University Of Nebraska Omaha

Tools For Data Analysts Final Project

ECON 8320

Emmanuel Erhirhie

Student – Athletes NIL Deals

INTRODUCTION

In recent years, the issue of student-athletes being compensated for their name, image, and likeness (NIL) has become a hotly debated topic in the world of college sports. Until recently, the National Collegiate Athletic Association (NCAA) prohibited student-athletes from accepting any form of compensation for their NIL, citing concerns about amateurism and preserving the integrity of college athletics. However, in 2021, the NCAA revised its rules to allow student-athletes to profit from their NIL, opening the door for them to sign endorsement deals, monetize their social media accounts, and participate in other commercial ventures. This change has significant implications for college sports, as it has the potential to fundamentally alter the way athletes are compensated and the way college sports are viewed and consumed. In this context, it is important to examine the issue of student-athlete NIL deals in detail, exploring the benefits and challenges of this new landscape and considering the potential implications for the future of college athletics.

Top of Form

PURPOSE

The purpose of this web scrapping is to gather data to show us student athletes getting NIL deals which has shown a test of good faith in which students devote their whole life to sport and get paid for their hard work.

METHODOLOGY

The project was done using VS CODE notebook where I downloaded python, and libraries like LXML, Beautifulsoup4, Pandas and Request.

I started the project by using beautiful soup to scrap the website picking an HTLM link that easy for me to extract the data where I later used htlm.parlay instead of lxml in order for the link be easily read by python. Then further went to use ‘Request’ to collect the data from the website, which was stored in a data frame called ‘Pandas’ which was cleaned and was later exported to as csv file that shows a table of all the athletes with their NIL deals, the school they play for and their social media accounts.

Below are the python codes used:

import requests

from bs4 import BeautifulSoup

import pandas as pd

# create empty lists to store the data

schools = []

sports = []

sponsorships = []

social\_media = []

base\_url = 'https://nilcollegeathletes.com'

def table\_per\_page(url, page\_number):

    response = requests.get(url + str(page\_number))

    # create a BeautifulSoup object with the HTML content

    soup = BeautifulSoup(response.content, 'html.parser')

    # find all the athlete student\_table on the page

    student\_table = soup.find('table',{'class': 'min-w-full'} )

    # print('student\_table is: ', student\_table)

    page\_df = pd.DataFrame(columns=['Name', 'Sponsor', 'University', 'Sport'])

    for row in student\_table.tbody.find\_all('tr'):

        # Find all data for each column

        columns = row.find\_all('td')

        sponsors = []

        athlete\_url\_suffix = columns[4].find('a')['href']

        athlete\_url\_path = base\_url + athlete\_url\_suffix

        print(athlete\_url\_suffix)

        athlete\_df = get\_athlete\_page(athlete\_url\_path)

        if(columns != []):

            if columns[1].ul:

                for sponsor in columns[1].ul.find\_all('li'):

                    sponsors.append(sponsor.text.strip().strip('\n'))

            name = columns[0].text.strip()

            sponsor = sponsors

            university = columns[2].text.strip()

            sport = columns[3].text.strip()

            if athlete\_df.get('Instagram') is not None:

                instagram = athlete\_df.get('Instagram')[0].strip('\n').split()[0]

            else:

                instagram = ''

            if athlete\_df.get('Twitter') is not None:

                twitter = athlete\_df.get('Twitter')[0].strip('\n').split()[0]

            else:

                twitter = ''

            page\_df = page\_df.\_append({'Name': name,  'Sponsor': sponsor, 'University': university, 'Sport': sport, 'Instagram': instagram, 'Twitter': twitter}, ignore\_index=True)

    # print(df)

    return(page\_df)

def get\_all\_pages(url):

    all\_pages = []

    for page in range(1,420):    #Change the 430 value to the number of pages on the website

    # send a GET request to the website

        df = table\_per\_page(url, page)

        # print(df)

        all\_pages.append(df)

    merged\_df = pd.concat(all\_pages)

    print(merged\_df)

    merged\_df.to\_csv('data\_frame\_student.csv')

def get\_athlete\_url():

    print('test')

def get\_athlete\_page(athlete\_base\_url):

    response = requests.get(athlete\_base\_url)

    soup = BeautifulSoup(response.content, 'html.parser')

    # print(soup.encode("utf-8"))

    student\_table = soup.find('dl')

    # student\_df = pd.DataFrame(columns=['University', 'Sport', 'Sponsor', 'Platforms', 'Instagram', 'Twitter'])

    athlete\_df = pd.DataFrame()

    student\_attribute = []

    for student\_head in student\_table.find\_all('dt'):

        student\_attribute.append(student\_head.text.strip())

    student\_attribute\_value = []

    for student\_value in student\_table.find\_all('dd'):

        student\_attribute\_value.append(student\_value.text.strip().strip('\n'))

    athlete\_df['key'] = student\_attribute

    athlete\_df['value'] = student\_attribute\_value

    return athlete\_df.set\_index('key').T  # flips rows into columns and columns into rows

def main():

