Detection And Blocking Of Advertisements Using Machine Learning

MENTOR : MR UTPAL RAY

BY:
SHUBHAM DUTTA(10)
ADITI DEO(45)

Our Goals:

Use supervised learning to bootstrap the process whether an URL is ad related or not.

Overcome the manual updation of easyList used by adblock plus to identify ads.

Deploy the model as an extension

Initial workflow achieved:

STEP 1	STEP 2	STEP 3	STEP 4	STEP 5
DATA COLLECTION USING INJECTED CONTENT SCRIPT	**DATA SENT TO FLASK SERVER USING AJAX**	**EXTRACTING FEATURES FROM DATA SET**	**SAVE DATA IN DATABASE**	**LABELING THE SAVED DATA BY MATCHING AGAINST EASYLIST**
 popup.js	 client.js	 eature_extractor py	 dbconn.py	 parser.py

Data Set:

- We have collected maximum URLs using injected content script through an extension.

- To generate training and ground truth labels for a collection of URL data,we compared URLs against EasyList filters used by Adblock plus.

➔ **Task: binary classification**

➔ **Total set: 26969 rows**

➔ **Ad based set: 6986 rows**

➔ **Non ad based set: 19983 rows**

Data Point:

General structure of an URL:

scheme://netloc/path;parameters?query#fragment

PROTOCOL HOST NAME(DOMAIN)

- We Parse the URL into six components, returning a 6-tuple for further processing.
  E.x:-
  o = urlparse('http://www.cwi.nl:80/%7Eguido/Python.html')

ParseResult(scheme='http', netloc='www.cwi.nl:80', path='/%7Eguido/Python.html', params='', query='', fragment='')

Feature Collection:

- For each URL, while gathering, we stored features that were computed within the context of the page as well as various URL-based and lexical features .
- Each data point was encoded as a 6-dimensional feature vector.
- All binary features are used to ensure equal weightage during training.

KEYWORDS: ads googlead,sponsored etc.

Valid query parameters

Lexical features

Semicolon to separate parameters

SIZE and DIMENSIONS mentioned in the URL

Requested url on same domain or not?

URL requested from iframe or not?

DATASET:

Ad based data

1	url	words	s_size	semicolor	size	words_ch	domain	f_iframe	res
2	https://adssettings.google.com/	1	0	0	0	1	0	1	1
3	https://adssettings.google.com/	1	0	0	0	1	0	1	1
4	https://adssettings.google.com/	1	0	0	0	1	0	1	1

Non Ad based data

219	/news/2018/11/saudi-top-prose	0	0	0	0	0	0	0	0
220	/customer-service/terms-of-ser	0	0	0	0	0	0	0	0
221	http://www.ntv.co.jp/englishne	0	0	0	0	0	1	0	0
222	https://www.ndtv.com/converg	0	0	0	0	0	1	0	0

Classification Model Used :

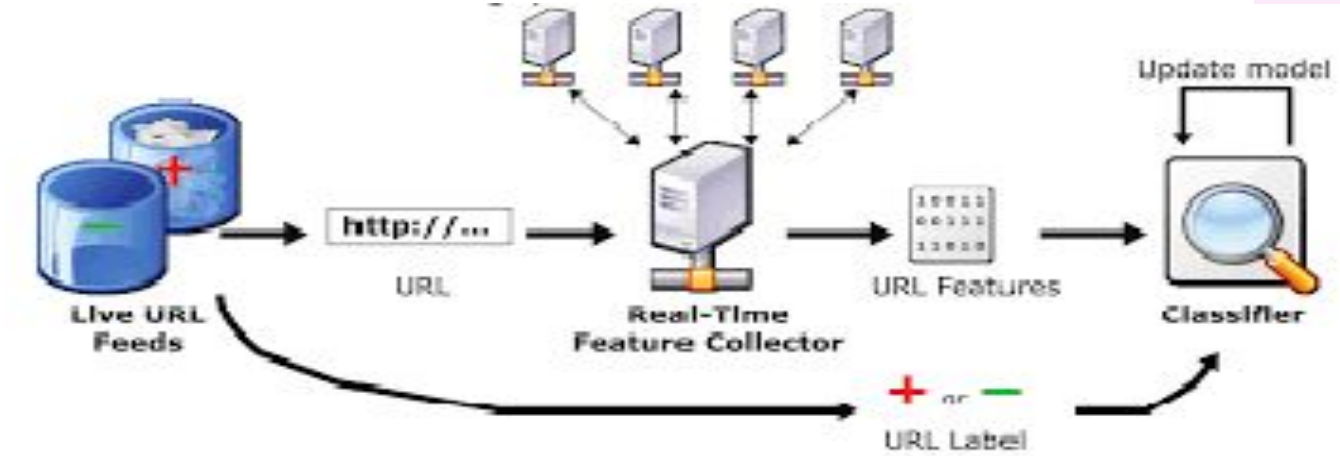
Below are the classification algorithm we employed due to their suitability for binary classification problems.More algorithms can be employed.Average accuracy are as follows.

