from autoscraper import AutoScraper

url = 'https://stackoverflow.com/questions/2081586/web-scraping-with-python'

# We can add one or multiple candidates here.

# You can also put urls here to retrieve urls.

wanted\_list = ["What are metaclasses in Python?"]

scraper = AutoScraper()

result = scraper.build(url, wanted\_list)

print(result)

output:

[

'How do I merge two dictionaries in a single expression in Python (taking union of dictionaries)?',

'How to call an external command?',

'What are metaclasses in Python?',

'Does Python have a ternary conditional operator?',

'How do you remove duplicates from a list whilst preserving order?',

'Convert bytes to a string',

'How to get line count of a large file cheaply in Python?',

"Does Python have a string 'contains' substring method?",

'Why is “1000000000000000 in range(1000000000000001)” so fast in Python 3?'

]

Scraper object

scraper.get\_result\_similar('https://stackoverflow.com/questions/606191/convert-bytes-to-a-string')

**Getting exact result**

from autoscraper import AutoScraper

url = 'https://finance.yahoo.com/quote/AAPL/'

wanted\_list = ["124.81"]

scraper = AutoScraper()

# Here we can also pass html content via the html parameter instead of the url (html=html\_content)

result = scraper.build(url, wanted\_list)

print(result)

proxies = {

"http": 'http://127.0.0.1:8001',

"https": 'https://127.0.0.1:8001',

}

result = scraper.build(url, wanted\_list, request\_args=dict(proxies=proxies))

Now we can get the price of any symbol:

scraper.get\_result\_exact('https://finance.yahoo.com/quote/MSFT/')

### Saving the model

We can now save the built model to use it later. To save:

# Give it a file path

scraper.save('yahoo-finance')

And to load:

scraper.load('yahoo-finance')