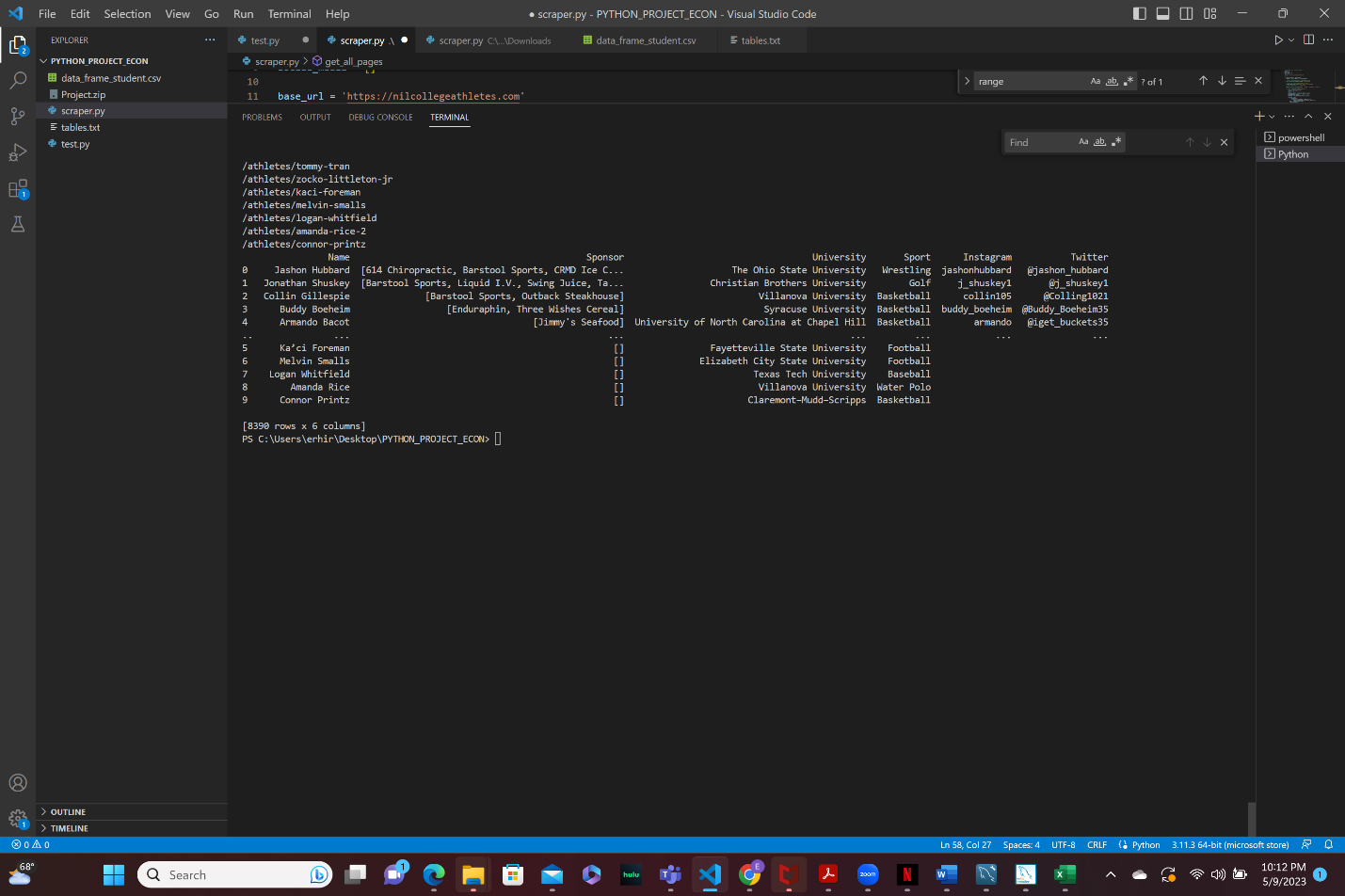
    athlete\_base\_url = "https://nilcollegeathletes.com/athletes"

    url = "https://nilcollegeathletes.com/athletes?page="

    get\_all\_pages(url)

main()

Screen Shots of My Dash board and Terminal



A screen shot of a computer

Description automatically generated with medium confidence

A computer screen shot of a program

Description automatically generated with low confidence

A picture containing text, computer, screenshot, display

Description automatically generated

A screen shot of a computer screen

Description automatically generated with medium confidence

EXPEREIENCE AND CHALLENGE

Watching YouTube videos made it look easy because the website they were scrapping from was easy, less complex and straightforward. Which was my major challenge because the website I was scrapping was very complicated, had a lot of data and layers upon layers upon layers of html, getting the right html was hard it took me weeks to get the html that would give me the precise data I want.

Lastly joining the table was hard. I had to study my data and do some research on how to join data from different pages of the website.

Link of My Research

<https://medium.com/geekculture/web-scraping-tables-in-python-using-beautiful-soup-8bbc31c5803e>

<https://www.projectpro.io/recipes/scrape-table-from-webpage-beautiful-soup>

<https://medium.com/analytics-vidhya/webscraping-a-site-with-pagination-using-beautifulsoup-fa0a09804445>