94.9%
SUPPORT VECTOR MACHINE

93.1%
LOGISTIC REGRESSION

- We train our data locally using the algorithms above and save the model in server using pickle.

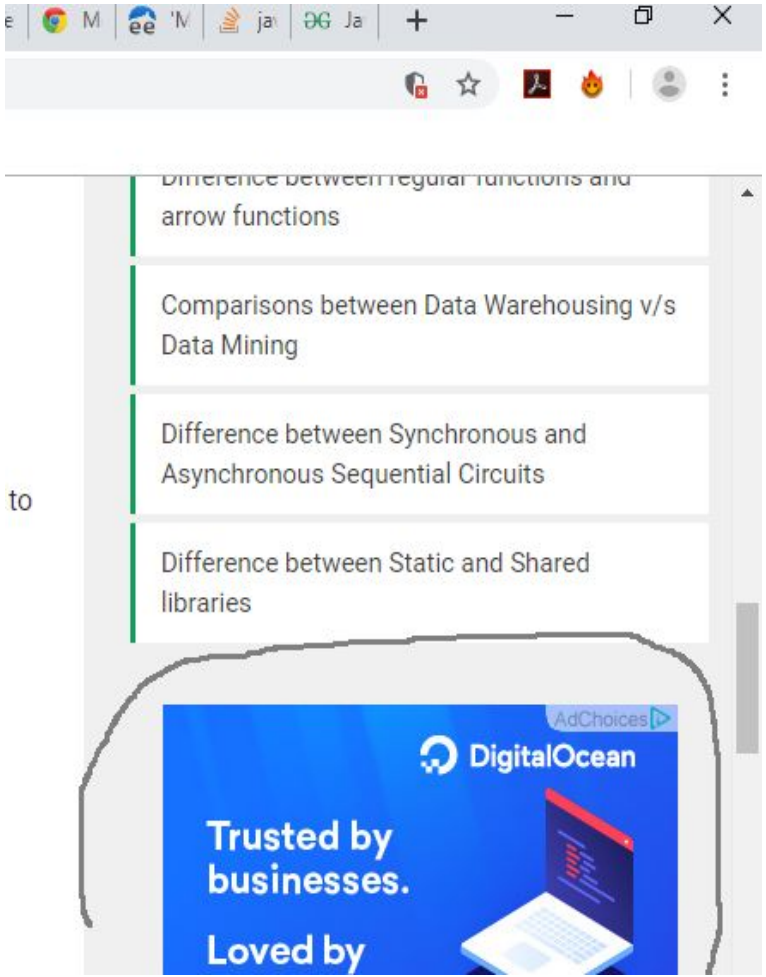
- To implement the classification, we made use of the machine learning development kit in Python called Scikit-Learn



