دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

درس یادگیری ماشین

پروژه گارگاه آموزشی مرتب‌سازی داده

نام و نام خانوادگی : پرهام زیلوچیان مقدم

شماره دانشجویی : 810198304

بهمن ماه 1399

فهرست

مقدمه

امروزه بیش از پیش اهمیت داده‌ها و استفاده از آن‌ها مشخص شده است. از این داده‌ها برای بهبود عملکرد سیستم‌ها و همچنین پیش‌بینی آینده و بسیاری دیگر از کارها استفاده می‌شود.

اما خب این داده‌ها همواره آماده وسرراست نیستند که ما بتوانیم به راحتی از آن‌ها استفاده کنیم. و خیلی از اوقات نیاز است که آن‌ها را از منابع مختلف جمع‌آوری کنیم و با هم ترکیب کنیم و نیز آن‌ها را تمیز کنیم تا بتوانیم که اطلاعاتی را که به آن نیاز داریم را بدست بیاوریم.

امروزه با گسترش هر چه بیشتر اینترنت و وبسایت‌های مختلف که در هر کدام از این منابع اطلاعات بسیار زیادی قرار می‌گیرد، اهمیت استفاده از داده‌هایی که در این وبسایت‌ها وجود دارد بیش از پیش شده است و خیلی از این وبسایت‌ها می‌توانند که منبع بسیار بزرگ و با اهمیت از داده‌هایی که نیاز داریم باشند.

به عنوان مثال امروزه با گسترش شبکه‌های اجتماعی ما می‌توانیم که با استخراج اطلاعات از این وبسایت‌ها اطلاعات بسیار جامعی را از افراد مختلف بدست بیاوریم و به عنوان مثال از این داده‌ها برای پیدا کردن یک متخصص که در کاری به آن نیاز داریم استفاده کنیم.

در این پروژه نیز از ما خواسته شده است که داده‌های موجود در یک وبسایت ورزشی را استخراج کنیم و از آن‌ها بیاییم و و اطلاعات مفیدی را که به ما جهت پیش‌بینی برنده یک لیگ کمک می‌کند را استخراج کنیم.

در این پروژه از ما خواسته شده است تا داده‌های مختلفی که از طریق Crowler از وبسایت ورزش3 بدست آمده است را ما استخراج کنیم و از آن‌ها اطلاعاتی را استخراج کنیم.

حال در ادامه تک‌تک به بررسی صفحات مختلفی که در اختیار ما قرار داده شده است می‌پردازیم.

آماده‌سازی داده‌ها برای استخراج :

در این قسمت توضیحاتی در این مورد را می‌دهم که من داده‌ها را جهت سادگی کار در گوگل کولب آماده می‌کنم.

ابتدا من داده‌هایی را که دارم در گیت شخصی خود بارگزاری می‌کنم.

اکنون در ادامه در گوگل کولب به صورت زیر اقدام به خواندن این دستورات می‌کنم:

```
!git clone https://github.com/parhamzm/Varzesh3-HTML-Crowler-Data.git
```

```
Cloning into 'Varzesh3-HTML-Crowler-Data'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.
```

شکل 1: خواندن داده‌ها از گیت شخصی خودم

سپس در ادامه چون داده‌ها به صورت فایل زیپ هستند، اقدام به unzip کردن آن‌ها می‌کنم و این کار را به صورت زیر انجام می‌دهم:

```
DIR_PATH = "Varzesh3-HTML-Crowler-Data/"
!unzip -o Varzesh3-HTML-Crowler-Data/varzesh3_data.zip -d ./
```

```
Archive:  Varzesh3-HTML-Crowler-Data/varzesh3_data.zip
  inflating: ./Jam-e-hazfi.csv
  inflating: ./tables.csv
  inflating: ./teams.csv
  inflating: ./transfers.csv
  inflating: ./WorldCup.csv
```

شکل 2: extract کردن فایل‌ها و دیتاست از فایل زیپ

همچنین خوب است که در این قسمت اشاره‌ای نیز به کتابخانه‌هایی که در این پروژه برای استخراج ویژگی‌ها استفاده کرده‌ام نیز بکنم:

```
import pandas as pd
import numpy as np
import csv
from bs4 import BeautifulSoup
import datetime
import matplotlib.pyplot as plt
%matplotlib inline
```

شکل 3: کتابخانه‌های استفاده شده در پروژه

اکنون در ادامه تک‌تک دیتاست‌ها و داده‌ها را مورد بررسی قرار می‌دهم:

داده‌های جام حذفی :

داده‌هایی که از جام حذفی در اختیار ما قرار داده شده است به صورت زیر می‌باشد:

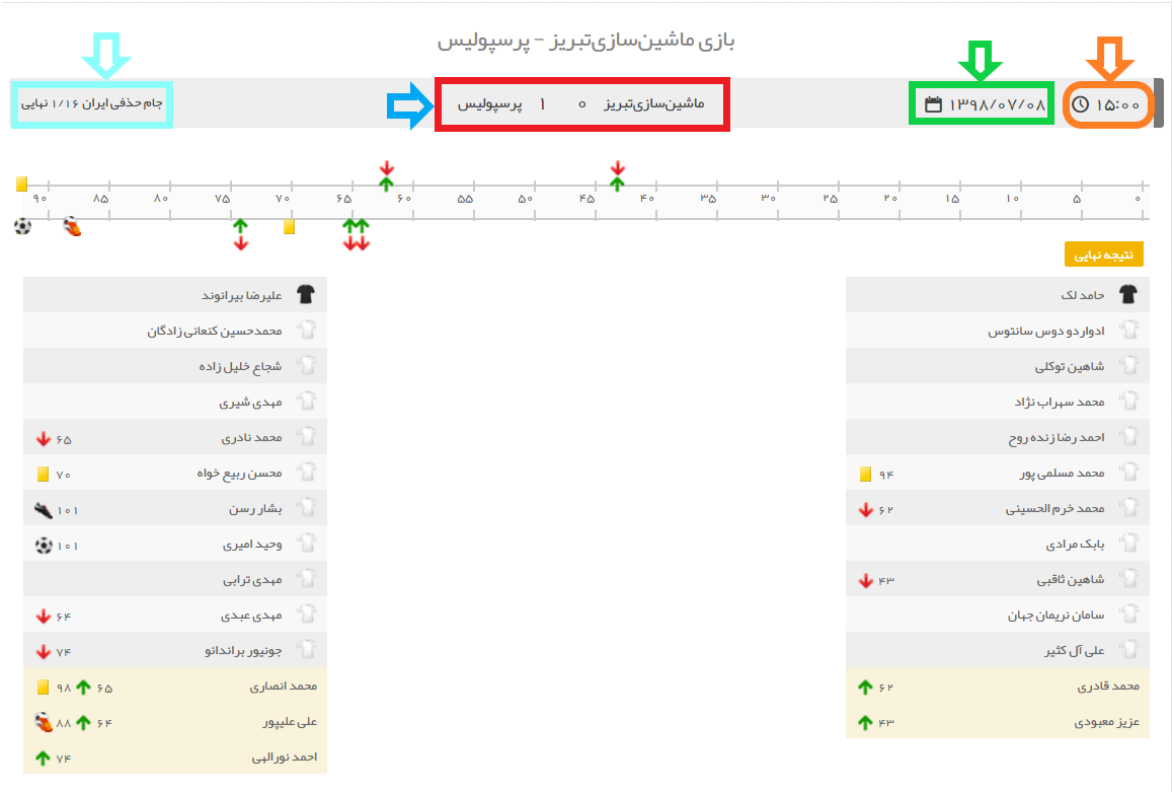
* Jame Hazfi:

```
[32]  1  jam_hazfi = pd.read_csv('Jam-e-hazfi.csv')
      2  pd.set_option("display.max_rows", None)
      3  display(jam_hazfi)
```

	Unnamed: 0	Year	Stage	html
0	0	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
1	1	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
2	2	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
3	3	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
4	4	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
5	5	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
6	6	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
7	7	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
8	8	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
9	9	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
10	10	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
11	11	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
12	12	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
13	13	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
14	14	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
15	15	جام حذفی ایران 98-99	یک شانزدهم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
16	16	جام حذفی ایران 98-99	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...

شکل 4: تعدادی از نمونه داده‌های موجود در فایل جام حذفی

سپس در ادامه من به عنوان نمونه صفحه سایت یکی از این موارد را باز می‌کنم و در ادامه قرار می‌دهم:



شکل 5: نمونه یک صفحه جام حذفی

همان‌طور که در شکل بالا می‌توانید که مشاهده کنید و من مشخص نیز کرده‌ام این اطلاعات را می‌توانیم که از این صفحه استخراج کنیم. یعنی اطلاعاتی مانند نام دو تیم شرکت‌کننده در بازی و همچنین تعداد گل‌های زده شده در هر یک از طرفین بازی و همچنین اطلاعاتی مانند ساعت و تاریخ انجام بازی. این‌ها مواردی هستند که ما از این صفحه می‌توانیم که استخراج کنیم.

ابتدا من یک جدول خالی برای مشخص کردن ستون‌های مورد نیاز به صورت زیر تشکیل می‌دهم:

```
todays_date = datetime.datetime.now().date()
index = np.array([np.arange(len(jam_hazfi.index))]).flatten()

columns = ['Type', 'Year', 'Stage', 'Team Right', 'Team Left', 'Score Right', 'Score Left', 'Time', 'Date', 'Penalty Right', 'Penalty Left']
df_jam_hazfi = pd.DataFrame(index=index, columns=columns)

```

شکل 6: تنظیم جدول برای استخراج داده‌ها

برای استخراج این داده‌ها به صورت زیر و با دستورات زیر این کار را انجام می‌دهم:

```python
for i in range(len(jam_hazfi.index)):
    soup_temp = BeautifulSoup(jam_hazfi.html[i], 'html.parser')
    team_left = soup_temp.find_all(class_='team-name left')[0].get_text()
    team_right = soup_temp.find_all(class_='team-name right')[0].get_text()
    score_left = soup_temp.find_all(class_='team-score left')[0].get_text()
    score_right = soup_temp.find_all(class_='team-score right')[0].get_text()
    time = soup_temp.find_all(class_='match-time pull-right')[0].get_text()
    date = soup_temp.find_all(class_='match-date pull-right')[0].get_text()
    df_jam_hazfi.at[i, 'Team Left'] = team_left
    df_jam_hazfi.at[i, 'Team Right'] = team_right

    df_jam_hazfi.at[i, 'Score Left'] = score_left[-1]
    df_jam_hazfi.at[i, 'Score Right'] = score_right[0]

    df_jam_hazfi.at[i, 'Stage'] = jam_hazfi.Stage[i]
    df_jam_hazfi.at[i, 'Year'] = jam_hazfi.Year[i][15:20]
    df_jam_hazfi.at[i, 'Type'] = jam_hazfi.Year[i][0:15]
    df_jam_hazfi.at[i, 'Time'] = time
    df_jam_hazfi.at[i, 'Date'] = date

    start = score_left.find("(") + len("(")
    # print(start)
    end = score_left.find(")")
    penalty_left = score_left[start:end]
    if start is 0:
        df_jam_hazfi.at[i, 'Penalty Left'] = 0
    else:
        df_jam_hazfi.at[i, 'Penalty Left'] = penalty_left

    start = score_right.find("(") + len("(")
    end = score_right.find(")")
    penalty_right = score_right[start:end]
    if start is 0:
        df_jam_hazfi.at[i, 'Penalty Right'] = 0
    else:
        df_jam_hazfi.at[i, 'Penalty Right'] = penalty_right
df_jam_hazfi
```

شکل 7: دستورات برای استخراج ویژگی‌های جام حذفی

سپس اکنون در ادامه من تعدادی از نتایج بدست آمده را قرار می‌دهم:

	Type	Year	Stage	Team Right	Team Left	Score Right	Score Left	Time	Date	Penalty Right	Penalty Left
0	جام حذفی ایران	98-99	یک شانزدهم نهایی	ماشین‌سازی‌تبریز	پرسپولیس	0	1	15:00	1398/07/08	0	0
1	جام حذفی ایران	98-99	یک شانزدهم نهایی	فشفایی شیراز	صنعت نفت آبادان	1	2	15:00	1398/07/07	0	0
2	جام حذفی ایران	98-99	یک شانزدهم نهایی	استقلال ماهشهر	مس نوین کرمان	2	1	16:00	1398/07/08	0	0
3	جام حذفی ایران	98-99	یک شانزدهم نهایی	سردار بوکان	شهرداری ماهشهر	1	2	15:00	1398/07/09	0	0
4	جام حذفی ایران	98-99	یک شانزدهم نهایی	خیبر خرم‌آباد	سپاهان	0	3	16:30	1398/07/08	0	0
5	جام حذفی ایران	98-99	یک شانزدهم نهایی	پیکان	شهدای رزکان کرج	4	0	17:15	1398/07/08	0	0
6	جام حذفی ایران	98-99	یک شانزدهم نهایی	فجر سپاسی	فولاد	2	2	15:00	1398/07/08	3	2
7	جام حذفی ایران	98-99	یک شانزدهم نهایی	گل ریحان البرز	استقلال	1	3	18:15	1398/07/08	0	0
8	جام حذفی ایران	98-99	یک شانزدهم نهایی	نساجی مازندران	شهر خودرو	0	1	15:30	1398/07/07	0	0
9	جام حذفی ایران	98-99	یک شانزدهم نهایی	گل گهرسیرجان	نفت مسجدسلیمان	0	1	16:20	1398/07/07	0	0
10	جام حذفی ایران	98-99	یک شانزدهم نهایی	خوشه طلایی ساوه	شاهین شهرداری‌بوشهر	3	3	17:15	1398/07/08	1	3
11	جام حذفی ایران	98-99	یک شانزدهم نهایی	سایپا	داماش گیلان	2	1	17:15	1398/07/07	0	0
12	جام حذفی ایران	98-99	یک شانزدهم نهایی	پاس	تراکتور	1	2	15:30	1398/07/07	0	0
13	جام حذفی ایران	98-99	یک شانزدهم نهایی	نیروی زمینی	نود ارومیه	0	2	15:30	1398/07/07	0	0
14	جام حذفی ایران	98-99	یک شانزدهم نهایی	پارس جنوبی جم	ذوب آهن	1	2	18:30	1398/07/08	0	0
15	جام حذفی ایران	98-99	یک شانزدهم نهایی	مس کرمان	بادران تهران	1	1	15:00	1398/07/06	4	2
16	جام حذفی ایران	98-99	یک هشتم نهایی	پرسپولیس	صنعت نفت آبادان	1	0	16:15	1398/09/05	0	0
17	جام حذفی ایران	98-99	یک هشتم نهایی	شهرداری ماهشهر	استقلال ماهشهر	2	1	15:30	1398/07/25	0	0
18	جام حذفی ایران	98-99	یک هشتم نهایی	سپاهان	پیکان	2	1	16:20	1398/09/05	0	0
19	جام حذفی ایران	98-99	یک هشتم نهایی	استقلال	فجر سپاسی	3	0	16:50	1398/07/25	0	0
20	جام حذفی ایران	98-99	یک هشتم نهایی	شهر خودرو	نفت مسجدسلیمان	0	1	16:20	1398/07/24	0	0

شکل 8: جدول بدست آمده از ویژگی‌های استخراج شده در جام حذفی

سپس در ادامه اقدام به ذخیره این فایل با فرمت csv. می‌کنم که بتوان از آن بعدا استفاده کرد.

برای ذخیره با فرمت csv از دستور زیر استفاده کردم:

```
38
39  df_jam_hazfi.to_csv("jam_hazfi_data.csv", index=False)
```

شکل 9: ذخیره نتایج و داده‌های استخراج شده با فرمت csv.

این فایل بدست آمده را من در پوشه Extracted Data قرار می‌دهم که همراه مابقی فایل‌های پروژه آپلود می‌شود.

استخراج داده‌های تیم‌ها :

داده‌هایی که به ما در این مجموعه داده شده است به صورت زیر هستند:

*** Teams:**

```
1  teams = pd.read_csv('teams.csv')
2  display(teams)
```

	Unnamed: 0	Competition	team	html
0	0	آرشیو جدول های لیگ برتر ایران	گل گهرسیرجان	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
1	1	آرشیو جدول های لیگ برتر ایران	نساجی مازندران	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
2	2	آرشیو جدول های لیگ برتر ایران	فولاد	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
3	3	آرشیو جدول های لیگ برتر ایران	پرسپولیس	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
4	4	آرشیو جدول های لیگ برتر ایران	شهر خودرو	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
5	5	آرشیو جدول های لیگ برتر ایران	استقلال	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
6	6	آرشیو جدول های لیگ برتر ایران	صنعت نفت آبادان	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
7	7	آرشیو جدول های لیگ برتر ایران	سپاهان	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
8	8	آرشیو جدول های لیگ برتر ایران	ذوب آهن	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
9	9	آرشیو جدول های لیگ برتر ایران	تراکتور	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
10	10	آرشیو جدول های لیگ برتر ایران	سایپا	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
11	11	آرشیو جدول های لیگ برتر ایران	نفت مسجدسلیمان	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
12	12	آرشیو جدول های لیگ برتر ایران	ماشین‌سازی‌تبریز	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
13	13	آرشیو جدول های لیگ برتر ایران	پیکان	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
14	14	آرشیو جدول های لیگ برتر ایران	آلومینیوم اراک	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
15	15	آرشیو جدول های لیگ برتر ایران	مس رفسنجان	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
16	16	آرشیو جدول های لیگ برتر ایران	پارس جنوبی جم	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
17	17	آرشیو جدول های لیگ برتر ایران	شاهین شهرداری‌بوشهر	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
18	18	آرشیو جدول های لیگ برتر ایران	سپیدرود رشت	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...

شکل 10: داده‌های داده شده به ما برای تیم‌ها

در ادامه می‌توانید صفحه وب یکی از این تیم‌ها را به عنوان نمونه مشاهده کنید:

شکل 11: نمونه صفحه یک تیم

همان‌طور که در شکل بالا می‌توانید مشاهده کنید و ما در آن مشخص کرده‌ایم ما قصد داریم تا داده‌های مربوط به این قسمت مشخص شده را تنها استخراج کنیم.

