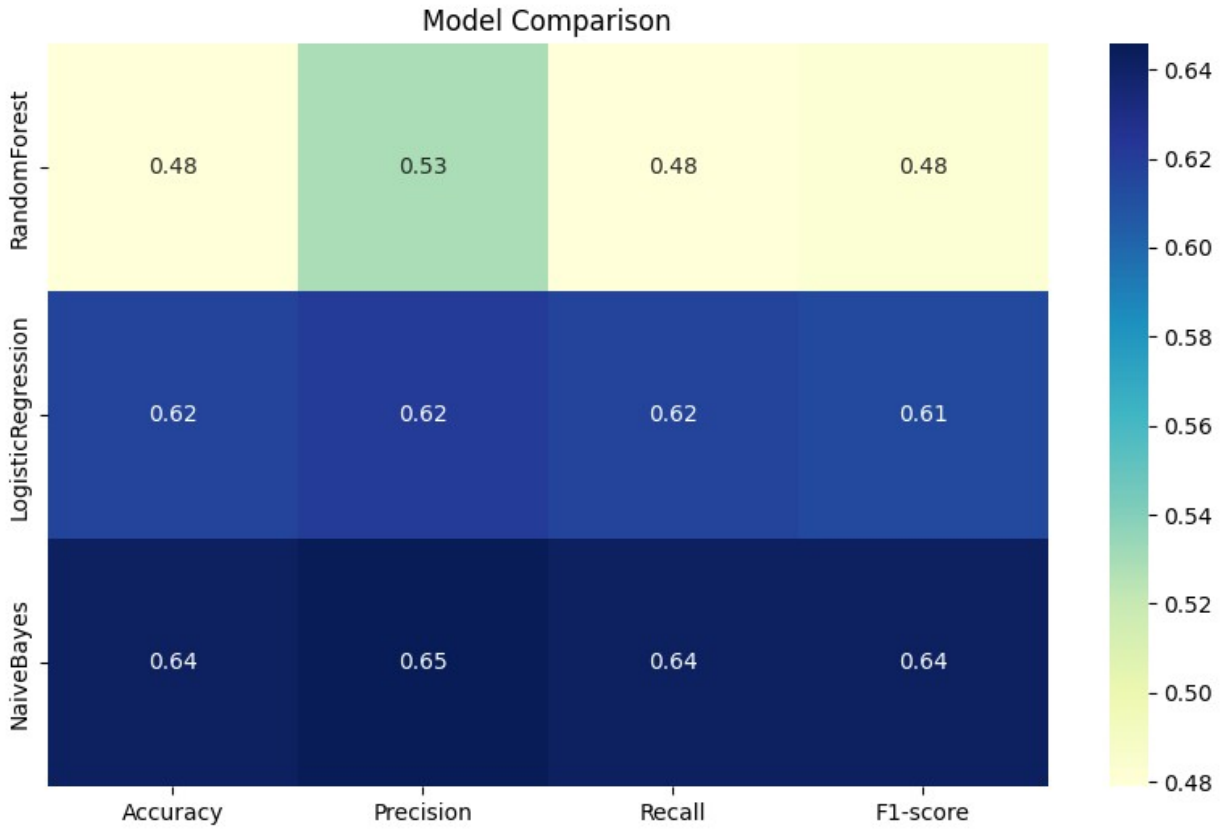
Split the dataset

The dataset has been split into:

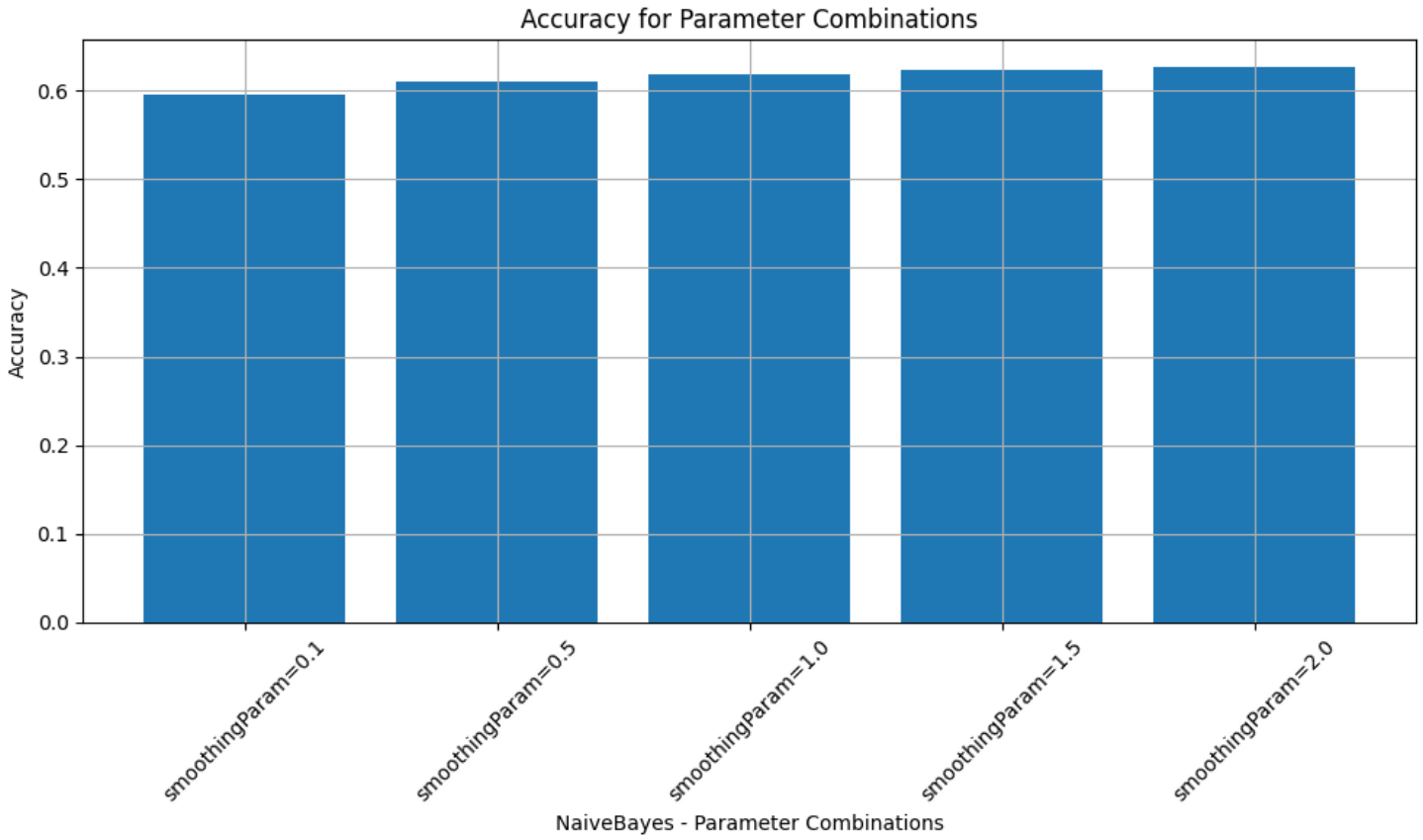
- **training set** 75% of the data
- **test set** the remaining 25% of the data