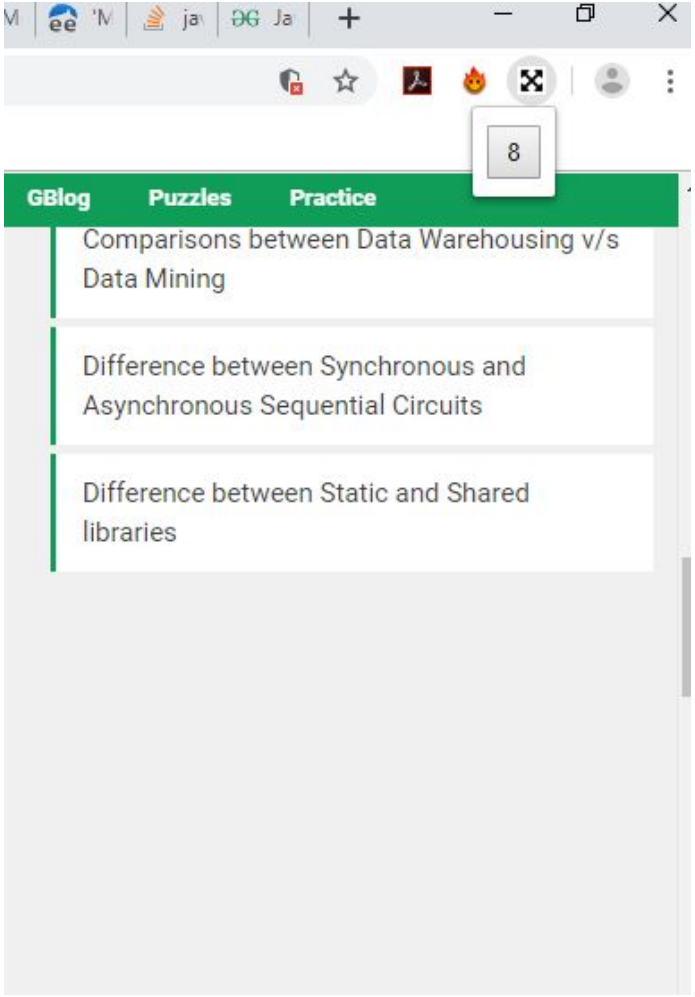
Work Flow:

STEP 1	STEP 2	STEP 3	STEP 4
PROJECT IS DEPLOYED AS AN EXTENSION .	**URL IS GATHERED USING CONTENT SCRIPT & SEND TO FLASK SERVER .**	**MODEL SAVED IN SERVER SENDS BINARY RESPONSE AS PREDICTION**	**UPDATE DOM OF WEB PAGE (BLOCK AD BASED URL).**