بدین منظور به صورت زیر عمل می‌کنیم:

اکنون در ادامه با دستورات زیر اقدام به تشکیل یک جدول و ستون‌های موردنیاز برای آن می‌پردازیم:

```
index = np.array([np.arange(len(teams.index))]).flatten()

columns = ['Competition', 'Team', 'Playes', 'Scores']
df_ = pd.DataFrame(index=index, columns=columns)
```

شکل 12:تشکیل جدول تیم‌ها و ستون‌های مورد نیاز برای آن‌ها

اکنون با کد زیر اقدام به استخراج داده‌های موجود در هر کدام از این صفحات HTML می‌کنم و آن‌ها را در جدولی که پیش‌تر تشکیل دادم قرار می‌دهم. و این کار را به صورت زیر انجام می‌دهم:

```python
for i in range(len(teams.index)):
    soup_temp = BeautifulSoup(teams.html[i], 'html.parser')
    team = teams.team[i]
    table = soup_temp.find('table') #, attrs={'class':'lineItemsTable'})
    if table is not None:
        table_body = table.find('tbody')

        rows = table_body.find_all('tr')
        for row in rows:
            cols = row.find_all('td')
            cols = [ele.text.strip() for ele in cols]
            if cols[1] == team:
                # print(cols)
                # print(cols[2], cols[3])
                df_.at[i, 'Playes'] = cols[2]
                df_.at[i, 'Scores'] = cols[3]
                break

    df_.at[i, 'Competition'] = teams.Competition[i]
    df_.at[i, 'Team'] = team
df_.dropna(inplace=True)
df_.reset_index(inplace=True, drop=True)
display(df_)
```

شکل 13: کد استخراج ویژگی‌های موردنیاز برای تیم

سپس در ادامه می‌توانید که نتایج بدست آمده از اجرای این مجموعه دستورات را مشاهده نمایید:

	Competition	Team	Playes	Scores
0	آرشیو جدول های لیگ برتر ایران	گل گهرسیرجان	2	6
1	آرشیو جدول های لیگ برتر ایران	نساجی مازندران	2	4
2	آرشیو جدول های لیگ برتر ایران	فولاد	2	4
3	آرشیو جدول های لیگ برتر ایران	پرسپولیس	2	4
4	آرشیو جدول های لیگ برتر ایران	شهر خودرو	1	3
5	آرشیو جدول های لیگ برتر ایران	استقلال	2	3
6	آرشیو جدول های لیگ برتر ایران	صنعت نفت آبادان	2	3
7	آرشیو جدول های لیگ برتر ایران	سپاهان	2	3
8	آرشیو جدول های لیگ برتر ایران	ذوب آهن	2	2
9	آرشیو جدول های لیگ برتر ایران	تراکتور	2	2
10	آرشیو جدول های لیگ برتر ایران	سایپا	2	2
11	آرشیو جدول های لیگ برتر ایران	نفت مسجدسلیمان	2	2
12	آرشیو جدول های لیگ برتر ایران	ماشین‌سازی‌تبریز	2	1
13	آرشیو جدول های لیگ برتر ایران	پیکان	1	0
14	آرشیو جدول های لیگ برتر ایران	آلومینیوم اراک	2	0
15	آرشیو جدول های لیگ برتر ایران	مس رفسنجان	2	0
16	آرشیو جدول های لیگ برتر ایران	پارس جنوبی جم	30	30
17	آرشیو جدول های لیگ برتر ایران	شاهین شهرداری‌بوشهر	30	22
18	آرشیو جدول های لیگ برتر ایران	سپیدرود رشت	30	20
19	آرشیو جدول های لیگ برتر ایران	استقلال خوزستان	30	12
20	آرشیو جدول های لیگ برتر ایران	گسترش فولاد	30	37

شکل 14: قسمتی از نتایج بدست آمده برای جدول تیم‌ها

اکنون در ادامه ما با دستورات زیر اقدام به ذخیره کردن نتایج بدست آمده در یک فایل csv. می‌کنیم تا بتوانیم که از آن‌ها بعدا استفاده کنیم.

بدین منظور و برای ذخیره نتایج در فرمت فایل csv از دستور زیر من استفاده می‌کنم:

```
24
25  df_teams.to_csv("teams_data.csv", index=False)
```

شکل 15: دستوره استفاده شده برای ذخیره فایل csv. داده‌های استخراج شده و نتایج بدست آمده

و شما می‌توانید که این فایل را در پوشه Extracted Data همراه فایل‌های آپلود شده مشاهده نمایید.

استخراج داده‌های جدول‌ها:

ابتدا من اقدام به لود کردن مجموعه دیتاست داده شده برای این قسمت می‌پردازم و این کار را به صورت زیر انجام می‌دهم:

*** Tables:**

```
1 tables = pd.read_csv('tables.csv')
2 display(tables)
```

	Unnamed: 0	Competition	Year	html
0	0	آرشیو جدول های لیگ برتر ایران	لیگ برتر (99-00)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
1	1	آرشیو جدول های لیگ برتر ایران	لیگ برتر (98-99)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
2	2	آرشیو جدول های لیگ برتر ایران	لیگ برتر (97-98)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
3	3	آرشیو جدول های لیگ برتر ایران	لیگ برتر (96-97)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
4	4	آرشیو جدول های لیگ برتر ایران	لیگ برتر (95-96)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
5	5	آرشیو جدول های لیگ برتر ایران	لیگ برتر (94-95)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
6	6	آرشیو جدول های لیگ برتر ایران	لیگ برتر (93-94)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
7	7	آرشیو جدول های لیگ برتر ایران	لیگ برتر (92-93)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
8	8	آرشیو جدول های لیگ برتر ایران	لیگ برتر (91-92)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
9	9	آرشیو جدول های لیگ برتر ایران	لیگ برتر (90-91)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
10	10	آرشیو جدول های لیگ دسته یک	99-98	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
11	11	آرشیو جدول های لیگ دسته یک	97-98	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
12	12	آرشیو جدول های لیگ دسته یک	96-97	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
13	13	آرشیو جدول های لیگ دسته یک	95-96	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
14	14	آرشیو جدول های لیگ دسته یک	94-95	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
15	15	آرشیو جدول های لیگ دسته یک	93-94 (گروه الف)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
16	16	آرشیو جدول های لیگ دسته یک	93-94 (گروه ب)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
17	17	آرشیو جدول های لیگ دسته یک	92-93 (گروه الف)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
18	18	آرشیو جدول های لیگ دسته یک	92-93 (گروه ب)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...

شکل 16: لود کردن داده‌های موجود در این مجموعه داده

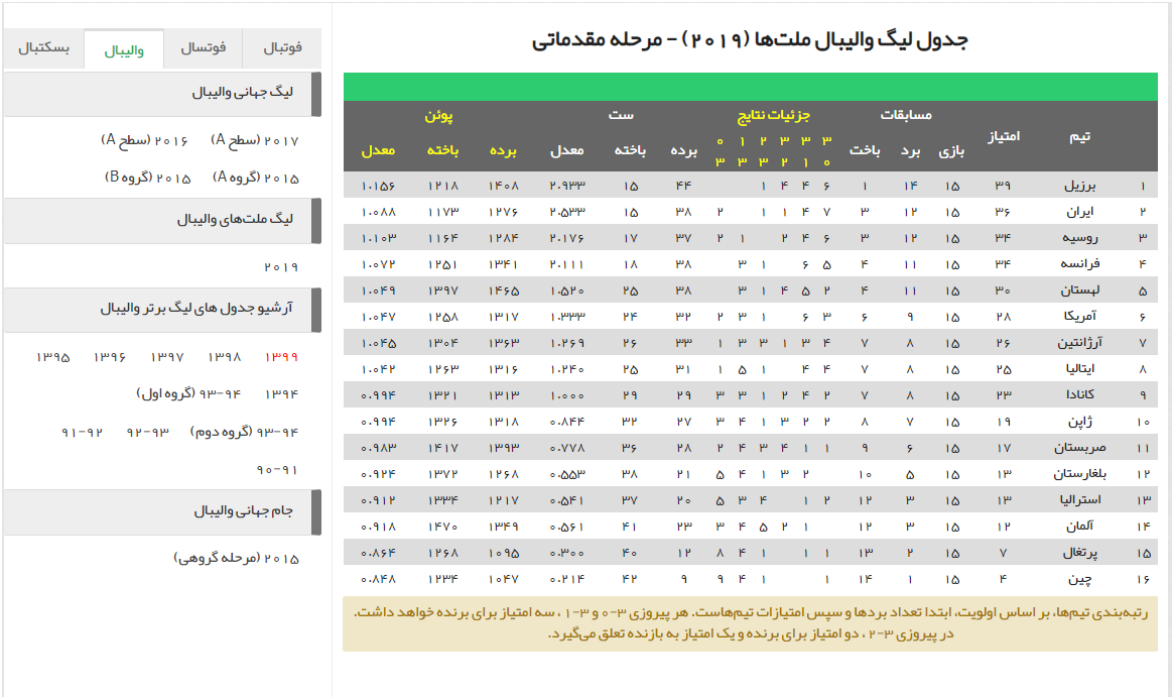
سپس در ادامه من صفحه وب یکی از این نمونه‌ها را به عنوان مثال قرار می‌دهم از یکی از لیگ‌های فوتبال:

شکل 17: نمونه یک صفحه یکی از لیگ‌های فوتبال

در ادامه من جدول یک لیگ بسکتبال را قرار می‌دهم:

شکل 18: نمونه جدول یک لیگ بسکتبال

سپس در ادامه نیز یک نمونه از لیگ والیبال را قرار می‌دهم:



فوتبال | فوتسال | والیبال | بسکتبال

لیگ جهانی والیبال

۲۰۱۷ (سطح A) ۲۰۱۶ (سطح A)

۲۰۱۵ (گروه A) ۲۰۱۵ (گروه B)

لیگ ملت‌های والیبال

۲۰۱۹

آرشیو جدول های لیگ برتر والیبال

۱۳۹۹ ۱۳۹۸ ۱۳۹۷ ۱۳۹۶ ۱۳۹۵

۱۳۹۴ ۹۳-۹۴ (گروه اول)

۹۴-۹۳ (گروه دوم) ۹۳-۹۲ ۹۲-۹۱

۹۱-۹۰

جام جهانی والیبال

۲۰۱۵ (مرحله گروهی)

جدول لیگ والیبال ملت‌ها (۲۰۱۹) - مرحله مقدماتی

	تیم	امتیاز	مسابقات				جزئیات نتایج						ست			پوئن	
			بازی	برد	باخت		۳-۰	۳-۱	۳-۲	۲-۳	۱-۳	۰-۳	برده	باخته	معدل	برده	باخته	معدل
۱	برزیل	۳۹	۱۵	۱۴	۱		۶	۴	۴	۱			۴۴	۱۵	۲.۹۳۳	۱۴۰۸	۱۲۱۸	۱.۱۵۶
۲	ایران	۳۶	۱۵	۱۲	۳		۷	۴	۱	۱		۲	۳۸	۱۵	۲.۵۳۳	۱۲۷۶	۱۱۷۳	۱.۰۸۸
۳	روسیه	۳۴	۱۵	۱۲	۳		۶	۴	۲		۱	۲	۳۷	۱۷	۲.۱۷۶	۱۲۸۴	۱۱۶۴	۱.۱۰۳
۴	فرانسه	۳۴	۱۵	۱۱	۴		۵	۶		۱	۳		۳۸	۱۸	۲.۱۱۱	۱۳۴۱	۱۲۵۱	۱.۰۷۲
۵	لهستان	۳۰	۱۵	۱۱	۴		۲	۵	۴	۱	۳		۳۸	۲۵	۱.۵۲۰	۱۴۶۵	۱۳۹۷	۱.۰۴۹
۶	آمریکا	۲۸	۱۵	۹	۶		۳	۶	۱		۳	۲	۳۲	۲۴	۱.۳۳۳	۱۳۱۷	۱۲۵۸	۱.۰۴۷
۷	آرژانتین	۲۶	۱۵	۸	۷		۴	۳	۱	۳	۳	۱	۳۳	۲۶	۱.۲۶۹	۱۳۶۳	۱۳۰۴	۱.۰۴۵
۸	ایتالیا	۲۵	۱۵	۸	۷		۴	۴		۱	۵	۱	۳۱	۲۵	۱.۲۴۰	۱۳۱۶	۱۲۶۳	۱.۰۴۲
۹	کانادا	۲۳	۱۵	۸	۷		۲	۴	۲	۱	۳	۳	۲۹	۲۹	۱.۰۰۰	۱۳۱۳	۱۳۲۱	۰.۹۹۴
۱۰	ژاپن	۱۹	۱۵	۷	۸		۲	۲	۳	۱	۴	۳	۲۷	۳۲	۰.۸۴۴	۱۳۱۸	۱۳۲۶	۰.۹۹۴
۱۱	صربستان	۱۷	۱۵	۶	۹		۱	۱	۴	۳	۴	۲	۲۸	۳۶	۰.۷۷۸	۱۳۹۳	۱۴۱۷	۰.۹۸۳
۱۲	بلغارستان	۱۳	۱۵	۵	۱۰		۲		۳	۱	۴	۵	۲۱	۳۸	۰.۵۵۳	۱۲۶۸	۱۳۷۲	۰.۹۲۴
۱۳	استرالیا	۱۳	۱۵	۳	۱۲		۲	۱		۴	۳	۵	۲۰	۳۷	۰.۵۴۱	۱۲۱۷	۱۳۳۴	۰.۹۱۲
۱۴	آلمان	۱۲	۱۵	۳	۱۲		۱		۲	۵	۴	۳	۲۳	۴۱	۰.۵۶۱	۱۳۴۹	۱۴۷۰	۰.۹۱۸
۱۵	پرتغال	۷	۱۵	۲	۱۳		۱	۱		۱	۴	۸	۱۲	۴۰	۰.۳۰۰	۱۰۹۵	۱۲۶۸	۰.۸۶۴
۱۶	چین	۴	۱۵	۱	۱۴		۱			۱	۴	۹	۹	۴۲	۰.۲۱۴	۱۰۴۷	۱۲۳۴	۰.۸۴۸

رتبه‌بندی تیم‌ها، بر اساس اولویت، ابتدا تعداد بردها و سپس امتیازات تیم‌هاست. هر پیروزی ۳-۰ و ۳-۱ ، سه امتیاز برای برنده خواهد داشت.
در پیروزی ۳-۲ ، دو امتیاز برای برنده و یک امتیاز به بازنده تعلق می‌گیرد.

شکل 19: نمونه جدول لیگ والیبال

همچنین در ادامه نیز من جدول فوتسال را نیز قرار می‌دهم:

فوتبال | فوتسال | والیبال | بسکتبال

آرشیو جدول های لیگ برتر فوتسال

۱۳۹۹ (گروه الف) ۱۳۹۹ (گروه ب) ۱۳۹۸

۱۳۹۷ ۱۳۹۶ ۹۵-۹۶ ۹۴-۹۵

۹۳-۹۴ ۹۲-۹۳ ۹۱-۹۲ ۹۰-۹۱

جدول فوتسال (۱۳۹۹) - گروه الف

فوتسال گروه الف

	تیم	بازیها	برد	مساوی	باخت	گل زده	گل خورده	تفاضل گل	امتیاز
۱	مقاومت البرز	۹	۶	۱	۲	۳۱	۲۰	۱۱	۱۹
۲	گیتی‌پسند	۸	۵	۲	۱	۲۷	۲۳	۴	۱۷
۳	کراپ الوند	۸	۴	۲	۲	۲۸	۲۲	۶	۱۴
۴	اهورا بهبهان	۹	۳	۲	۴	۱۵	۲۰	-۵	۱۱
۵	فردوس قم	۹	۲	۳	۴	۲۱	۲۴	-۳	۹
۶	شهید منصوری	۹	۲	۱	۶	۱۹	۲۶	-۷	۷
۷	شهروند ساری	۸	۱	۳	۴	۲۴	۳۰	-۶	۶

شکل 20: نمونه یک لیگ فوتسال

همان‌طور که در موارد بالا می‌توانید به خوبی مشاهده کنید، جدول‌های موارد بالا به غیر از فوتسال و فوتبال بقیه با هم کمی متفاوت است و ما نمی‌توانیم که همگی را در یک جدول جا دهیم! به همین دلیل ما برای فوتبال، والیبال و بسکتبال جداول جداگانه تشکیل می‌دهیم و آن‌ها را می‌کشیم.

قبل از این که به سراغ رسم این جدول‌ها برویم ابتدا داده‌ای که داریم را به صورت زیر نمایش می‌دهم و مشاهده می‌کنم:

*** Tables:**

```
[24]  1  tables = pd.read_csv('tables.csv')
      2  display(tables)
```

	Unnamed: 0	Competition	Year	html
0	0	آرشیو جدول های لیگ برتر ایران	لیگ برتر (00-99)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
1	1	آرشیو جدول های لیگ برتر ایران	لیگ برتر (99-98)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
2	2	آرشیو جدول های لیگ برتر ایران	لیگ برتر (98-97)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
3	3	آرشیو جدول های لیگ برتر ایران	لیگ برتر (97-96)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
4	4	آرشیو جدول های لیگ برتر ایران	لیگ برتر (96-95)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
5	5	آرشیو جدول های لیگ برتر ایران	لیگ برتر (95-94)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
6	6	آرشیو جدول های لیگ برتر ایران	لیگ برتر (94-93)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
7	7	آرشیو جدول های لیگ برتر ایران	لیگ برتر (93-92)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
8	8	آرشیو جدول های لیگ برتر ایران	لیگ برتر (92-91)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
9	9	آرشیو جدول های لیگ برتر ایران	لیگ برتر (91-90)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
10	10	آرشیو جدول های لیگ دسته یک	99-98	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
11	11	آرشیو جدول های لیگ دسته یک	97-98	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
12	12	آرشیو جدول های لیگ دسته یک	96-97	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
13	13	آرشیو جدول های لیگ دسته یک	95-96	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
14	14	آرشیو جدول های لیگ دسته یک	94-95	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
15	15	آرشیو جدول های لیگ دسته یک	93-94 (گروه الف)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
16	16	آرشیو جدول های لیگ دسته یک	93-94 (گروه ب)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
17	17	آرشیو جدول های لیگ دسته یک	92-93 (گروه الف)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
18	18	آرشیو جدول های لیگ دسته یک	92-93 (گروه ب)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...

شکل 21:جدول دیتاهای داده شده برای جداول مختلف

اکنون در ادامه به سراغ رسم تک‌تک این جدول‌ها بسته به نوع آن‌ها می‌روم:

قبل از این موضوع تمامی نوع‌هایی که داریم را من بدست می‌آورم و این کار را به صورت زیر انجام می‌دهم:

```
1  tables.Competition.unique()
```

```
array(['آرشیو جدول های لیگ برتر ایران', 'آرشیو جدول های لیگ دسته یک',
       'لیگ ملت ', 'لیگ جهانی والیبال\u200cهای والیبال',
       'لیگ قهرمانان آسیا 2020', 'آرشیو جدول های لیگ برتر فوتسال',
       'لیگ قهرمانان اروپا 2021-2020', 'آرشیو جدول های لیگ برتر انگلیس',
       'آرشیو جدول های بوندسلیگای آلمان',
       'آرشیو جدول های لالیگای اسپانیا', 'آرشیو جدول های سری آ ایتالیا',
       'آرشیو جدول های لوشامپیونا فرانسه', 'آرشیو جدول های اردیویسه هلند',
       'جام جهانی 2018 روسیه', 'آرشیو جدول های لیگ برتر والیبال',
       'جام جهانی والیبال', 'آرشیو جدول های لیگ برتر بسکتبال ایران'],
      dtype=object)
```

شکل 22:تمامی نوع لیگ‌ها و نوع ورزش‌های موجود در این دیتای داده شده

اکنون ابتدا به بیان نحوه استخراج و به دست آوردن جدول فوتبال می‌پردازیم:

❖ فوتبال و فوتسال:

برای انجام این کار از کد زیر استفاده می‌کنیم:

```
df_concat = []
for i in range(1, len(tables.index)):
    if tables.Competition[i].strip() in ['آرشیو جدول های لیگ برتر ایران', 'آرشیو جدول های لیگ دس', 'آرشیو جدول های لالیگای اسپانیا', 'جام جهانی 2018 روسیه', 'آرشیو جدول های بوندسلیگای آلمان', 'ته یک آرشیو جدول ', 'های سری آ ایتالیا',
        'آرشیو جدول های لیگ برتر فوتسال', 'آرشیو جدول های اردیویسه هلند', 'آرشیو جدول های لوشامپیونا فرانسه', 'لی
گ قهرمانان آسیا 2020', 'لیگ قهرمانان اروپا 2020-2021', 'آرشیو جدول های لیگ برتر انگلیس']:
        temp = pd.read_html(tables['html'][i])[0]
        temp.drop(temp.columns[0], axis=1, inplace=True)
        temp.drop(temp.columns[-1], axis=1, inplace=True)
        competition = [tables.Competition[i].strip().replace('آرشیو جدول های ', '') f
or k in range(len(temp.index))]
        temp.columns = ["Team", "Played", "Won", "Drawn", "Lost", "Goals For", "G
oals Agains", "Goal difference", "Points"]
        temp.insert(0, "Competition", competition, True)

        if tables.Competition[i].strip() == 'آرشیو جدول های لیگ برتر ایران':
            start = tables.Year[i].find("(") + 1
            end = tables.Year[i].find(")")
            year = tables.Year[i][start:end]
            years = [year for k in range(len(temp.index))]
            temp.insert(1, "Year", years, True)

        if tables.Competition[i].strip() in ['آرشیو جدول های لیگ دسته یک', 'آرشیو جدول های ب
ونسلیگای آلمان', 'آرشیو جدول های سری آ ایتالیا', 'آرشیو جدول های لالیگای اسپانیا',
        'آرشیو جدول های لیگ برتر فوتسال', 'آرشیو جدول های اردیویسه هلند', 'آرشیو جدول های لوشامپیونا فرانسه', 'آر
شیو جدول های لیگ برتر انگلیس']:
            year = tables.Year[i]
            years = [year for k in range(len(temp.index))]
            temp.insert(1, "Year", years, True)

        if tables.Competition[i].strip() == "جام جهانی 2018 روسیه":
            year = tables.Year[i]
            year = "(" +year + ")" + " 2018"
            years = [year for k in range(len(temp.index))]
            temp.insert(1, "Year", years, True)

        if tables.Competition[i].strip() == 'لیگ قهرمانان آسیا 2020':
            year = tables.Year[i]
            year = "(" +year + ")" + " 2020"
            years = [year for k in range(len(temp.index))]
            temp.insert(1, "Year", years, True)

        if tables.Competition[i].strip() == 'لیگ قهرمانان اروپا 2021-2020':
            year = tables.Year[i]
            year = "(" +year + ")" + " 2020-2021"
            years = [year for k in range(len(temp.index))]
            temp.insert(1, "Year", years, True)


        df_concat.append(temp)
        # df_concat = pd.concat([df_concat, temp[1:][:]], axis=1, ignore_index=Tr
ue)
df_football = pd.concat(df_concat)
```

```
47. df_football.reset_index(inplace=True, drop=True)
48. display(df_football)
```