Model Comparison

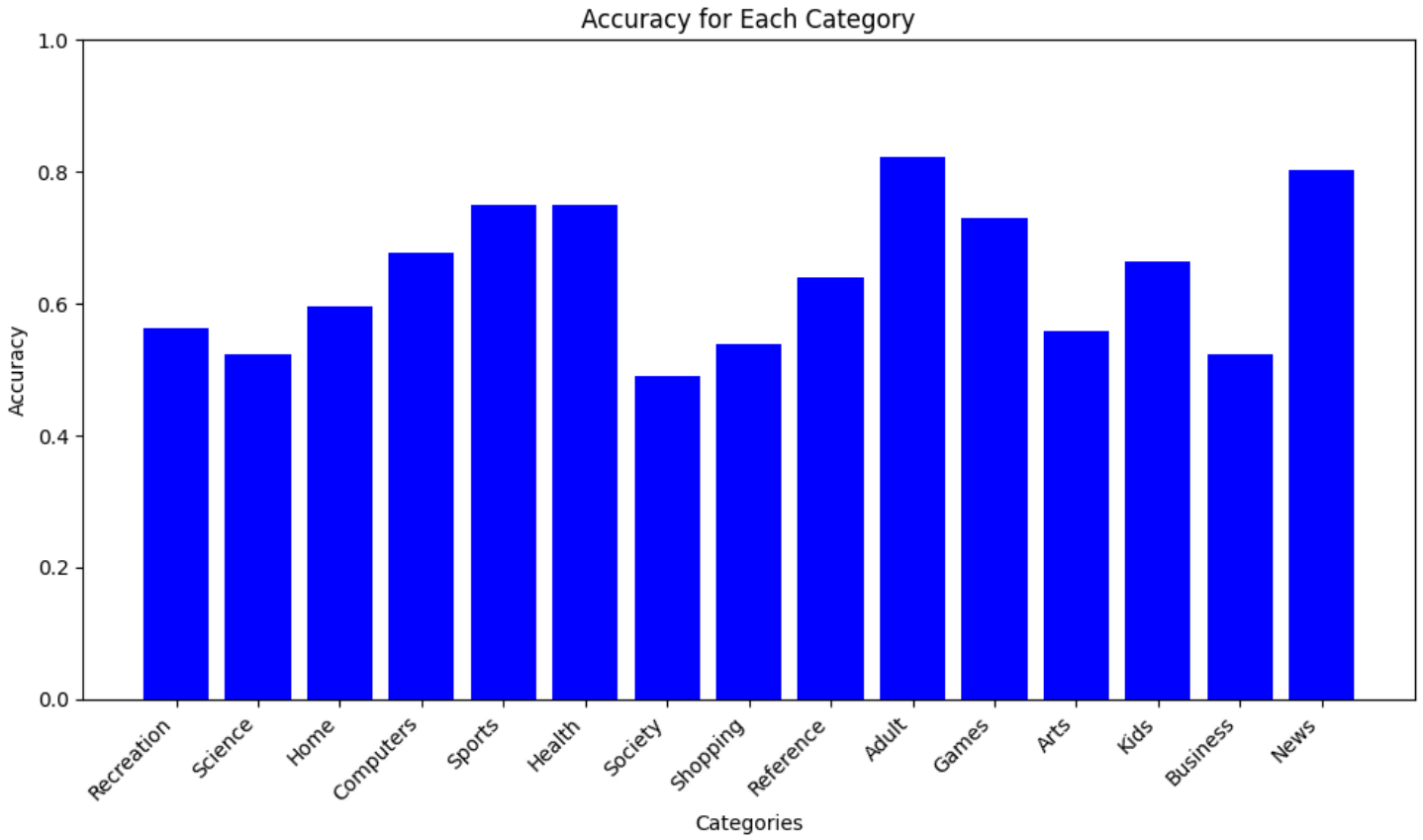
Web resource classification with cross validation and parameter tuning



Model Comparison



Model Comparison



Live demo

Web Resource Classificator

Navigate to a web resource and find out which category it belongs to.
Classification can only take place if the resource has a nont empty tag title and a tag meta description.

Conclusions

- The assumption that features are conditionally independent fits the problem well
- Classifying web resources is a challenging task because of the wide range of domains and topics covered on the Internet.
- Having an up-to-date data set is crucial

Thank you for your attention