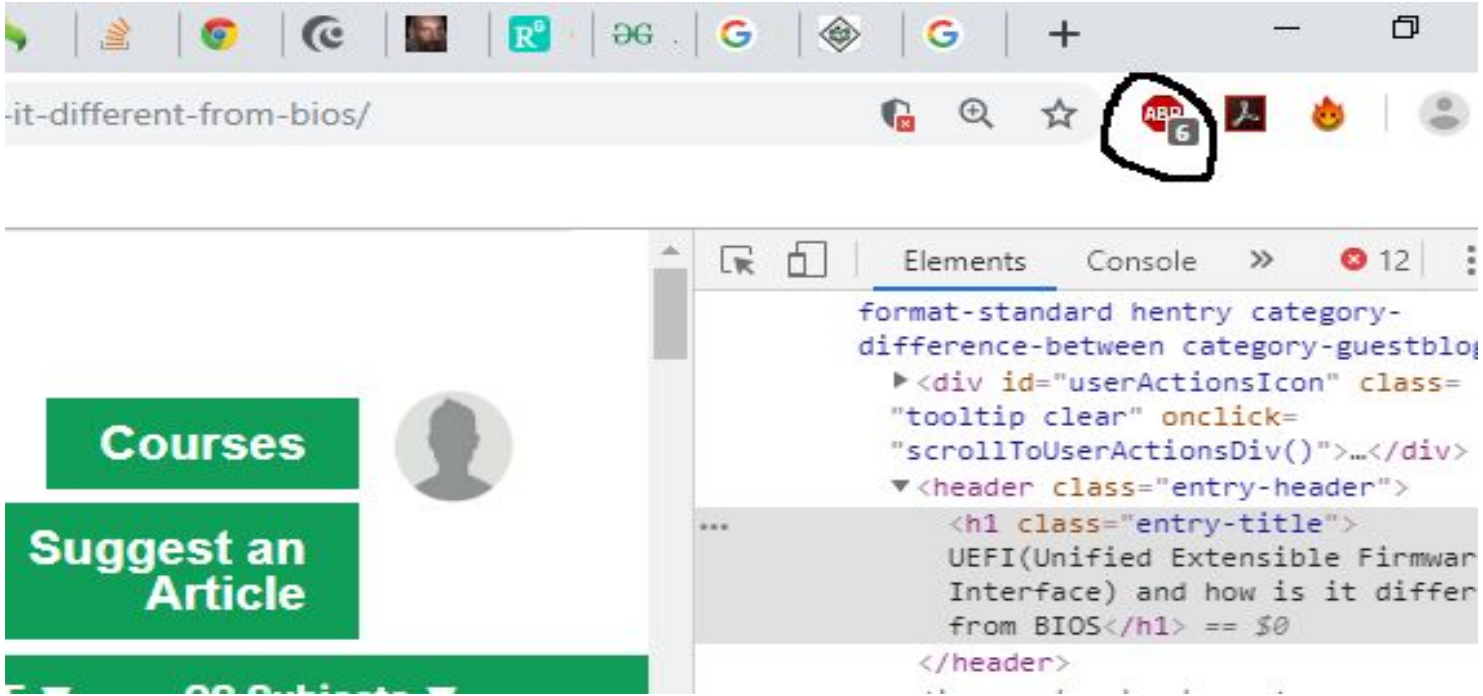
- The new data points will also be saved in the database along with its label which will be used to train the model again with enough new data and old data.This updation is done locally time to time.