پس از اجرای کد فوق می‌توانید که نتیجه به دست آمده از آن را در ادامه مشاهده کنید:

	Competition	Year	Team	Played	Won	Drawn	Lost	Goals For	Goals Agains	Goal difference	Points
0	لیگ برتر ایران	98-99	پرسپولیس	30	21	4	5	46	17	29	67
1	لیگ برتر ایران	98-99	استقلال	30	14	11	5	55	31	24	53
2	لیگ برتر ایران	98-99	فولاد	30	14	9	7	28	19	9	51
3	لیگ برتر ایران	98-99	تراکتور	30	14	8	8	31	23	8	50
4	لیگ برتر ایران	98-99	سپاهان	30	12	13	5	39	22	17	49
5	لیگ برتر ایران	98-99	شهر خودرو	30	12	10	8	27	25	2	46
6	لیگ برتر ایران	98-99	صنعت نفت آبادان	30	11	8	11	29	33	-4	41
7	لیگ برتر ایران	98-99	نفت مسجدسلیمان	30	7	17	6	24	22	2	38
8	لیگ برتر ایران	98-99	نساجی مازندران	30	8	14	8	30	32	-2	38
9	لیگ برتر ایران	98-99	گل گهرسیرجان	30	7	12	11	27	34	-7	33
10	لیگ برتر ایران	98-99	ماشین‌سازی‌تبریز	30	8	7	15	28	40	-12	31
11	لیگ برتر ایران	98-99	ذوب آهن	30	7	9	14	31	39	-8	30
12	لیگ برتر ایران	98-99	پیکان	30	6	11	13	38	44	-6	29
13	لیگ برتر ایران	98-99	سایپا	30	5	14	11	24	35	-11	29
14	لیگ برتر ایران	98-99	پارس جنوبی جم	30	4	15	11	20	30	-10	27
15	لیگ برتر ایران	98-99	شاهین شهرداری‌بوشهر	30	4	10	16	26	57	-31	22
16	لیگ برتر ایران	97-98	پرسپولیس	30	16	13	1	36	14	22	61
17	لیگ برتر ایران	97-98	سپاهان	30	15	13	2	46	20	26	58
18	لیگ برتر ایران	97-98	استقلال	30	16	9	5	40	13	27	57
19	لیگ برتر ایران	97-98	پدیده شهرخودرو	30	16	8	6	32	16	16	56
20	لیگ برتر ایران	97-98	تراکتور	30	14	10	6	42	25	17	52
21	لیگ برتر ایران	97-98	ذوب آهن	30	9	13	8	28	28	0	40
22	لیگ برتر ایران	97-98	سایپا	30	9	11	10	28	33	-5	38
23	لیگ برتر ایران	97-98	فولاد	30	9	11	10	30	39	-9	38
24	لیگ برتر ایران	97-98	صنعت نفت آبادان	30	7	16	7	31	30	1	37

شکل 23: قسمتی از جدول به دست آمده برای لیگ‌های مختلف فوتبال

سپس در ادامه به بررسی جدول برای لیگ‌های والیبال می‌پردازم:

❖ والیبال:

برای بدست آوردن جدول مربوط به لیگ‌های والیبال از کد زیر که نوشته‌ایم استفاده می‌کنیم که در ادامه آن را قرار می‌دهم:

*** Volleyball:**

```
df_concat = []
for i in range(1, len(tables.index)):
    if tables.Competition[i].strip() in ['لیگ جهانی والیبال', 'جام جهانی والیبال', 'آرشیو جدول های لیگ برتر والیبال']:
        temp = pd.read_html(tables['html'][i])[0]
        temp.drop(temp.columns[0], axis=1, inplace=True)

        competition = [tables.Competition[i].strip().replace('آرشیو جدول های ', '') for k in range(len(temp.index))]
        arrays = [["Team", "Point", "Matchs", "Matchs", "Matchs",
           "Resul Details", "Resul Details", "Resul Details", "Resul Details", "Resul Details", "Resul Details",
           "Set", "Set", "Set", "Points", "Points", "Points"],
          ["Team", "Point", "Played", "Won", "Lost", "30", "31", "32", "23", "13", "03", "Won", "Lost", "Avg", "Won", "Lost", "Avg"],]
        tuples = list(zip(*arrays))

        index = pd.MultiIndex.from_tuples(tuples, names=["first", "second"])
        temp.columns = index
        temp.insert(0, "Competition", competition, True)

        if tables.Competition[i].strip() in ['لیگ جهانی والیبال', 'جام جهانی والیبال', 'آرشیو جدول های لیگ برتر والیبال']:
            year = tables.Year[i]
            years = [year for k in range(len(temp.index))]
            temp.insert(1, "Year", years, True)


        df_concat.append(temp)
        # df_concat = pd.concat([df_concat, temp[1:][:]], axis=1, ignore_index=True)
df_volleyball = pd.concat(df_concat)
df_volleyball.reset_index(inplace=True, drop=True)
df_volleyball.fillna("-", inplace=True)
display(df_volleyball)
```

شکل 24: کد برای بدست آوردن جدول لیگ‌های مختلف والیبال موجود در این دیتا

سپس در ادامه نیز می‌توانید که نتایج و جدول بدست آمده از اجرای این کد را مشاهده کنید:

first	Competition	Year	Team	Point	Matchs			Resul Details						Set			Points		
second			Team	Point	Played	Won	Lost	30	31	32	23	13	03	Won	Lost	Avg	Won	Lost	Avg
0	لیگ جهانی والیبال	2017 (سطح A)	فرانسه	25	9	8	1	4	4	-	1	-	-	26	7	3.714	792	673	1.177
1	لیگ جهانی والیبال	2017 (سطح A)	برزیل	19	9	6	3	1	5	-	1	2	-	22	14	1.571	817	786	1.039
2	لیگ جهانی والیبال	2017 (سطح A)	صربستان	18	9	6	3	2	3	1	1	1	1	21	14	1.500	797	767	1.039
3	لیگ جهانی والیبال	2017 (سطح A)	روسیه	14	9	5	4	3	-	2	1	2	1	19	16	1.188	787	775	1.015
4	لیگ جهانی والیبال	2017 (سطح A)	کانادا	12	9	5	4	-	2	3	-	3	1	18	20	0.900	844	858	0.984
5	لیگ جهانی والیبال	2017 (سطح A)	آمریکا	14	9	4	5	3	1	-	2	3	-	19	16	1.188	816	804	1.015
6	لیگ جهانی والیبال	2017 (سطح A)	بلژیک	14	9	4	5	1	2	1	3	-	2	18	19	0.947	808	820	0.985
7	لیگ جهانی والیبال	2017 (سطح A)	لهستان	12	9	4	5	1	2	1	1	3	1	17	19	0.895	806	801	1.006
8	لیگ جهانی والیبال	2017 (سطح A)	بلغارستان	10	9	4	5	-	1	3	1	2	2	16	22	0.727	819	861	0.951
9	لیگ جهانی والیبال	2017 (سطح A)	آرژانتین	11	9	3	6	-	2	1	3	-	3	15	22	0.682	800	837	0.956
10	لیگ جهانی والیبال	2017 (سطح A)	ایران	7	9	3	6	-	1	2	-	2	4	11	23	0.478	739	795	0.930
11	لیگ جهانی والیبال	2017 (سطح A)	ایتالیا	6	9	2	7	1	-	1	1	5	1	13	23	0.565	798	846	0.943
12	لیگ جهانی والیبال	2016 (سطح A)	برزیل	31	13	11	2	5	4	2	-	1	1	34	14	2.429	1155	1021	1.131
13	لیگ جهانی والیبال	2016 (سطح A)	صربستان	30	13	10	3	4	4	2	2	-	1	34	17	2.000	1186	1102	1.076
14	لیگ جهانی والیبال	2016 (سطح A)	فرانسه	26	13	8	5	6	1	1	3	2	-	32	18	1.778	1192	1116	1.068
15	لیگ جهانی والیبال	2016 (سطح A)	آمریکا	24	11	8	3	3	4	1	1	2	-	28	15	1.867	1020	964	1.058
16	لیگ جهانی والیبال	2016 (سطح A)	ایتالیا	23	13	7	6	5	2	-	2	-	4	25	20	1.250	1044	1006	1.038
17	لیگ جهانی والیبال	2016 (سطح A)	روسیه	15	9	5	4	3	2	-	-	-	4	15	14	1.071	698	662	1.054
18	لیگ جهانی والیبال	2016 (سطح A)	لهستان	10	11	4	7	-	2	2	-	3	4	15	27	0.556	947	1013	0.935
19	لیگ جهانی والیبال	2016 (سطح A)	ایران	9	9	4	5	-	1	3	-	3	2	15	22	0.682	777	861	0.902
20	لیگ جهانی والیبال	2016 (سطح A)	بلژیک	11	9	3	6	2	-	1	3	-	3	15	20	0.750	760	779	0.976
21	لیگ جهانی والیبال	2016 (سطح A)	آرژانتین	10	9	3	6	2	-	1	2	3	1	16	20	0.800	824	835	0.987
22	لیگ جهانی والیبال	2016 (سطح A)	بلغارستان	2	9	1	8	-	-	1	-	5	3	8	26	0.308	706	810	0.872

شکل 25: نتایج بدست آمده برای لیگ والیبال

در مورد دیتای والیبال یک نکته وجود دارد که خوب است به آن اشاره کنم:

نکته این است که همان‌طور که به عنوان مثال در شکل زیر می‌توانید که مشاهده کنید فیلدهایی هستند که در جدول موجود در وبسایت وجود ندارند:

جدول لیگ والیبال ملت‌ها (۲۰۱۹) - مرحله مقدماتی

	تیم	امتیاز	مسابقات			جزئیات نتایج						ست			پوئن		
			بازی	برد	باخت	۳-۰	۳-۱	۳-۲	۲-۳	۱-۳	۰-۳	برده	باخته	معدل	برده	باخته	معدل
۱	برزیل	۳۹	۱۵	۱۴	۱	۶	۴	۴	۱			۴۴	۱۵	۲.۹۳۳	۱۴۰۸	۱۲۱۸	۱.۱۵۶
۲	ایران	۳۶	۱۵	۱۲	۳	۷	۴	۱	۱		۲	۳۸	۱۵	۲.۵۳۳	۱۲۷۶	۱۱۷۳	۱.۰۸۸
۳	روسیه	۳۴	۱۵	۱۲	۳	۶	۴	۲			۲	۳۷	۱۷	۲.۱۷۶	۱۲۸۴	۱۱۶۴	۱.۱۰۳
۴	فرانسه	۳۴	۱۵	۱۱	۴	۵	۶		۱	۳		۳۸	۱۸	۲.۱۱۱	۱۳۴۱	۱۲۵۱	۱.۰۷۲
۵	لهستان	۳۰	۱۵	۱۱	۴	۲	۵	۴	۱	۳		۳۸	۲۵	۱.۵۲۰	۱۴۶۵	۱۳۹۷	۱.۰۴۹
۶	آمریکا	۲۸	۱۵	۹	۶	۳	۶	۱	۳	۲		۳۲	۲۴	۱.۳۳۳	۱۳۱۷	۱۲۵۸	۱.۰۴۷
۷	آرژانتین	۲۶	۱۵	۸	۷	۴	۳	۱	۳	۳	۱	۳۳	۲۶	۱.۲۶۹	۱۳۶۳	۱۳۰۴	۱.۰۴۵
۸	ایتالیا	۲۵	۱۵	۸	۷	۴	۴		۱	۵	۱	۳۱	۲۵	۱.۲۴۰	۱۳۱۶	۱۲۶۳	۱.۰۴۲
۹	کانادا	۲۳	۱۵	۸	۷	۲	۴	۲	۱	۳	۳	۲۹	۲۹	۱.۰۰۰	۱۳۱۳	۱۳۲۱	۰.۹۹۴
۱۰	ژاپن	۱۹	۱۵	۷	۸	۲	۲	۳	۱	۴	۳	۲۷	۳۲	۰.۸۴۴	۱۳۱۸	۱۳۲۶	۰.۹۹۴
۱۱	صربستان	۱۷	۱۵	۶	۹	۱	۱	۴	۳	۴	۲	۲۸	۳۶	۰.۷۷۸	۱۳۹۳	۱۴۱۷	۰.۹۸۳
۱۲	بلغارستان	۱۳	۱۵	۵	۱۰	۲	۳	۱	۴	۵		۲۱	۳۸	۰.۵۵۳	۱۲۶۸	۱۳۷۲	۰.۹۲۴
۱۳	استرالیا	۱۳	۱۵	۳	۱۲	۲	۱	۴		۳	۵	۲۰	۳۷	۰.۵۴۱	۱۲۱۷	۱۳۳۴	۰.۹۱۲
۱۴	آلمان	۱۲	۱۵	۳	۱۲	۱	۲	۵	۱	۴	۳	۲۳	۴۱	۰.۵۶۱	۱۳۴۹	۱۴۷۰	۰.۹۱۸
۱۵	پرتغال	۷	۱۵	۲	۱۳	۱	۱	۱	۱	۴	۸	۱۲	۴۰	۰.۳۰۰	۱۰۹۵	۱۲۶۸	۰.۸۶۴
۱۶	چین	۴	۱۵	۱	۱۴	۱		۱	۱	۴	۹	۹	۴۲	۰.۲۱۴	۱۰۴۷	۱۲۳۴	۰.۸۴۸

رتبه‌بندی تیم‌ها، بر اساس اولویت، ابتدا تعداد بردها و سپس امتیازات تیم‌هاست. هر پیروزی ۳-۰ و ۳-۱ ، سه امتیاز برای برنده خواهد داشت. در پیروزی ۳-۲ ، دو امتیاز برای برنده و یک امتیاز به بازنده تعلق می‌گیرد.

شکل 26: فیلدهای خالی موجود در این جدول که مشخص شده‌اند

این فیلدهای خالی تقریبا در همه جدول‌ها وجود دارند و به همین منظور ما در تولید جدول نهایی این مقادیر خالی را با کاراکتر "-" جایگزین کرده‌ایم.

اکنون در ادامه به بررسی لیگ بسکتبال می‌پردازیم:

❖ بسکتبال:

در این لیگ نیز ما از کدهای زیر برای بدست آوردن نتایج استفاده کرده‌ایم:

*** Basketball:**

```
df_concat = []
for i in range(1, len(tables.index)):
    if tables.Competition[i].strip() in ['آرشیو جدول های لیگ برتر بسکتبال ایران']:
        temp = pd.read_html(tables['html'][i])[0]
        temp.drop(temp.columns[0], axis=1, inplace=True)
        temp.drop(temp.columns[-1], axis=1, inplace=True)
        temp.drop(temp.columns[-1], axis=1, inplace=True)
        competition = [tables.Competition[i].strip().replace('آرشیو جدول های ', '') for k in range(len(temp.index))]
        temp.columns = ["Team", "Played", "Won", "Lost", "Goals For", "Goals Agains", "Goal difference", "Points"]
        temp.insert(0, "Competition", competition, True)

        if tables.Competition[i].strip() in ['آرشیو جدول های لیگ برتر بسکتبال ایران']:
            year = tables.Year[i]
            years = [year for k in range(len(temp.index))]
            temp.insert(1, "Year", years, True)


        df_concat.append(temp)
        # df_concat = pd.concat([df_concat, temp[1:][:]], axis=1, ignore_index=True)
df_basketball = pd.concat(df_concat)
df_basketball.reset_index(inplace=True, drop=True)
display(df_basketball)
```

شکل 27: کدهای استفاده شده برای بدست آوردن لیگ‌های بسکتبال

اکنون در ادامه پس از اجرای این کدها می‌توانید که نتایج بدست آمده را مشاهده کنید:

	Competition	Year	Team	Played	Won	Lost	Goals For	Goals Agains	Goal difference	Points
0	لیگ برتر بسکتبال ایران	1398	شهرداری گرگان	26	24	2	1850	1413	437	50
1	لیگ برتر بسکتبال ایران	1398	پتروشیمی ب.ا	26	21	5	1929	1514	415	47
2	لیگ برتر بسکتبال ایران	1398	شیمیدر	26	20	6	2108	1831	277	46
3	لیگ برتر بسکتبال ایران	1398	پالایش نفت آبادان	26	20	6	1948	1718	230	46
4	لیگ برتر بسکتبال ایران	1398	مهرام	26	19	7	1943	1697	246	45
5	لیگ برتر بسکتبال ایران	1398	آویژه صنعت مشهد	26	18	8	2010	1866	144	44
6	لیگ برتر بسکتبال ایران	1398	اکسون	26	12	14	1935	1933	2	38
7	لیگ برتر بسکتبال ایران	1398	شهرداری بندرعباس	26	11	15	1797	1881	-84	37
8	لیگ برتر بسکتبال ایران	1398	ذوب آهن	26	11	15	1789	1872	-83	36
9	لیگ برتر بسکتبال ایران	1398	توفارفان آذرشهر	26	7	19	1575	1710	-135	33
10	لیگ برتر بسکتبال ایران	1398	مس کرمان	26	6	20	1731	2079	-348	32
11	لیگ برتر بسکتبال ایران	1398	شورا و شهرداری قزوین	26	5	21	1805	2055	-250	31
12	لیگ برتر بسکتبال ایران	1398	رعد یدافد شهرکرد	26	5	21	1667	2067	-400	31
13	لیگ برتر بسکتبال ایران	1398	نیروی زمینی	26	3	23	1537	1990	-453	28
14	لیگ برتر بسکتبال ایران	1397	پتروشیمی ب.ا	16	15	1	1388	972	416	31
15	لیگ برتر بسکتبال ایران	1397	شیمیدر	16	14	2	1209	1000	209	30
16	لیگ برتر بسکتبال ایران	1397	شهرداری گرگان	16	11	5	1148	1101	47	27
17	لیگ برتر بسکتبال ایران	1397	پالایش نفت آبادان	16	11	5	1171	1047	124	27
18	لیگ برتر بسکتبال ایران	1397	ذوب آهن	16	8	8	1068	1061	7	24
19	لیگ برتر بسکتبال ایران	1397	آویژه صنعت مشهد	16	4	12	1143	1209	-66	20
20	لیگ برتر بسکتبال ایران	1397	پگاه تهران	16	4	12	1118	1257	-139	20
21	لیگ برتر بسکتبال ایران	1397	نیروی زمینی	16	4	12	1012	1231	-219	20
22	لیگ برتر بسکتبال ایران	1397	رعد یدافد شهرکرد	16	1	14	920	1270	-350	17
23	لیگ برتر بسکتبال ایران	96-97	پتروشیمی ب.ا	16	15	1	1383	1063	320	31
24	لیگ برتر بسکتبال ایران	96-97	شهرداری تبریز	16	12	4	1336	1057	279	28

شکل 28: قسمتی از نتایج بدست آمده برای قسمت لیگ بسکتبال

اکنون در این قسمت من ذخیره نتایج هر سه این جدول‌ها را قرار می‌دهم که آن‌ها را با فرمت CSV. ذخیره می‌کنم.

