feature\_names = []

for key, values in vectorizers.items():

for f in vectorizers[key].get\_feature\_names():

feature\_names.append(key + '\_' + f)

df = pd.DataFrame(data=X\_pos, index=pos.index, columns=feature\_names)

df['url']=df.index

df = df.reset\_index(drop=True)

df.to\_csv('features.csv', index=False)

The approach at hand is solely based on the extraction of

features from intrinsic characteristics of the source code, more precise the archived fake-shops’ **main landing page**. **Images** or other **metadata** were **not** included in the analysis.

The following features were identified as potentially relevant through a process of manual code inspection: **tokenized HTML, CSS and JavaScript text, comments and individual tags, tag-attribute-value patterns as well as the HTML tree structure**.

The data-to-feature converter is implemented in Python and

the features were extracted from the ground truth dataset. The resulting textual data was converted to numeric values in order to train the machine learning models and evaluate their classification abilities.

Our implementation of the analytics component is based on a TF-IDF vectorizer. It converts textual features into a matrix representation. The TF-IDF value increases proportionally to the frequency with which a word occurs in the document and is offset by the number of documents in the corpus in which the word is contained.

Adding additional non related features as for example the shops **geographical location, payment methods, shown trustmarks**, individual models for pre-detecting the existence of shop or CMS systems but also including human understandable features are required improvements. The authors plan on gathering empirical data on **listed products and price points** in fake-shops, to better understand means and ways on how fraudulent offerings address their target groups

Training data is supposed to be at <https://malzwei.at/acl_users/credentials_cookie_auth/require_login?came_from=https%3A//malzwei.at/anfrage/anfrage> but it requires login

(20 GB of archived fake-shop HTML, CSS, JavaScript and 345 MB of extracted features and metadata), free to re-use for scientific and **noncommercial research** purposes.

POC code extracts **url, date, titles, text, images, css, js, forms, iframes, stats**

result = HTMLExtractorOutput(

url=url,

date=datetime.now(),

titles=r.get\_titles(),

meta=r.get\_meta(),

urls=r.get\_urls(),

text=r.get\_text(clean\_lxml),

images=r.get\_images(),

svgs=r.get\_svgs(),

css=r.get\_css(),

js=r.get\_js(),

forms=r.get\_forms(),

iframes=r.get\_iframes(),

stats=r.get\_stats(),

)