

Overcoming Challenges and Increasing Efficiency



Pratheerth Padman
FREELANCE DATA SCIENTIST



Module Overview



Challenges in web scraping

Tips to increase scraping efficiency

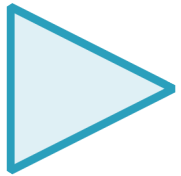
Best practices



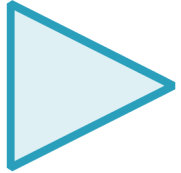
Challenges in Web Scraping



Lack of Bot Access



Websites free to decide if they allow bot access



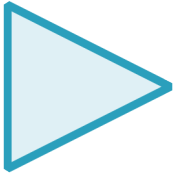
Reasons could vary from the malicious to the benign



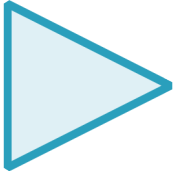
Always check robots.txt and terms of service



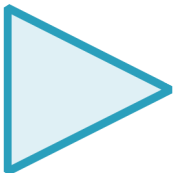
CAPTCHA's



Implemented to keep spam away



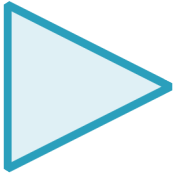
Basic scripts tend to fail



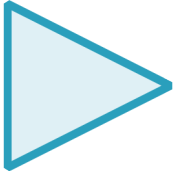
Advancements have been made in both captcha tech. and
workarounds



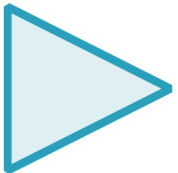
Frequent Structural Changes



To improve UX and add features



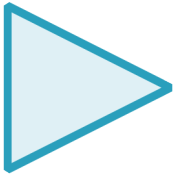
Frequent structural changes pose a heavy problem for scrapers



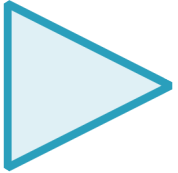
Monitor changes / outsource scraping projects



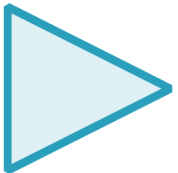
Getting Blocked / Banned



Unnaturally high requests from same IP – IP blocking



Honeypot traps



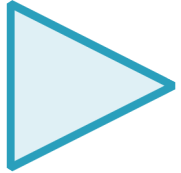
Bot blocking services



Dynamic Content



Plenty of dynamic content != crawler friendly



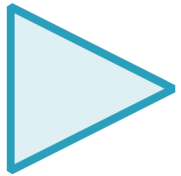
Lazy loading images, infinite scrolling



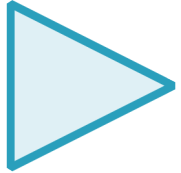
Selenium has a lot of techniques to overcome challenges of dynamic content



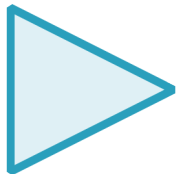
Questionable Data Quality



Important to collect quality data lest it poisons further usage



Difficult to do since it needs to be done in real time



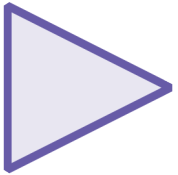
Test cases to check scraped data is helpful



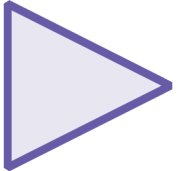
Tips to Increase Efficiency & Best Practices



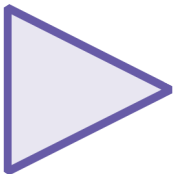
Avoid Loading Images



Nearly all modern, dynamic websites contain lots of images



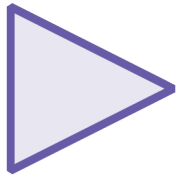
Selenium loads everything, including images. Slows script down



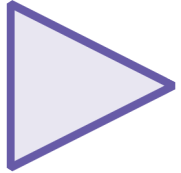
Use chromeoptions to load pages without loading images



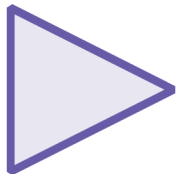
Employ Disk Cache



Disk caching reduces page load times



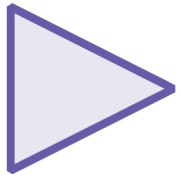
Stores assets like css, javascript



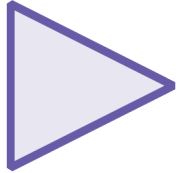
Can be implemented with chromeoptions



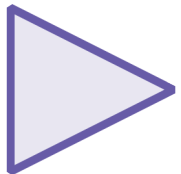
Close Driver with Quit



Important to properly close driver after finishing task



Selenium webdrivers use additional resources



Use `driver.quit()` instead of `driver.close()`



Best Practices

Respect robots.txt

**Don't hit servers
too frequently**

**Don't scrape
during peak hours**

**Use a headless
browser**

**Avoid using same
crawling pattern**

**Embrace
transparency**



Summary



Comprised of two sections – challenges and tips for web scraping

Explored 6 different types of challenges

Went over 3 tips to increase efficiency

Learned 6 accepted best practices