ابتدا دستور مربوط به ذخیره داده‌های استخراج شده فوتبال را قرار می‌دهم:

```
49
50  df_football.to_csv("tables_football_data.csv", index=False)
```

شکل 29: ذخیره نتایج و داده‌های استخراج شده مربوط به فوتبال و فوتسال

اکنون در ادامه من دستور ذخیره داده‌های استخراج شده و نتایج بدست آمده والیبال را قرار می‌دهم:

```
30
31  df_volleyball.to_csv("tables_volleyball_data.csv", index=False)
```

شکل 30: ذخیره نتایج و داده‌های استخراج شده مربوط به والیبال

اکنون نیز من در ادامه دستور ذخیره‌سازی داده‌های استخراج شده مربوط به بسکتبال را قرار می‌دهم:

```
23
24  df_basketball.to_csv("tables_basketball_data.csv", index=False)
```

شکل 31:ذخیره نتایج و داده‌های استخراج شده مربوط بسکتبال

تمامی فایل‌های بدست آمده در بالا را می‌توانید که در پوشه Extracted Data موجود در پوشه فایل‌های آپلود شده مشاهده نمایید.

استخراج داده‌های نقل و انتقالات:

ابتدا اقدام به باز کردن فایل مربوط به این دیتاست که به ما داده شده است می‌کنیم و این کار را به صورت زیر انجام می‌دهم و نتیجه آن نیز به این صورت است:

* Transfers:

```
[10]  1  transfers = pd.read_csv('transfers.csv')
      2  display(transfers)
```

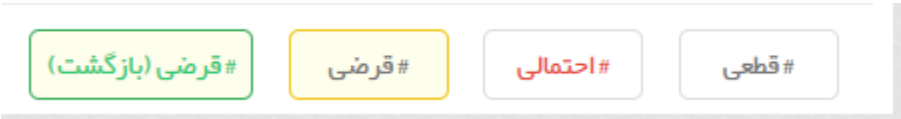
	Unnamed: 0	transfers	html
0	0	نقل و انتقالات لیگ برتر 99-00	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
1	1	نقل و انتقالات اروپا 2021-2020	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
2	2	نقل و انتقالات نیم فصل لیگ برتر 99-98	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
3	3	نقل و انتقالات اروپا 2020-2019	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
4	4	نقل و انتقالات لیگ برتر 99-98	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
5	5	نقل و انتقالات نیم فصل لیگ برتر 98-97	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
6	6	نقل و انتقالات لیگ برتر 98-97	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
7	7	نقل و انتقالات اروپا 2018-2019	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
8	8	نقل و انتقالات نیم فصل لیگ برتر 97-96	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
9	9	نقل و انتقالات لیگ برتر 97-96	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
10	10	نقل و انتقالات نیم فصل لیگ برتر 96-95	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
11	11	نقل و انتقالات لیگ برتر 96-95	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
12	12	نقل و انتقالات اروپا 2017-2016	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
13	13	نقل و انتقالات نیم فصل لیگ برتر 95-94	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
14	14	نقل و انتقالات اروپا نیم فصل 2016-2015	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
15	15	نقل و انتقالات لیگ برتر 95-94	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
16	16	نقل و انتقالات اروپا 2016-2015	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...

شکل 32:دیتاهای داده شده برای نقل و انتقالات

اکنون در ادامه نیز من تصویری از صفحه مربوط به این داده‌ها در وبسایت مربوطه را قرار می‌دهم:

شکل 33: قسمتی از صفحه مربوط به نقل و انتقالات در سایت ورزش3

همچنین خوب است به این موضوع نیز اشاره کنم که 4 حالت دارند بازیکنان جابه‌جا شده که در تصویر زیر می‌توانید که آن‌ها را مشاهده نمایید:



شکل 34: حالت‌های مختلف نقل و انتقالات

همچنین چون دو حالت مختلف بازیکنان ورودی به تیم و نیز بازیکنان خروجی از تیم را داریم این حالت‌ها به 8 حالت تبدیل می‌شوند که این موارد را من در جدولی که باید تولید کنم باید تماما پوشش بدهم.

اکنون در ادامه می‌توانید که دستورات ما برای ساخت و ایجاد این جدول و ستون‌های آن را مشاهده نمایید:

```
index = np.array([np.arange(len(transfers.index))]).flatten()

columns = ['Transfer', 'Year', 'Team', "('قطعی' ,'ورودی')",
           "('احتمالی' ,'ورودی')", "('قرضی' ,'ورودی')", "('قرضی (بازگشت)' ,'ورودی')",
           "('قطعی' ,'خروجی')", "('احتمالی' ,'خروجی')", "('قرضی' ,'خروجی')", "('قرضی (بازگشت)' ,'خروجی')"]
df_ = pd.DataFrame(index=index, columns=columns)
# df_
```

شکل 35: دستورات برای ایجاد ستون‌های جدول نقل و انتقالات

سپس اکنون در ادامه برای استخراج ویژگی‌های مربوط به نقل و انتقالات از داخل دیتای داده شده نیز از دستورات زیر ما استفاده می‌کنیم:

```
iter_main = 0
for i in range(len(transfers.index)):
    soup_temp = BeautifulSoup(transfers.html[i], 'html.parser')
    team_list = soup_temp.find_all(class_='m3g-ct-col ct-team-name')
    team_in = soup_temp.find_all('div', {'class':'m3g-ct-col ct-in'})
    team_out = soup_temp.find_all('div', {'class':'m3g-ct-col ct-out'})

    for k in range(1, len(team_list)):
        team_name = team_list[k].get_text().strip()
        permanent_in = []
        permanent_out = []
        loan_return_in = []
        loan_return_out = []
        likely_in = []
        likely_out = []
        loan_in = []
        loan_out = []
        all_in_perm = team_in[k].find_all('span', {'class':['ct-player-name ct-permanent ', 'ct-player-name ct-permanent has-cat']})
        all_in_loan_return = team_in[k].find_all('span', {'class':['ct-player-name ct-loan-b ', 'ct-player-name ct-loan has-cat ct-loan-b']})
        all_in_likely = team_in[k].find_all('span', {'class':['ct-player-name ct-permanent likely has-cat', 'ct-player-name ct-permanent likely ']})
        all_in_loan = team_in[k].find_all('span', {'class':['ct-player-name ct-loan has-cat', 'ct-player-name ct-loan ']})

        all_out_perm = team_out[k].find_all('span', {'class':['ct-player-name ct-permanent ', 'ct-player-name ct-permanent has-cat']})
        all_out_loan_return = team_out[k].find_all('span', {'class':['ct-player-name ct-loan-b ', 'ct-player-name ct-loan has-cat ct-loan-b']})
        all_out_likely = team_out[k].find_all('span', {'class':['ct-player-name ct-permanent likely has-cat', 'ct-player-name ct-permanent likely ']})
        all_out_loan = team_out[k].find_all('span', {'class':['ct-player-name ct-loan has-cat', 'ct-player-name ct-loan ']})
        items = [all_in_perm, all_in_loan_return, all_in_likely, all_in_loan, all_out_perm, all_out_loan_return, all_out_likely, all_out_loan]
        for iteration, item in enumerate(items):
            for p in range(len(item)):
                if iteration == 0:
                    permanent_in.append(item[p].get_text().strip())
                elif iteration == 1:
                    loan_return_in.append(item[p].get_text().strip())
                elif iteration == 2:
```

```
                    likely_in.append(item[p].get_text().strip())
                elif iteration == 3:
                    loan_in.append(item[p].get_text().strip())
                elif iteration == 4:
                    permanent_out.append(item[p].get_text().strip())
                elif iteration == 5:
                    loan_return_out.append(item[p].get_text().strip())
                elif iteration == 6:
                    likely_out.append(item[p].get_text().strip())
                elif iteration == 7:
                    loan_out.append(item[p].get_text().strip())
        df_.at[iter_main, 'Team'] = team_name
        year = [s for s in transfers.transfers[i].split() if s[0].isdigit()]
        transfer_name = transfers.transfers[i].replace(year[0], '')
        df_.at[iter_main, 'Transfer'] = transfer_name.strip()
        df_.at[iter_main, 'Year'] = year[0].strip()
        if not permanent_in:
            permanent_in = ['بدون ورودی قطعی']
        if not likely_in:
            likely_in = ['بدون ورودی احتمالی']
        if not loan_in:
            loan_in = ['بدون ورودی قرضی']
        if not loan_return_in:
            loan_return_in = ['بدون ورودی قرضی (بازگشت)']
        if not permanent_out:
            permanent_out = ['بدون خروجی قطعی']
        if not likely_out:
            likely_out = ['بدون خروجی احتمالی']
        if not loan_out:
            loan_out = ['بدون خروجی قرضی']
        if not loan_return_out:
            loan_return_out = ['بدون خروجی قرضی (بازگشت)']

        df_.at[iter_main, "('ورودی', 'قطعی')"] = permanent_in
        df_.at[iter_main, "('ورودی', 'احتمالی')"] = likely_in
        df_.at[iter_main, "('ورودی', 'قرضی')"] = loan_in
        df_.at[iter_main, "('ورودی', 'قرضی (بازگشت)')"] = loan_return_in
        df_.at[iter_main, "('خروجی', 'قطعی')"] = permanent_out
        df_.at[iter_main, "('خروجی', 'احتمالی')"] = likely_out
        df_.at[iter_main, "('خروجی', 'قرضی')"] = loan_out
        df_.at[iter_main, "('خروجی', 'قرضی (بازگشت)')"] = loan_return_out
        iter_main +=1
df_.reset_index(inplace=True, drop=True)
display(df_)
```

اکنون پس از اجرای این دستورات نتیجه مطابق زیر بدست می‌آید:

	Transfer	Year	Team	('ورودی', 'قطعی')	('ورودی', 'احتمالی')	('ورودی', 'قرضی')	('ورودی', 'قرضی') (بازگشت)	('خروجی', 'قطعی')	('خروجی', 'احتمالی')	('خروجی', 'قرضی')	('خروجی', 'قرضی') (بازگشت)
0	نقل و انتقالات لیگ برتر	99-00	استقلال	بابک مرادی, سید احمد موسوی, ... متین کریم زاده, م	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[سجاد آقایی]	شاهین طاهرخانی, علی دشتی, ... محسن کریمی, میلاد	[سیدحسین حسینی]	[حسین پورحمیدی, محمد بلباغی]	[بدون خروجی قرضی (بازگشت)]
1	نقل و انتقالات لیگ برتر	99-00	آلومینیوم اراک	اسماعیل شریفات, پوریا آریانکیا, ... جابر انصاری, ش	[بدون ورودی احتمالی]	[حسین پورحمیدی]	[بدون ورودی قرضی (بازگشت)]	[محمود فاتح رحمتی]	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[امیرحسین جدی]
2	نقل و انتقالات لیگ برتر	99-00	پارس جنوبی جم	[بدون ورودی قطعی]	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	پوریا آریانکیا, متین کریم زاده, محمد ... نوری, فری	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[بدون خروجی قرضی (بازگشت)]
3	نقل و انتقالات لیگ برتر	99-00	پرسپولیس	آرمان رمضانی, احسان پهلوان, ... حامد لک, سجد آقا	[فرشاد فرجی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	علیرضا بیرانوند, محمد داوری, ... شجاع خلیل زاده	[بدون خروجی احتمالی]	[امیر روستایی]	[بدون خروجی قرضی (بازگشت)]
4	نقل و انتقالات لیگ برتر	99-00	پیکان	اسماعیل بابایی, حسین حیدری, ... رضا جبیره, شاهین	[بدون ورودی احتمالی]	[امیر روستایی]	[امیرحسین جدی]	جابر انصاری, سجد واسمی, جلال ... الدین علی محمدی	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[بدون خروجی قرضی (بازگشت)]
5	نقل و انتقالات لیگ برتر	99-00	تراکتور	امین اسدی, حنیف فرعباسی, حمید ... بوحمدان, سعید و	[امین قاسمی نژاد, سیدحسین حسینی]	[بدون ورودی قرضی]	[ابوالفضل رزاق پور]	محمد رشید مظاهری, رضا اسدی, ... ساسان انصاری, ایم	[بدون خروجی احتمالی]	[ابوالفضل رزاق پور]	[بدون خروجی قرضی (بازگشت)]
6	نقل و انتقالات لیگ برتر	99-00	ذوب آهن	ابوالفضل عکاشه, دانیال مومنی, ... شهاب گردان, عجب	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	مهدی مهدی پور, احسان پهلوان, ... معراج اسماعیلی	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[سجاد آقایی]
7	نقل و انتقالات لیگ برتر	99-00	سایپا	حسین مالکی, حسین ماهینی, حمید ... گلزاری, فریبرز	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	مرتضی آقاجان, آرمان رمضانی, ... محمد شریفی, معین	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[بدون خروجی قرضی (بازگشت)]
8	نقل و انتقالات لیگ برتر	99-00	سپاهان	جلال الدین علی محمدی, دانیال ... اسماعیلی فر, روح	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	محمد ایران پوریان, سجد آقایی, ... ابوالفضل عکا	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[بدون خروجی قرضی (بازگشت)]
9	نقل و انتقالات لیگ برتر	99-00	شاهین شهرداری‌بوشهر	[بدون ورودی قطعی]	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	اسماعیل شریفات, حسین مالکی, [پیمن سالاری	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[ابوالفضل رزاق پور]
10	نقل و انتقالات لیگ برتر	99-00	شهر خودرو	احمد الجبوری, امیرحسین گریمی, ... بهداش میلانیان	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	محمدحسین مرادمند, میلاد سرلک, ... علی نبی زاده, م	[فرشاد فرجی, امین قاسمی نژاد]	[بدون خروجی قرضی]	[بدون خروجی قرضی (بازگشت)]
11	نقل و انتقالات لیگ برتر	99-00	صنعت نفت آبادان	حامد فاتح زاده, حسن بیت سجد, ... حکیم نصاری, رضا	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	مصطفی احمدی, عیسی آل کثیر, ... رضا جبیره, محمد دن	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[بدون خروجی قرضی (بازگشت)]
12	نقل و انتقالات لیگ برتر	99-00	فولاد	مهران درخشان مهر, بهزاد ... نودانگه, حسین ابراهی	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	شهاب گردان, حسن بیت سجد, ... عباس بوعذار, امیرحس	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[بدون خروجی قرضی (بازگشت)]
13	نقل و انتقالات لیگ برتر	99-00	گل گهر سیرجان	آرمین سهرابیان, رضا شریفی, سجد ... صادقی, علی ا	[بدون ورودی احتمالی]	[بدون ورودی قرضی]	[بدون ورودی قرضی (بازگشت)]	سید احمد موسوی, عبدالله حسینی, ... مهرداد عبدی, م	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[بدون خروجی قرضی (بازگشت)]
14	نقل و انتقالات لیگ برتر	99-00	ماشین‌سازی‌تبریز	اکبر آقایی, ایمان یاصفا, بهمن ... سالاری, پیمان	[بدون ورودی احتمالی]	[ابوالفضل رزاق پور]	[بدون ورودی قرضی (بازگشت)]	شاهین توکلی, بابک مرادی, حامد ... ذکی, محمد خرم ا	[بدون خروجی احتمالی]	[بدون خروجی قرضی]	[بدون خروجی قرضی (بازگشت)]

شکل 36: قسمتی از نتیجه بدست آمده برای نقل و انتقالات

اکنون در ادامه نیز من اقدام به ذخیره‌سازی نتایج و داده‌های استخراج شده نقل و انتقالات با فرمت csv. می‌پردازم:

برای انجام این کار از دستور زیر استفاده می‌کنم:

```
79
80  df_jam_hazfi.to_csv("transfers_data.csv", index=False)
```

شکل 37: دستور استفاده شده برای ذخیره داده‌های استخراج شده از نقل و انتقالات

این فایل را می‌توانید که در پوشه Extracted Data موجود در فایل آپلود شده پروژه مشاهده نمایید.

استخراج داده‌های جام جهانی:

در این قسمت ما به بررسی و استخراج داده‌های مربوط به جام‌های جهانی می‌پردازیم.

ابتدا در این قسمت من اقدام به لود کردن داده‌هایی که در اختیار ما قرار داده‌اند می‌کنم و این کار را به صورت زیر انجام می‌دهم و نتایج آن نیز به صورت زیر هستند:

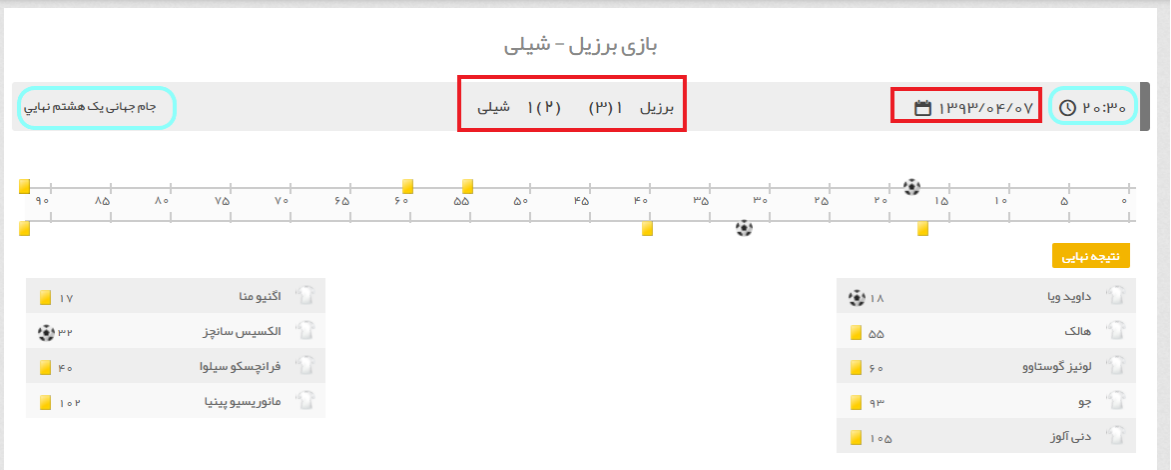
* World Cup:

```
[15]  1  world_cup = pd.read_csv('WorldCup.csv')
      2  display(world_cup)
```

	Unnamed: 0	Year	Stage	html
0	0	جام جهانی 2014	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
1	1	جام جهانی 2014	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
2	2	جام جهانی 2014	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
3	3	جام جهانی 2014	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
4	4	جام جهانی 2014	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
5	5	جام جهانی 2014	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
6	6	جام جهانی 2014	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
7	7	جام جهانی 2014	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
8	8	جام جهانی 2014	یک چهارم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
9	9	جام جهانی 2014	یک چهارم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
10	10	جام جهانی 2014	یک چهارم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
11	11	جام جهانی 2014	یک چهارم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
12	12	جام جهانی 2014	نیمه نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
13	13	جام جهانی 2014	نیمه نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
14	14	جام جهانی 2014	فینال	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
15	15	جام جهانی 2018	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
16	16	جام جهانی 2018	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
17	17	جام جهانی 2018	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...
18	18	جام جهانی 2018	یک هشتم نهایی	<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 T...

شکل 38: داده‌های داده شده برای جام‌جهانی

سپس اکنون در ادامه یک نمونه از صفحه یکی از موارد بالا را قرار می‌دهم که اطلاعات موجود در آن و آن‌هایی که باید استخراج شود را می‌توانید که مشاهده کنید:



شکل 39: صفحه مربوط به یکی از بازی‌های جام جهانی

همان‌طور که در شکل بالا نیز می‌توانید مشاهده کنید، موارد مهم که باید استخراج شوند مشخص شده‌اند.