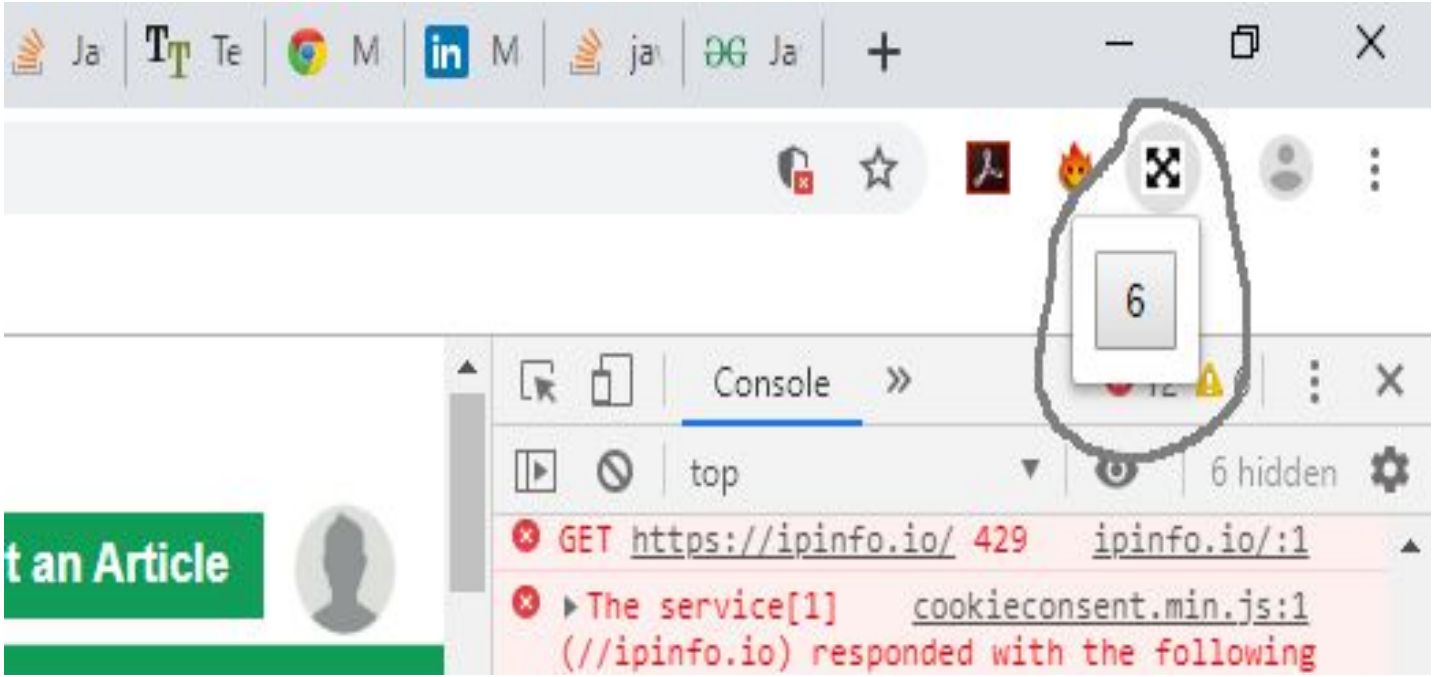
ADS BLOCKED USING OUR EXTENSION



- **ADS blocked by adblock plus.**



- **ADS blocked by our extension on same website.**



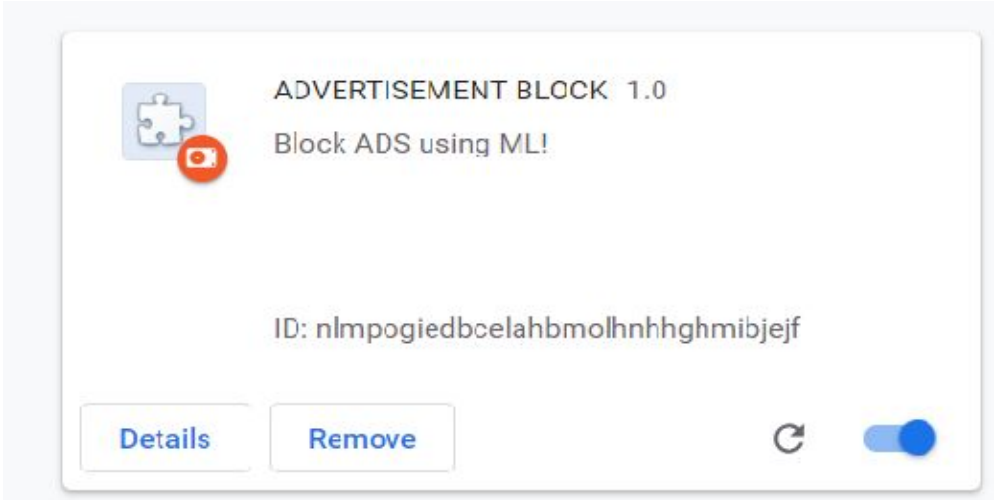
Deployment:

Model will be deployed as an extension

Users just need to install the extension in chrome browser which will ask for prediction from server running on a different machine. (using ip of server machine)

Web application frame

-work used is “**flask**”



Future Work:

- More diverse collection of data in more optimized & efficient way.
- Reaction time taken by webpages to block ad can be reduced.
- More adequate features can be extracted as per data and more feature selection methods can be used.

References:

- Bhagavatula, Sruti & Dunn, Christopher & Kanich, Chris & Gupta, Minaxi & Ziebart, Brian. (2014). Leveraging Machine Learning to Improve Unwanted Resource Filtering. Proceedings of the ACM Conference on Computer and Communications Security. 2014. 95-102. 10.1145/2666652.2666662.
- http://flask.pocoo.org/docs/1.0/
- https://easylist-downloads.adblockplus.org/easylist.txt(filter list used by adblock)

Thank You!