اکنون در ادامه نیز من از دستورات زیر استفاده می‌کنم برای ساخت یک جدول به همراه ستون‌های مورد نیاز برای نمایش و استخراج داده‌ها و قرار دادن آن‌ها در این جدول:

```
index = np.array([np.arange(len(world_cup.index))]).flatten()

columns = ['Type', 'Year', 'Stage', 'Team Right', 'Team Left',
           'Score Right', 'Score Left', 'Time', 'Date', 'Penalty Right', 'Penalty Left']
df_world_cup = pd.DataFrame(index=index, columns=columns)

```

شکل 40: دستورات استفاده شده برای تشکیل جدول به همراه ستون‌های آن برای داده‌های جام جهانی

اکنون در ادامه نیز من کد استفاده شده برای بدست آوردن و استخراج داده‌ها را قرار می‌دهم:

```python
for i in range(len(world_cup.index)):
    soup_temp = BeautifulSoup(world_cup.html[i], 'html.parser')
    team_left = soup_temp.find_all(class_='team-name left')[0].get_text()
    team_right = soup_temp.find_all(class_='team-name right')[0].get_text()
    score_left = soup_temp.find_all(class_='team-score left')[0].get_text()
    score_right = soup_temp.find_all(class_='team-score right')[0].get_text()
    time = soup_temp.find_all(class_='match-time pull-right')[0].get_text()
    date = soup_temp.find_all(class_='match-date pull-right')[0].get_text()
    df_world_cup.at[i, 'Team Left'] = team_left
    df_world_cup.at[i, 'Team Right'] = team_right

    df_world_cup.at[i, 'Score Left'] = score_left[-1]
    df_world_cup.at[i, 'Score Right'] = score_right[0]

    df_world_cup.at[i, 'Stage'] = world_cup.Stage[i]
    df_world_cup.at[i, 'Year'] = world_cup.Year[i][10:14]
    df_world_cup.at[i, 'Type'] = world_cup.Year[i][0:9]
    df_world_cup.at[i, 'Time'] = time
    df_world_cup.at[i, 'Date'] = date

    start = score_left.find("(") + len("(")
    # print(start)
    end = score_left.find(")")
    penalty_left = score_left[start:end]
    if start is 0:
        df_world_cup.at[i, 'Penalty Left'] = 0
    else:
        df_world_cup.at[i, 'Penalty Left'] = penalty_left

    start = score_right.find("(") + len("(")
    end = score_right.find(")")
    penalty_right = score_right[start:end]
    if start is 0:
        df_world_cup.at[i, 'Penalty Right'] = 0
    else:
        df_world_cup.at[i, 'Penalty Right'] = penalty_right
display(df_world_cup)
df_world_cup.to_csv("world_cup_data.csv", index=False)
```

شکل 41: کد استفاده شده برای استحراج داده‌های جام جهانی

اکنون در ادامه پس از اجرای این کد نتایج به صورت زیر بدست می‌آید:

	Type	Year	Stage	Team Right	Team Left	Score Right	Score Left	Time	Date	Penalty Right	Penalty Left
0	جام جهانی	2014	یک هشتم نهایی	برزیل	شیلی	1	1	20:30	1393/04/07	3	2
1	جام جهانی	2014	یک هشتم نهایی	کلمبیا	اروگوئه	2	0	00:30	1393/04/08	0	0
2	جام جهانی	2014	یک هشتم نهایی	فرانسه	نیجریه	2	0	20:30	1393/04/09	0	0
3	جام جهانی	2014	یک هشتم نهایی	آلمان	الجزایر	2	1	00:30	1393/04/10	0	0
4	جام جهانی	2014	یک هشتم نهایی	هلند	مکزیک	2	1	20:30	1393/04/08	0	0
5	جام جهانی	2014	یک هشتم نهایی	کاستاریکا	یونان	1	1	00:30	1393/04/09	5	3
6	جام جهانی	2014	یک هشتم نهایی	آرژانتین	سوئیس	1	0	20:30	1393/04/10	0	0
7	جام جهانی	2014	یک هشتم نهایی	بلژیک	آمریکا	2	1	00:30	1393/04/11	0	0
8	جام جهانی	2014	یک چهارم نهایی	برزیل	کلمبیا	2	1	00:30	1393/04/14	0	0
9	جام جهانی	2014	یک چهارم نهایی	فرانسه	آلمان	0	1	20:30	1393/04/13	0	0
10	جام جهانی	2014	یک چهارم نهایی	هلند	کاستاریکا	0	0	00:30	1393/04/15	4	3
11	جام جهانی	2014	یک چهارم نهایی	آرژانتین	بلژیک	1	0	20:30	1393/04/14	0	0
12	جام جهانی	2014	نیمه نهایی	برزیل	آلمان	1	7	00:30	1393/04/18	0	0
13	جام جهانی	2014	نیمه نهایی	هلند	آرژانتین	0	0	00:30	1393/04/19	2	4
14	جام جهانی	2014	فینال	آلمان	آرژانتین	1	0	23:30	1393/04/22	0	0
15	جام جهانی	2018	یک هشتم نهایی	اروگوئه	پرتغال	2	1	22:30	1397/04/09	0	0
16	جام جهانی	2018	یک هشتم نهایی	فرانسه	آرژانتین	4	3	18:30	1397/04/09	0	0
17	جام جهانی	2018	یک هشتم نهایی	برزیل	مکزیک	2	0	18:30	1397/04/11	0	0
18	جام جهانی	2018	یک هشتم نهایی	بلژیک	ژاپن	3	2	22:30	1397/04/11	0	0
19	جام جهانی	2018	یک هشتم نهایی	اسپانیا	روسیه	1	1	18:30	1397/04/10	3	4
20	جام جهانی	2018	یک هشتم نهایی	کرواسی	دانمارک	1	1	22:30	1397/04/10	3	2
21	جام جهانی	2018	یک هشتم نهایی	سوئد	سوئیس	1	0	18:30	1397/04/12	0	0
22	جام جهانی	2018	یک هشتم نهایی	کلمبیا	انگلیس	1	1	22:30	1397/04/12	3	4

شکل 42: قسمتی از نتایج بدست آمده برای داده‌های جام جهانی

در ادامه نیز اکنون باید اقدام به ذخیره این داده‌ها و نتایج به فرمت csv. بکنیم.

برای ذخیره‌سازی داده‌ها به صورت زیر عمل می‌کنیم:

```
38
39  df_world_cup.to_csv("world_cup_data.csv", index=False)
```

شکل 43: دستور استفاده شده برای ذخیره‌سازی داده‌های استخراج شده

همچنین می‌توانید که این فایل بدست آمده را در پوشه Extracted Data که در فایل آپلود شده قرار دارد، مشاهده کنید.

سوالات گزارش کارگاه:

من در این قسمت سعی می‌کنم تا به سوالاتی که از ما خواسته شده است به آن پاسخ دهیم اشاره کنم و در مورد آن‌ها توضیحاتی را ارائه کنم.

❖ سوال 1:

در پاسخ به این سوال که از ما خواسته شده است که بگوییم که چه داده‌هایی را از داخل دیتاست‌هایی که در اختیار ما قرار داده شده است استخراج کرده‌ایم باید بگویم که تقریبا به این سوال در گزارش پیاده‌سازی کارگاه که در بخش‌های قبلی بخش به بخش به آن پرداختم اشاره کردم و در این جا دیگر مجدد به آن نمی‌پردازم.

❖ سوال 2:

در این سوال از ما خواسته شده است تا هزینه استخراج هر کدام از این ویژگی‌های که استخراج کرده‌ایم را بیان کنیم.

اینطور که من از TA گرامی سوال کردم گفتند که به این سوال باید از دیدگاه صنعتی و ... پاسخ بدهیم و برداشت من بیشتر هزینه‌های مادی و از این دست بود که باید به آن اشاره کنیم. به همین منظور من به صورت زیر به این سوال پاسخ می‌دهم:

برای انجام این کار استخراج داده‌ها روش‌های مختلفی وجود دارد که بسته به هر کدام از این روش‌ها هزینه انجام آن نیز متفاوت می‌باشد.

روش اول: برون سپاری پروژه

معمولا برون‌پاری پروژه در کارهای استخراج داده رایج هست. و مخصوصا در صورتی که پروژه‌ای که دارید سر راست باشد و پیچیدگی خاصی نداشته باشد و همچنین نیاز به توضیحات زیاد و شخصی‌سازی‌های زیادی نداشته باشد، این کار توجیه‌پذیرتر نیز هست.

اما خب در این روش باید این موضوع را مد نظر داشته باشیم که هزینه در این نوع پروژه‌ها براساس نرخ ساعتی محاسبه می‌شود. به عنوان مثال من با جستجویی که در نت کردم نرخ ساعتی متوسط برای این کارها در خارج از کشور از 30 تا 60 دلار در هر ساعت شروع می‌شود و تا 100 دلار نیز می‌تواند برسد.

و این موضوع و این هزینه‌ها برای پروژه‌هایی که طولانی مدت هستند و نیز نیاز به اسکیل کردن دارند می‌تواند خیلی بیشتر نیز در طولانی مدت افزایش پیدا کند و این خیلی خوب نیست.

روش دوم: ساخت استخراج کننده مخصوص به خود

روش‌های مختلفی برای انجام این کار هست می‌توانیم که با پایتون مانند کاری که در این پروژه انجام دادیم کدی را بنویسیم یا این که در اکسل دستوراتی را بنویسیم و بتوانیم که این داده‌ها را استخراج کنیم. و این موضوع بستگی به پروژه و کارکردی که نیاز داریم دارد.

درصورتی که خودمان قصد داریم که این ابزار را پیاده کنیم به چند نکته باید توجه کنیم:

اول: از چه پلتفرم یا زبان برنامه‌نویسی‌ای برای ساخت استخراج کننده خود استفاده می‌کنید؟

دوم: خودت پیاده‌سازی می‌کنی یا این که به کسی می‌دهی برایت پیاده‌سازی کند؟

سوم: هزینه برون‌سپاری پروژه استخراج داده‌ات چقدر است؟

چهارم: این برنامه استخراج کننده‌ات قرار است که بر روی تنها یک وبسایت اجرا شود یا این که قرار است بر روی چندین وبسایت مختلف اجرا شود؟

پنجم: موضوع مهم دیگر زمانبندی پروژه هست! آیا شما زمان کافی برای ساخت یک استخراج کننده و رفع تمامی باگ‌های آن دارید یا خیر؟

همان‌طور که می‌توانید مشاهده کنید ساخت یک استخراج‌کننده داده می‌تواند که پروژه بزرگی برای انجام دادن باشد. این موضوع تماما به نیازهای پروژه شما و همین‌طور منابع موجود در شرکت شما بستگی دارد.

در اکثر مواقع ما باید که به دنبال راه‌حل سریع‌تر، ارزان‌تر و نیز راحت‌تر باشیم.

روش سوم: استفاده از یکی از استخراج‌کننده‌های موجود

در بسیاری از حالت‌ها، استفاده از استخراج‌کننده‌های موجود یکی از بهترین راه‌حل‌ها است برای رفع نیازهای ما!

بسیاری از استخراج‌کننده‌ها در طول چندین سال توسعه داده شده‌اند و بهبود پیدا کرده‌اند و آن‌ها قادر هستند که دیتاها را از انواع مختلفی از وبسایت‌ها استخراج کنند. همچنین خوبی دیگری که این استخراج کننده‌ها دارند این است که این‌ها باگ‌ها و ارورهای آن‌ها بسیار کم است.

و در این حالت وقتی که بحث هزینه نیز می‌شود بسته به استخراج‌کننده‌ای که انتخاب می‌کنید و نیازهای پروژه شما متفاوت است. بسیاری از استخراج کننده‌های آماده پلن‌ها و طرح‌های رایگان نیز در کنار طرح‌های پولی دارند و طرح‌های پولی آن‌ها نیز عموما flat هست برای پروژه‌های ما و این ما را از هزینه‌های گران‌قیمت ساعتی رها می‌کند.

در آخر نیز خوب است به این موضوع اشاره کنم که بهترین استخراج کننده بسته به پروژه شما می‌تواند که متفاوت باشد و هزینه آن برای شما پایین‌تر یا بالاتر باشد.

همچنین برای بدست آوردن هزینه استخراج ویژگی‌ها چون ما علاوه بر موارد بالا هزینه‌های تمیزکردن و data cleaning را نیز داشتیم و باید آن را نیز به آن اضافه کنیم.

بر طبق تحقیقی که کردم سه نوع هزینه برای تمیزسازی دیتا تعریف می‌شود که عبارت‌اند از:

هزینه داده‌های تکراری – هزینه داده‌های گم‌شده - هزینه داده‌های جعلی یا اشتباه

ما در این پروژه داده‌های تکراری و داده‌های اشتباه و جعلی نداشتیم و تنها Missing داشتیم. که همان Nan های ما بود!

و باابررسی من برای هر 10000تا داده Missing چیزی حدود 1000 تا 3000 دلار هزینه آن می‌شود!

اما خب ما در همه جدول‌ها داده Missing نداشتیم! در ادامه من هزینه هر کدام را به صورت تقریبی می‌آورم:

هزینه‌های استخراج را به صورت ساعتی محاسبه کردم و در ادامه به صورت زیر اشاره می‌کنم:

جدول جام حذفی و جام جهانی هر کدام 2 ساعت و هر ساعت من 50 دلار در نظر گرفتم! و در این داده‌ها ما داده گم شده‌ای نداشتیم پس هر جدول 100 دلار و در مجموع 200 دلار برای این دو جدول هزینه استخراج آن‌ها می‌شود.

```
Type             0
Year             0
Stage            0
Team Right       0
Team Left        0
Score Right      0
Score Left       0
Time             0
Date             0
Penalty Right    0
Penalty Left     0
dtype: int64
```

شکل 44: تعداد Missing Valueها در جدول جام حذفی

```
Type             0
Year             0
Stage            0
Team Right       0
Team Left        0
Score Right      0
Score Left       0
Time             0
Date             0
Penalty Right    0
Penalty Left     0
dtype: int64
```

شکل 45: تعداد Missing Valueها در جدول جام جهانی

اکنون به سراغ جدول تیم‌ها می‌رویم:

در این جدول زمانی که برای استخراج از من گرفت حدود 5 ساعت بود و با حساب کردن ساعتی 50 دلار می‌شود 250 دلار هزینه استخراج!

همچنین این جدول 20 عدد داده Missing داشت که با حساب کردن با آن فرمول داده شده و اگر بگیریم 1000 دلار به ازای 10000 داده می‌شود:

```
Competition     0
Team            0
Playes         10
Scores         10
dtype: int64
```

شکل 46: داده‌های Missing جدول تیم‌ها

پس با توجه به این موضوع حدود 2 دلار نیز هزینه رفع این داده‌های Missing می‌شود.

اکنون به سراغ جدول نقل و انتقالات می‌رویم:

در این جدول زمانی حدود 8 ساعت را از من گرفت که با حساب کردن ساعتی 50 دلار می‌شود 400 دلار در مجموع برای استخراج داده!

همچنین این جدول داده‌های Missing هم داشت که طبق شکل زیر حدود 1762 داده هست! و با محاسبه طبق فرمول می‌شود حدود 176 دلار!

```
Transfer                          0
Year                              0
Team                              0
('ورودی' , 'قطعی')               36
('ورودی' , 'احتمالی')           314
('ورودی' , 'قرضی')              261
('ورودی' , 'قرضی (بازگشت)')     287
('خروجی' , 'قطعی')               34
('خروجی' , 'احتمالی')           308
('خروجی' , 'قرضی')               231
('خروجی' , 'قرضی (بازگشت)')      291
dtype: int64
```

شکل 47: تعداد Missingها در هر ستون جدول نقل و انتقالات

اکنون به سراغ جدول‌ها می‌رویم!

جدول فوتبال و بسکتبال داده Missing نداشتند! و ساخت هر کدام از این جدول‌ها چون مشابه بودند ساخت یکی حدود 5 ساعت و تعمیم آن‌ها به بقیه 1 ساعتی زمان برد پس در مجموع حدود 6 ساعتی وقت گرفت! پس می‌شود 300 دلار برای ساخت این جدول‌ها!

حال به سراغ Missingها می‌رویم:

```
Competition         0
Year                0
Team                0
Played              0
Won                 0
Lost                0
Goals For           0
Goals Agains        0
Goal difference     0
Points              0
dtype: int64
```

شکل 48: داده‌های Missing جدول بسکتبال

```
Competition         0
Year                0
Team                0
Played              0
Won                 0
Drawn               0
Lost                0
Goals For           0
Goals Agains        0
Goal difference     0
Points              0
dtype: int64
```

شکل 49: داده‌های Missing جدول فوتبال

```
first          second
Competition              0
Team           Team      0
Point          Point     0
Matchs         Played    0
               Won       0
               Lost      0
Resul Details  30       42
               31       38
               32       42
               23       43
               13       32
               03       41
Set            Won       0
               Lost      0
               Avg       0
Points         Won       0
               Lost      0
               Avg       0
dtype: int64
```

شکل 50: داده‌های Missing جدول والیبال

تنها جدولی که داده Missing دارد جدول والیبال است که حدود 238 داده Miss شده دارد و هزینه آن‌ها چیزی حدود 23 دلار می‌شود!

❖ سوال 3:

در این سوال از ما پرسیده شده است که این هزینه‌هایی که در مرحله قبل محاسبه کردیم چه توجیهی دارد!

درست است که این هزینه‌ها حتی برای چنین پروژه پیش‌پا افتاده و کم اهمیتی نیز زیاد است اما فرض کنید که در یک شرکت باشیم که قرار است بر مبنای این داده‌ها تصمیم‌گیری شود و این داده‌ها مهم باشند حال در این حالت می‌بینیم که مثلا اگر داده‌ها تمیز نشده باشند و قابل اطمینان به اندازه کافی نباشند می‌توانند که منجر به تصمیم‌گیری‌های اشتباه برای ما بشوند که هزینه آن مخصوصا در دراز مدت می‌تواند که بسیار بیشتر باشد.

همچنین کلا انجام کار استخراج داده نیز اگر نیاز یک شرکت باشد این داده‌ها در دنیای امروزه حیاتی هستند! به عنوان مثال یک فروشگاه خرده فروشی می‌تواند با استخراج داده‌های محصولات فروشگاه‌های

رقیب خود از موجودی اجناس و قیمت‌های آن فروشگاه‌ها مطلع شود و بررسی کند که آیا قیمت‌ها و کالاهایش آیا توانایی رقابت با آن فروشگاه‌ها را همچنان دارند یا در حال از دست دادن مزیت‌های رقابتی خود مانند قیمت کمتر یا موجودی کالاهای بیشتر هست! و با انجام این کار می‌تواند که قبل از این که فاجعه روی دهد و مشتریان خود را از دست بدهد استراتژی مناسبی را مدیران شرکت بچینند و موجودی و قیمت‌ها را مجدد رقابتی کنند!

اگر این کار انجام نشود و فروشگاه تنها به داده‌های فروشگاه خودش متکی باشد! چندین ماه و شاید بیشتر طول می‌کشد تا داده‌ها فروشگاه کاهش فروش را نشان دهند و نشان دهند که مشتری‌ها در حال از دست رفتن هستند! و تا آن زمان فروشگاه تا بخواهد که اقدام به اصلاح خود کند سرمایه زیادی را از دست می‌دهد و مهمتر از همه اعتماد مشتریانش را از دست می‌دهد و بازگرداندن این اعتماد کار بسیار سختی هست و هزینه آن به مراتب کمتر از این هزینه‌هایی هست که من به آن اشاره کردم!

❖ سوال 4:

در این سوال از ما پرسیده شده است که برای این که بفهمیم داده‌های استخراج شده ما قابل اعتماد هستند، چه کاری می‌توانیم که انجام دهیم؟

در این قسمت روش‌های مختلفی برای انجام این کار وجود دارد! در این مثال ما حتی با مواردی می‌توانیم با نگاه کردن و مقایسه دیتای تولید شده با سایت نیز متوجه بشویم که آیا کارمان را درست انجام داده‌ایم یا خیر چون دیتا ما کم است و همین‌طور ثابت است و به روز نمی‌شود این روش جواب می‌دهد. اما اگر بخواهم پاسخ کلی‌تری بدهم:

یک راه خوب که دکتر شیرازی در کارگاه نیز به آن اشاره کردند این است که بیاییم و توزیع داده‌ها یا هیستوگرام آن‌ها را رسم کنیم و این کار مخصوصا در مورد این دیتاستی که ما در اختیار داریم خوب جواب می‌دهد و ببینیم که توزیع داده‌ها و مقدارهای آن‌ها در چه محدوده‌ای هست! با انجام این کار می‌توانیم که به سادگی Outlierها و داده‌هایی که غیر معمول هستند را شناسایی کنیم!

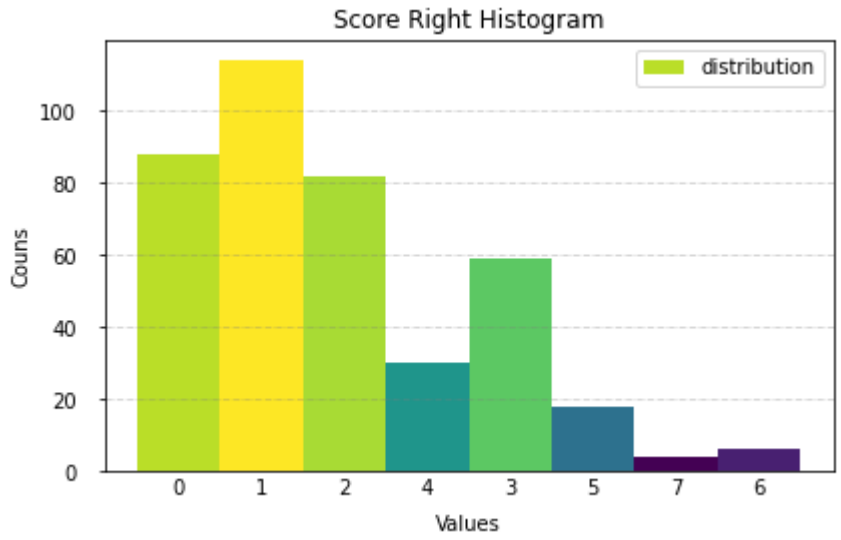
به عنوان مثال در مورد داده‌های جام حذفی و ... می‌توانیم که با رسم توزیع و هیستوگرام گل‌های زده شده توسط هر تیم بررسی کنیم که تعداد این گل‌ها در چه محدوده‌ای هست! و ما می‌بینیم که مثلا گل‌ها همگی بین 0 تا حداکثر 10 گل هست! حال اگر یک تیم در یک بازی در این نمودار ما باشد که 100 گل

زده باشد در نتیجه ما متوجه می‌شویم که مشکلی روی داده است و این داده ما احتمالا با یک مشکلی روبه رو هست! من در ادامه این کار را برای تعدادی از جدول‌های موجود انجام می‌دهم:

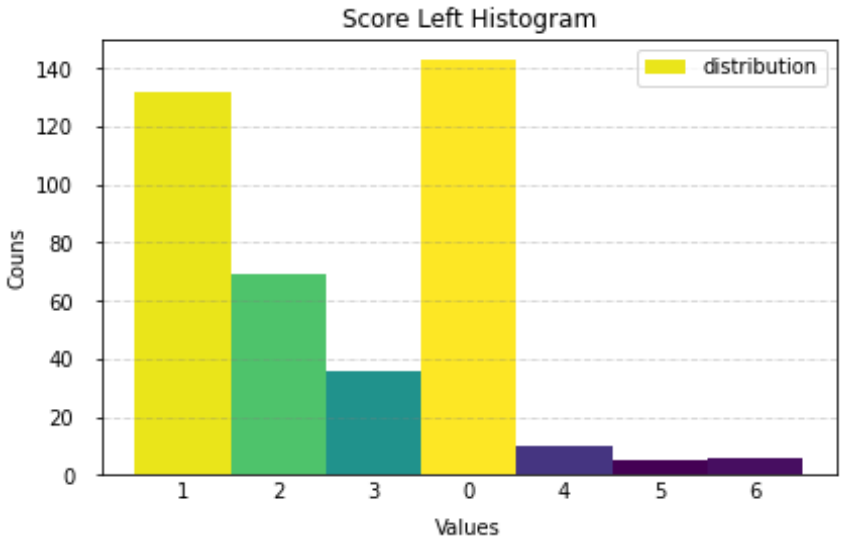
ابتدا من کدی که به این منظور استفاده کرده‌ام را قرار می‌دهم:

```
def plot_hist(data, name):
    legend = ['distribution']
    fig, axs = plt.subplots(1, 1,
                            # figsize =(5, 5),
                            tight_layout = True)

    # Add padding between axes and labels
    axs.xaxis.set_tick_params(pad=5)
    axs.yaxis.set_tick_params(pad=10)
    # Add x, y gridlines
    axs.grid(b=True, color='grey', linestyle='-.', linewidth=0.5, alpha=0.6)

    n_bins = (len(data.value_counts()))
    print(data.value_counts())#.index)
    N, bins, patches = axs.hist(data, bins=int(n_bins), label='')
    plt.draw()
    ticks = [(t.get_text()) for t in axs.get_xticklabels()]
    # Setting color
    fracs = ((N**(1 / 5)) / N.max())
    norm = colors.Normalize(fracs.min(), fracs.max())

    for thisfrac, thispatch in zip(fracs, patches):
        color = plt.cm.viridis(norm(thisfrac))
        thispatch.set_facecolor(color)

    # Label the raw counts and the percentages below the x-axis...
    bin_centers = 0.5 * np.diff(bins) + bins[:-1]
    for count, x in zip(ticks, bin_centers):
        # Label the raw counts
        axs.annotate(int(count), xy=(x, 0), xycoords=('data', 'axes fraction'),
            xytext=(0, -3), textcoords='offset points', va='top', ha='center')

    # Adding extra features
    plt.xlabel("Values")
    # print(ticks)
    plt.xticks(ticks='-', label='')
    axs.xaxis.set_label_coords(0.5, -0.1)
    plt.ylabel("Couns")
    plt.legend(legend)
    plt.title(name+ ' Histogram')

    # Show plot
    plt.show()
```

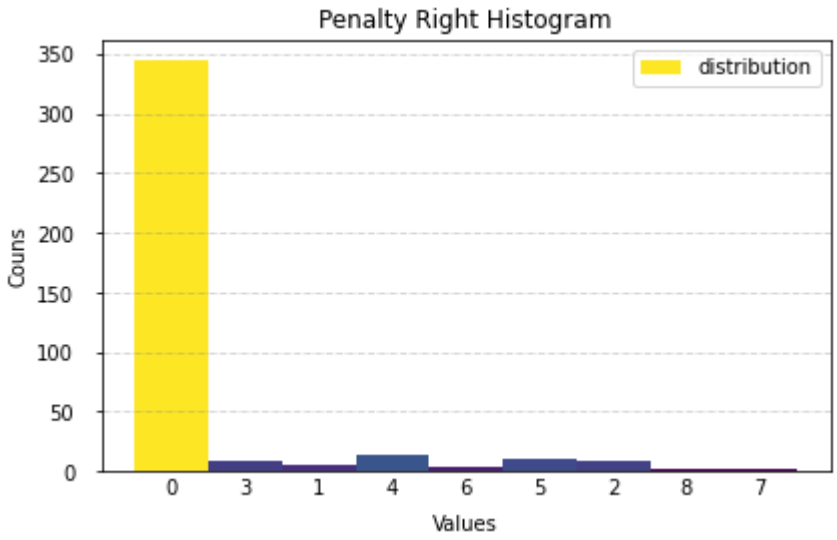
اکنون در ادامه نمودارهای بدست آمده را قرار می‌دهم:



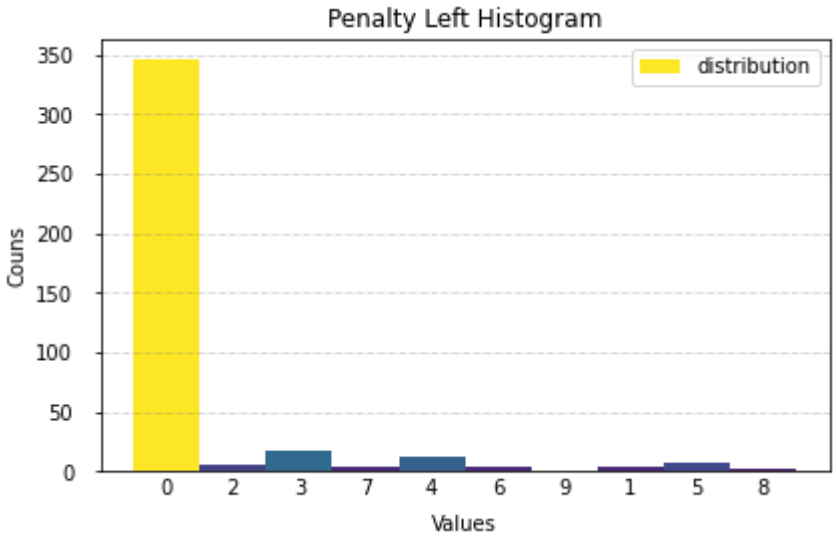
شکل 51: هیستوگرام گل‌های یک طرف تیم‌ها در جدول جام حذفی



شکل 52: هیستوگرام گل‌های طرف دیگر تیم‌ها در جدول جام حذفی



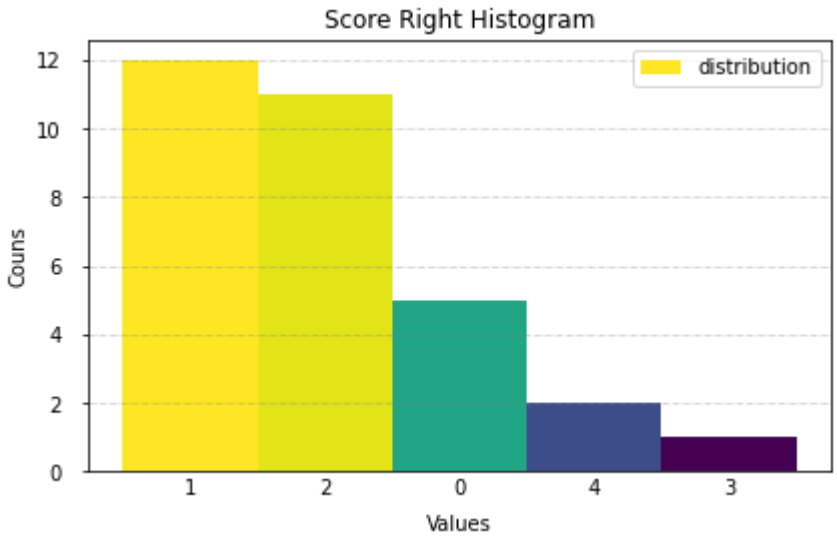
شکل 53:هیستوگرام تعداد پنالتی‌های یک طرف در جدول جام حذفی



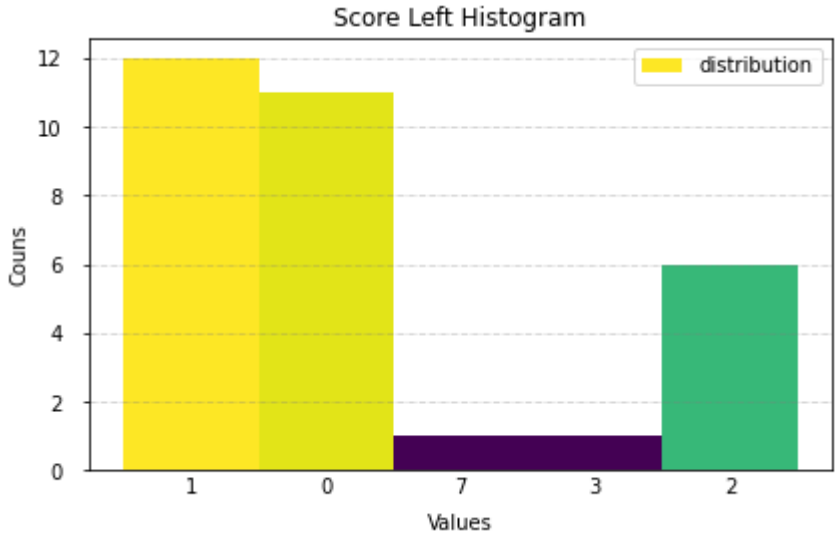
شکل 54:هیستوگرام تعداد پنالتی‌های طرف دیگر در جدول جام حذفی

همان‌طور که می‌توانید مشاهده کنید تمامی مقادیر در بازه حدود زیر 10 هستند و مقدار خارج از این بازه که outlier باشد را نداریم.

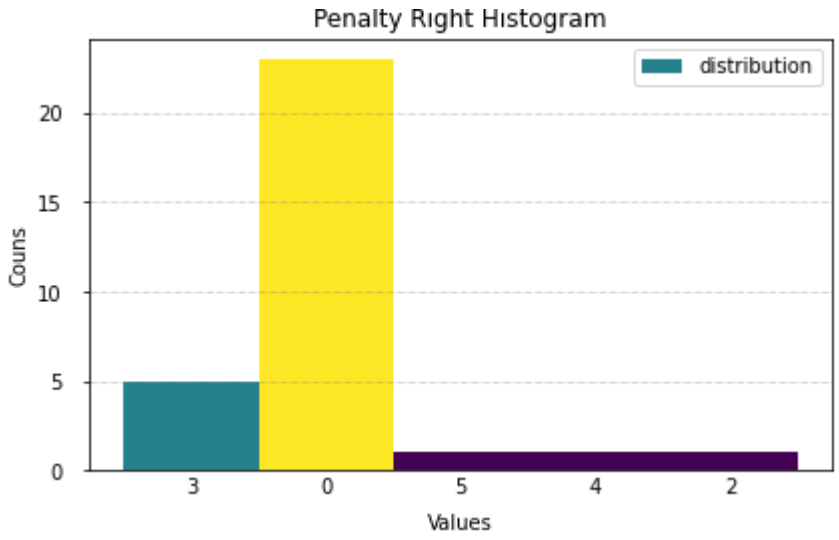
حال برای جدول جام جهانی نیز این کار را انجام می‌دهیم:



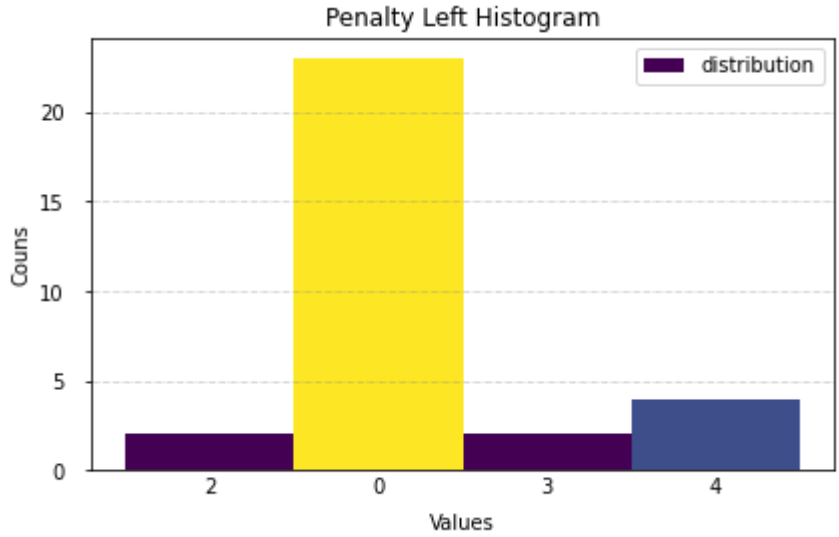
شکل 55: تعداد گل‌های زده شده تیم‌های یک طرف در جدول جام جهانی



شکل 56: تعداد گل‌های زده شده طرف دیگر در جدول جام جهانی



شکل 57:تعداد پنالتی‌های زده شده تیم یک طرف در جدول جام جهانی



شکل 58: تعداد پنالتی‌های زده شده طرف دیگر در جدول جام جهانی

همان‌طور که می‌بینیم در اینجا نیز همه داده‌ها در یک محدوده خاصی هستند و outlier ای را نمی‌توانیم که مشاهده کنیم.

در جدول نقل و انتقالات نیز به عنوان مثال می‌توانیم که طول لیست هر کدام از حالت‌ها را در نظر بگیریم که مثلا یک تیم در یک فصل معمولا در محدود 1 تا 10 یا 20 تا نقل و انتقال دارد و حال اگر یک تیمی بیاید و 200 تا نقل و انتقال داشته باشد معلوم می‌شود که مشکلی در کار هست.

در ادامه ابتدا من کد تغییر یافته برای اعمال این موضوع را قرار داده‌ام:

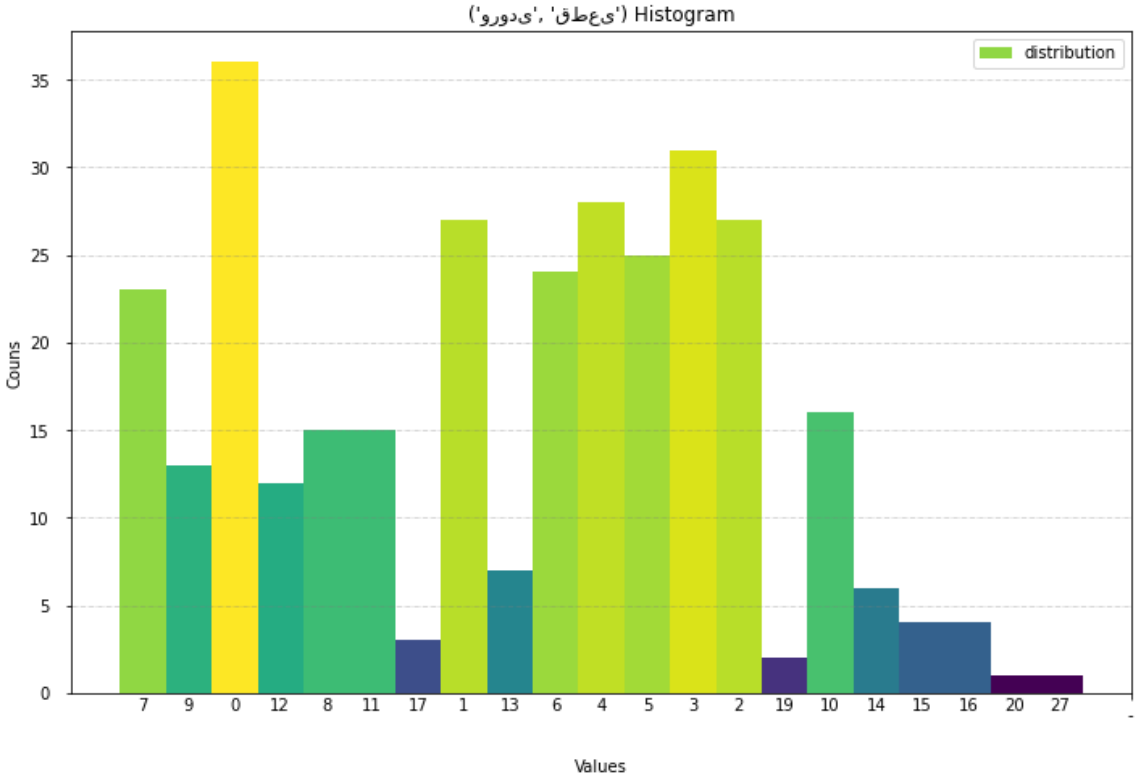
```
index = np.array([np.arange(len(transfers.index))]).flatten()

columns = ['Transfer', 'Year', 'Team', "('ورودی', 'قطعی')",
           "('ورودی', 'احتمالی')", "('ورودی', 'قرضی')", "('ورودی', 'قرضی (بازگشت)')",
           "('خروجی', 'قطعی')", "('خروجی', 'احتمالی')", "('خروجی', 'قرضی')", "('خروجی', 'قرضی (بازگشت)')"]
df_transfers_test = pd.DataFrame(index=index, columns=columns)
# df_transfers
iter_main = 0
for i in range(len(transfers.index)):
    soup_temp = BeautifulSoup(transfers.html[i], 'html.parser')
    team_list = soup_temp.find_all(class_='m3g-ct-col ct-team-name')
    team_in = soup_temp.find_all('div', {'class':'m3g-ct-col ct-in'})
    team_out = soup_temp.find_all('div', {'class':'m3g-ct-col ct-out'})

    for k in range(1, len(team_list)):
        team_name = team_list[k].get_text().strip()
        permanent_in = []
        permanent_out = []
        loan_return_in = []
        loan_return_out = []
        likely_in = []
        likely_out = []
        loan_in = []
        loan_out = []
        all_in_perm = team_in[k].find_all('span', {'class':['ct-player-name ct-permanent ', 'ct-player-name ct-permanent has-cat']})
        all_in_loan_return = team_in[k].find_all('span', {'class':['ct-player-name ct-loan-b ', 'ct-player-name ct-loan has-cat ct-loan-b']})
        all_in_likely = team_in[k].find_all('span', {'class':['ct-player-name ct-permanent likely has-cat', 'ct-player-name ct-permanent likely ']})
        all_in_loan = team_in[k].find_all('span', {'class':['ct-player-name ct-loan has-cat', 'ct-player-name ct-loan ']})

        all_out_perm = team_out[k].find_all('span', {'class':['ct-player-name ct-permanent ', 'ct-player-name ct-permanent has-cat']})
        all_out_loan_return = team_out[k].find_all('span', {'class':['ct-player-name ct-loan-b ', 'ct-player-name ct-loan has-cat ct-loan-b']})
        all_out_likely = team_out[k].find_all('span', {'class':['ct-player-name ct-permanent likely has-cat', 'ct-player-name ct-permanent likely ']})
        all_out_loan = team_out[k].find_all('span', {'class':['ct-player-name ct-loan has-cat', 'ct-player-name ct-loan ']})
        items = [all_in_perm, all_in_loan_return, all_in_likely, all_in_loan, all_out_perm, all_out_loan_return, all_out_likely, all_out_loan]
        for iteration, item in enumerate(items):
            for p in range(len(item)):
                if iteration == 0:
                    permanent_in.append(item[p].get_text().strip())
                elif iteration == 1:
                    loan_return_in.append(item[p].get_text().strip())
                elif iteration == 2:
                    likely_in.append(item[p].get_text().strip())
                elif iteration == 3:
                    loan_in.append(item[p].get_text().strip())
                elif iteration == 4:
                    permanent_out.append(item[p].get_text().strip())
                elif iteration == 5:
                    loan_return_out.append(item[p].get_text().strip())
                elif iteration == 6:
                    likely_out.append(item[p].get_text().strip())
                elif iteration == 7:
                    loan_out.append(item[p].get_text().strip())
        df_transfers_test.at[iter_main, 'Team'] = team_name
        year = [s for s in transfers.transfers[i].split() if s[0].isdigit()]
```

```
        transfer_name = transfers.transfers[i].replace(year[0], '')
        df_transfers_test.at[iter_main, 'Transfer'] = transfer_name.strip()
        df_transfers_test.at[iter_main, 'Year'] = year[0].strip()


        df_transfers_test.at[iter_main, "('ورودی','قطعی')"] = str(len(permanent_in))

        df_transfers_test.at[iter_main, "('ورودی','احتمالی')"] = str(len(likely_in))
        df_transfers_test.at[iter_main, "('ورودی','قرضی')"] = str(len(loan_in))
        df_transfers_test.at[iter_main, "('ورودی','قرضی (بازگشت)')"] = str(len(loan_re
turn_in))
        df_transfers_test.at[iter_main, "('خروجی','قطعی')"] = str(len(permanent_out
))
        df_transfers_test.at[iter_main, "('خروجی','احتمالی')"] = str(len(likely_out))

        df_transfers_test.at[iter_main, "('خروجی','قرضی')"] = str(len(loan_out))
        df_transfers_test.at[iter_main, "('خروجی','قرضی (بازگشت)')"] = str(len(loan_r
eturn_out))
        iter_main +=1
df_transfers_test.reset_index(inplace=True, drop=True)
display(df_transfers_test)
print(df_transfers_test.isna().sum())
```

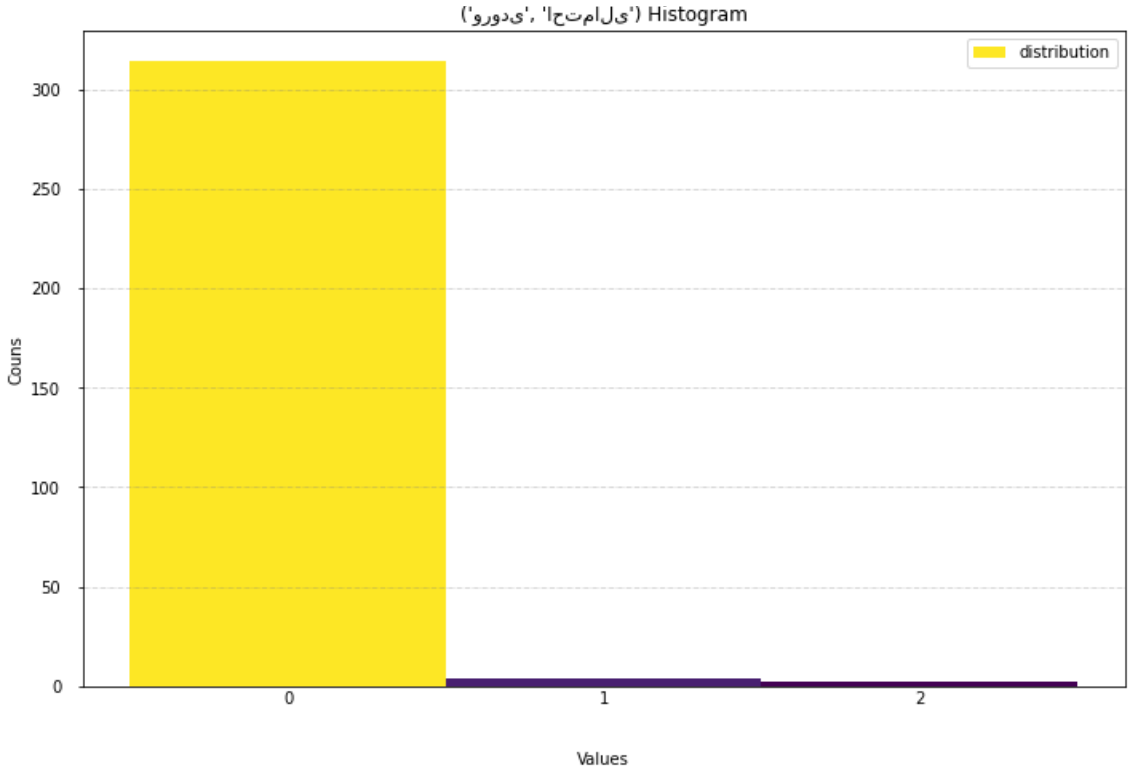
سپس اقدام به اجرا و به دست آوردن نمودارهای هیستوگرام کردم به صورت زیر:

```
plot_hist(df_transfers_test["('قطعی' ,'ورودی')"], "('قطعی' ,'ورودی')")
plot_hist(df_transfers_test["('احتمالی' ,'ورودی')"], "('احتمالی' ,'ورودی')")
plot_hist(df_transfers_test["('قرضی' ,'ورودی')"], "('قرضی' ,'ورودی')")
plot_hist(df_transfers_test["('قرضی (بازگشت)' ,'ورودی')"], "('قرضی (بازگشت)' ,'ورودی')")
plot_hist(df_transfers_test["('قطعی' ,'خروجی')"], "('قطعی' ,'خروجی')")
plot_hist(df_transfers_test["('احتمالی' ,'خروجی')"], "('احتمالی' ,'خروجی')")
plot_hist(df_transfers_test["('قرضی' ,'خروجی')"], "('قرضی' ,'خروجی')")
plot_hist(df_transfers_test["('قرضی (بازگشت)' ,'خروجی')"], "('قرضی (بازگشت)' ,'خروجی')")
```

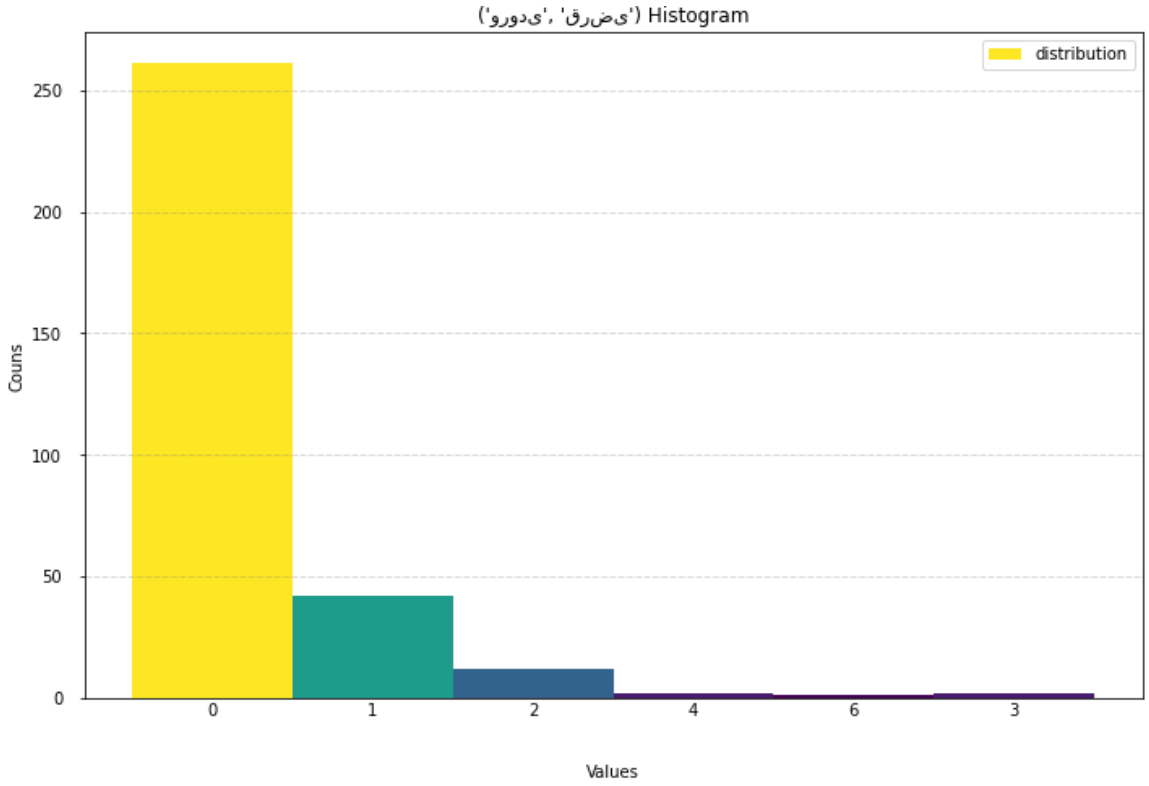
شکل 59


('ورودی', 'قطعی') Histogram
distribution
Couns
Values
0
5
10
15
20
25
30
35
7 9 0 12 8 11 17 1 13 6 4 5 3 2 19 10 14 15 16 20 27

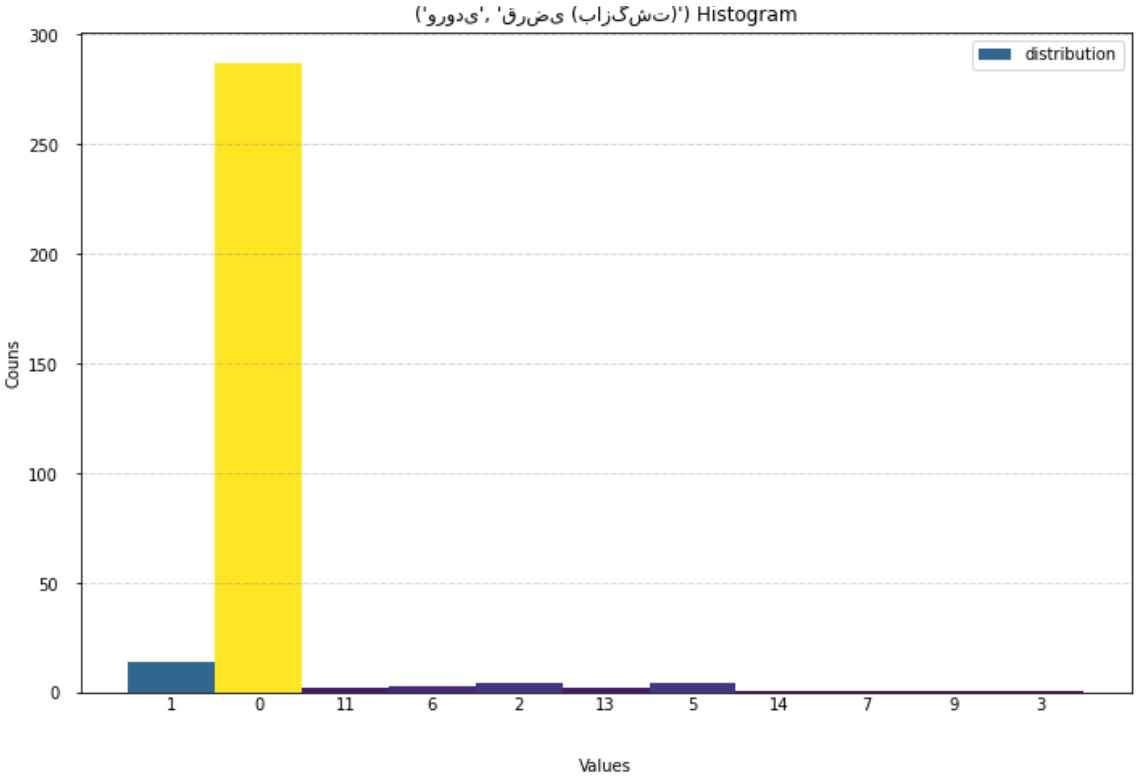
شکل 60


('ورودی', 'احتمالی') Histogram
distribution
Couns
Values
0
50
100
150
200
250
300
0 1 2

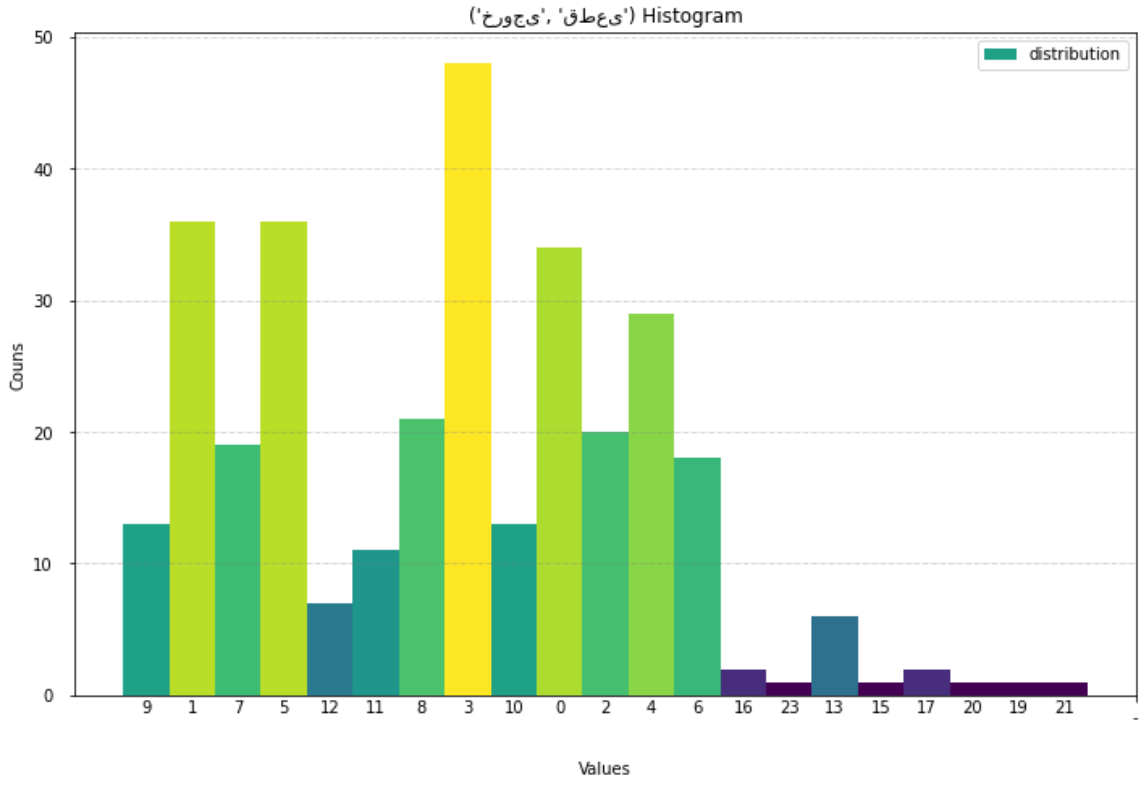
شکل 61


('ورودی', 'قرضی') Histogram
distribution
250
200
150
100
50
0
Couns
0
1
2
4
6
3
Values

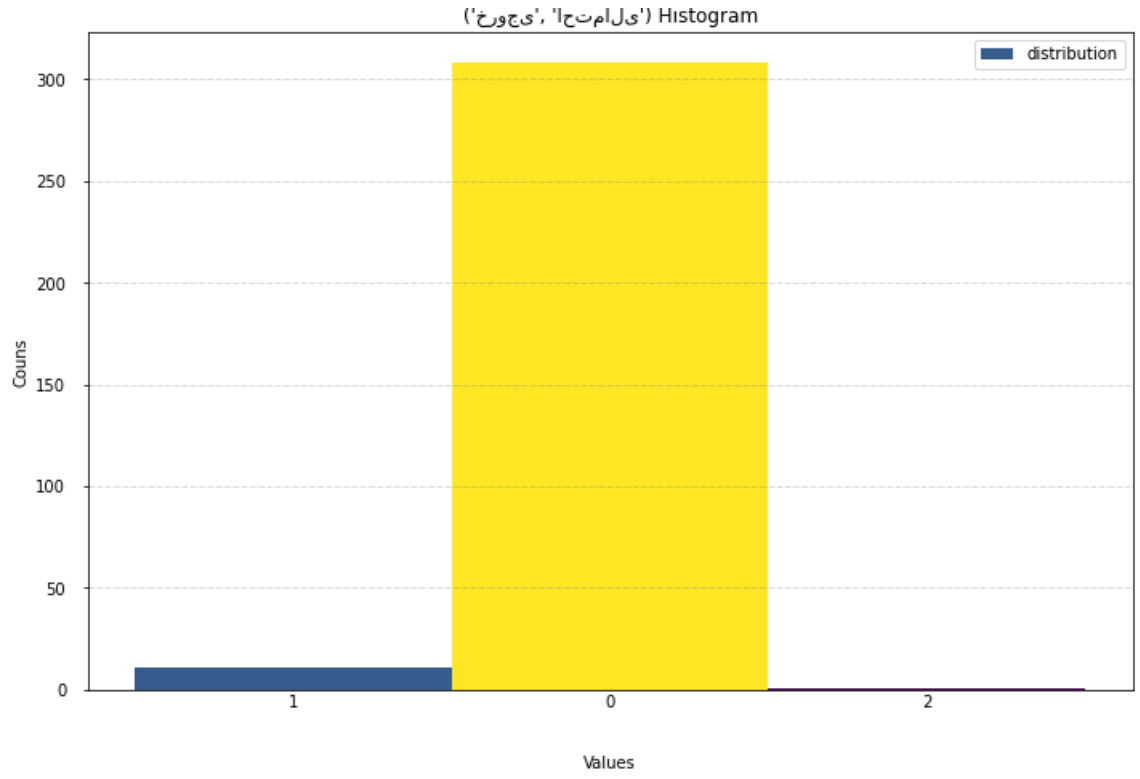
شکل 62


('ورودی', 'قرضی (بازگشت)') Histogram
distribution
300
250
200
150
100
50
0
Couns
1
0
11
6
2
13
5
14
7
9
3
Values

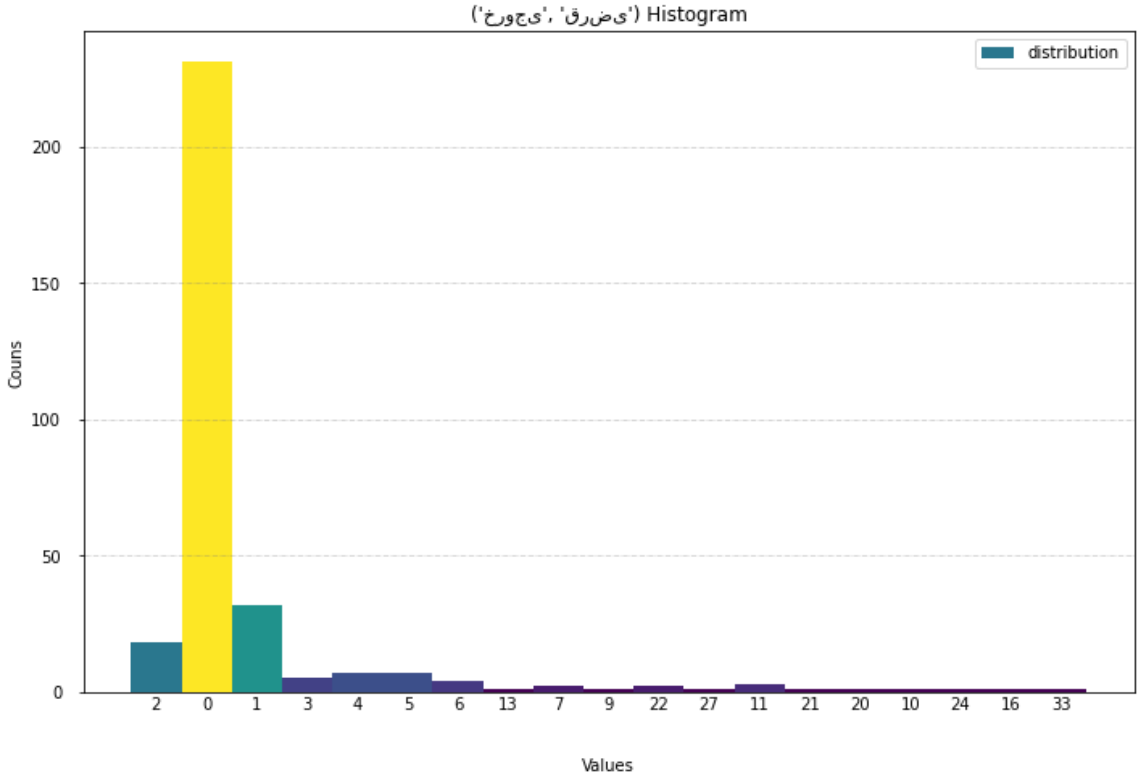
شکل 63


('خوجی', 'قطعی') Histogram
distribution
Couns
Values
0
10
20
30
40
50
9
1
7
5
12
11
8
3
10
0
2
4
6
16
23
13
15
17
20
19
21

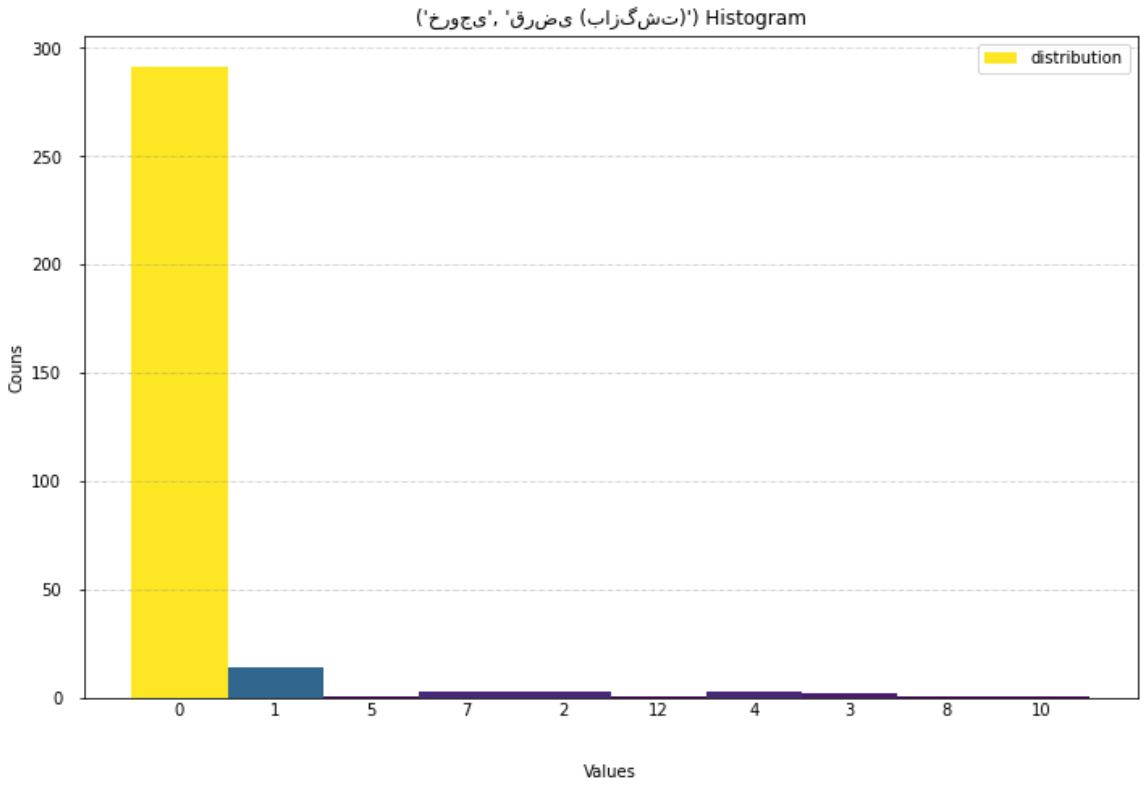
شکل 64


('خوجی', 'احتمالی') Histogram
distribution
Couns
Values
0
50
100
150
200
250
300
1
0
2

شکل 65


('خروجی', 'قرضی') Histogram
distribution
Couns
Values
0
50
100
150
200
2
0
1
3
4
5
6
13
7
9
22
27
11
21
20
10
24
16
33

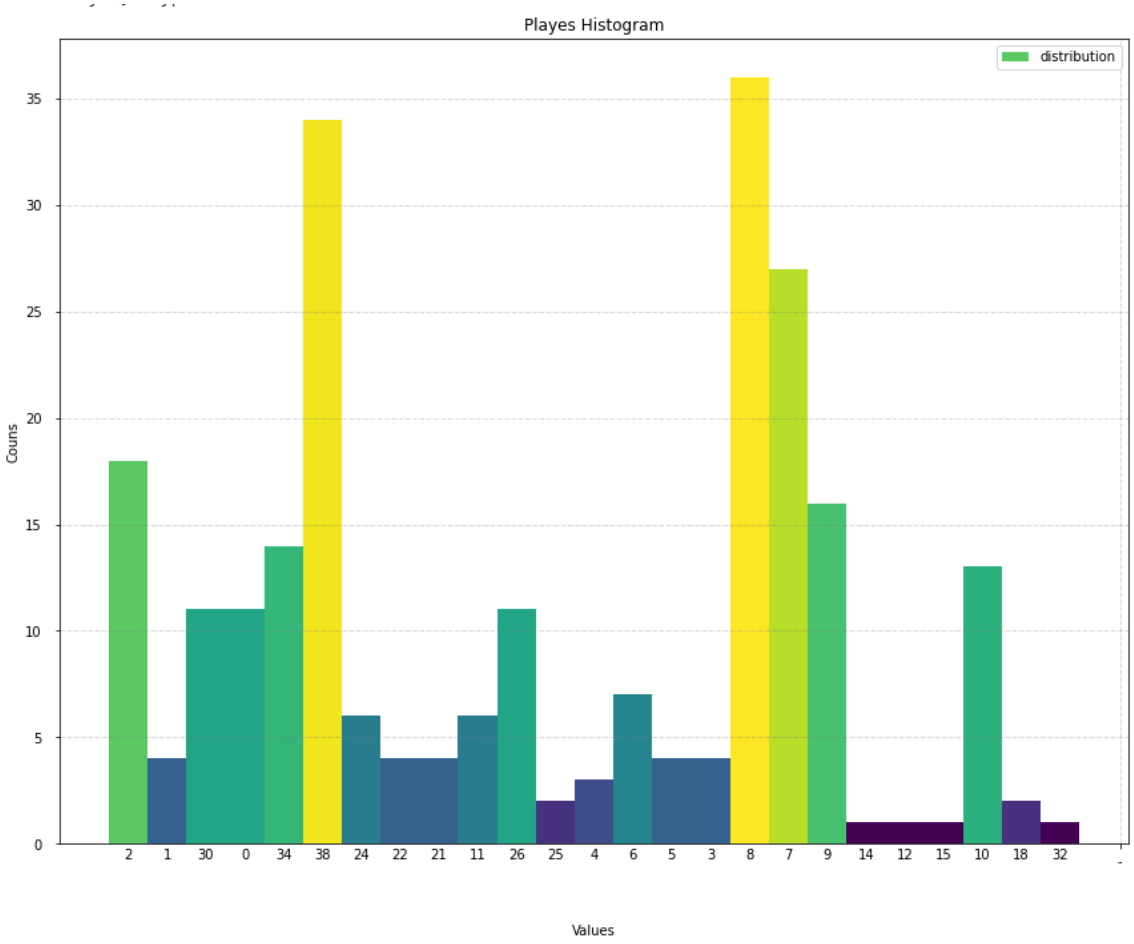
شکل 66


('خروجی', 'قرضی (بازگشت)') Histogram
distribution
Couns
Values
0
50
100
150
200
250
300
0
1
5
7
2
12
4
3
8
10

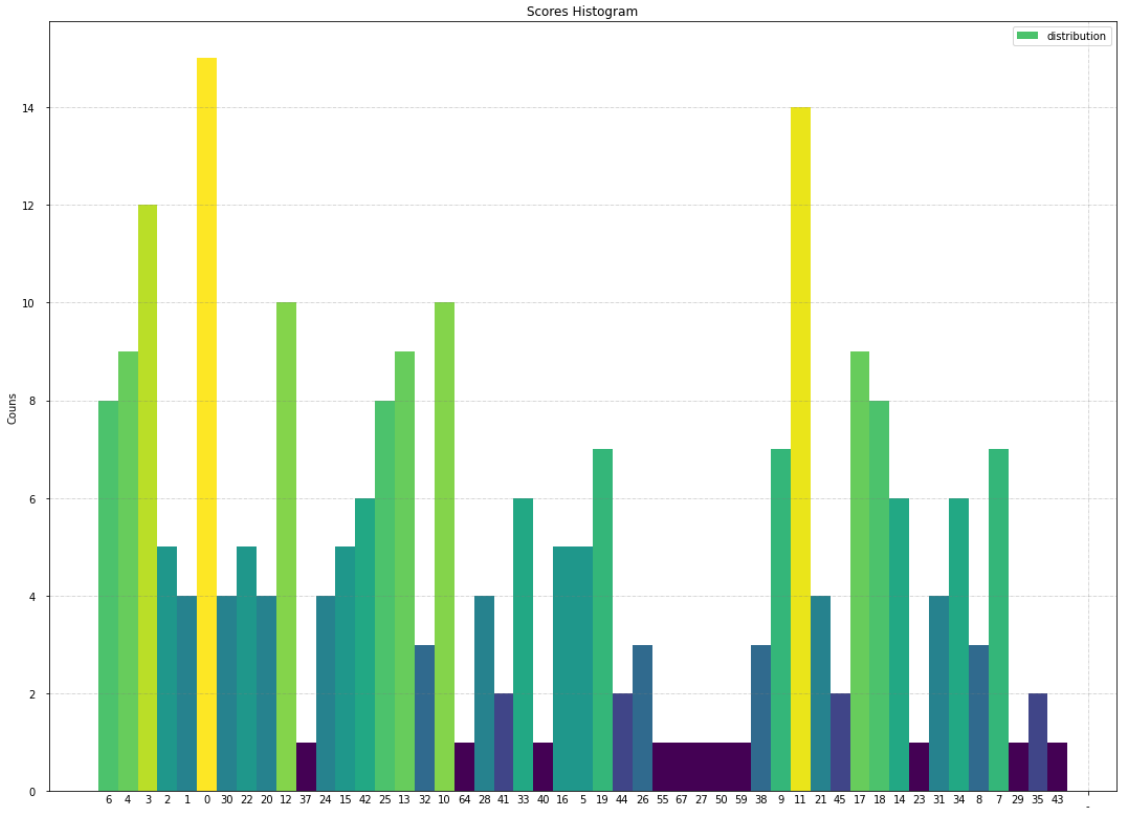
شکل 67

همان‌طور که در شکل‌های بالا نیز می‌توانید که مشاهده کنید این جدول نقل و انتقالات نیز تقریبا تمامی حالات آن طول لیست‌ها که به آن اشاره کردم در یک محدوده مشخصی است و موردی تحت عنوان outlier را نمی‌توانیم که در آن شناسایی کنیم به همین دلیل نتیجه می‌گیریم که این جدول نیز پیاده‌سازی خوبی داشته است.

حال جدول تیم را نیز برایش Histogramهایش را رسم می‌کنم که می‌توانید در ادامه آن‌ها را مشاهده نمایید:



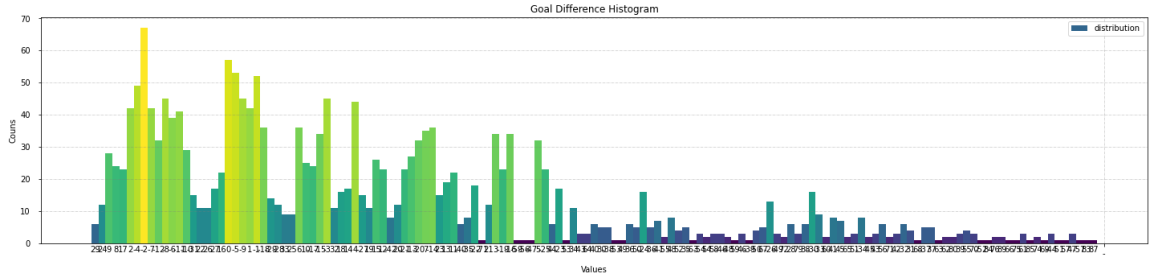
شکل 68: هیستوگرام بازی‌های انجام شده تیم‌ها



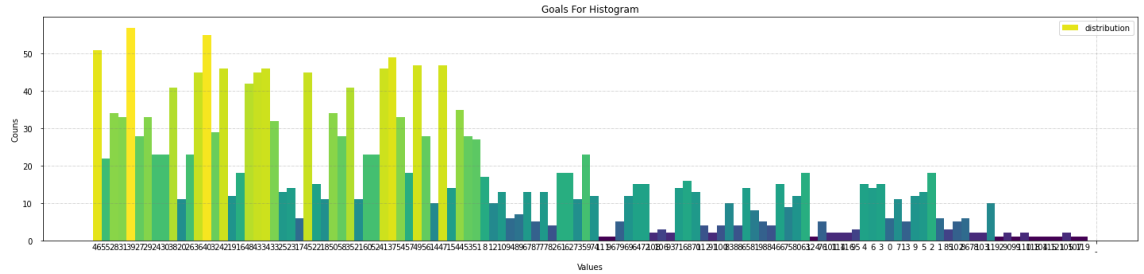
شکل 69: هیستوگرام امتیازات کسب شده تیم‌ها

همان‌طور که در دو جدول بالا نیز می‌توانید که مشاهده کنید در این دو جدول نیز تقریبا مقادیر در یک بازه مشخصی حالا زیر 50 حدودا هستند و عدد خیلی غیر متعارفی را نمی‌توانیم که مشاهده کنیم پس در اینجا نیز outlier نداریم.

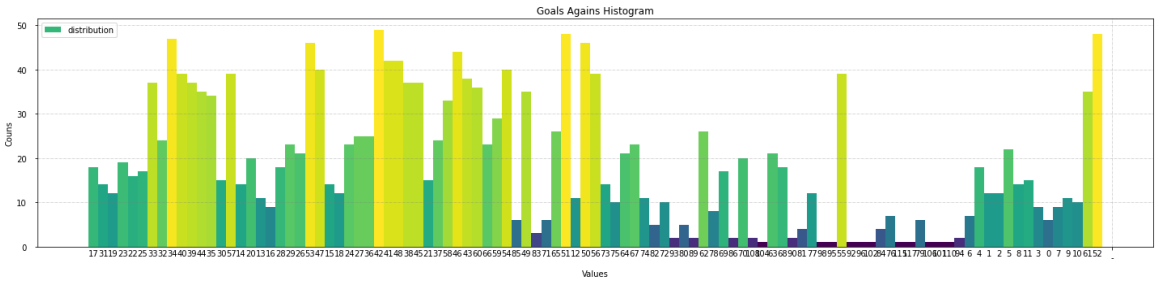
در ادامه نیز از جدول مربوط به جداول‌تیم‌ها مربوط به فوتبال را نشان می‌دهم و به همین ترتیب می‌توان برای بقیه نیز رسم کرد:



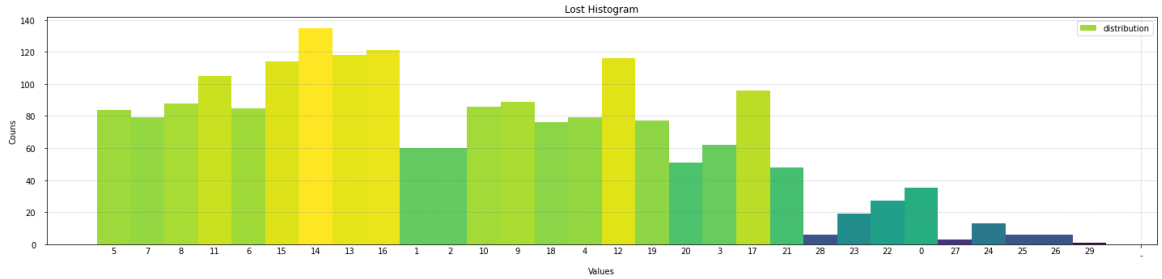
شکل 70



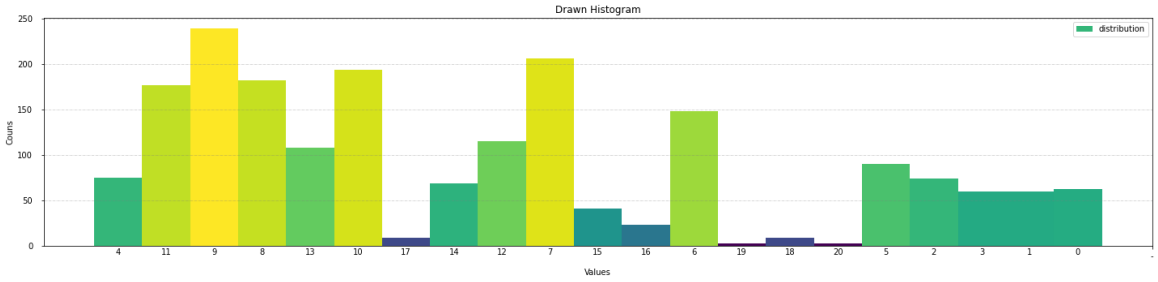
شکل 71:



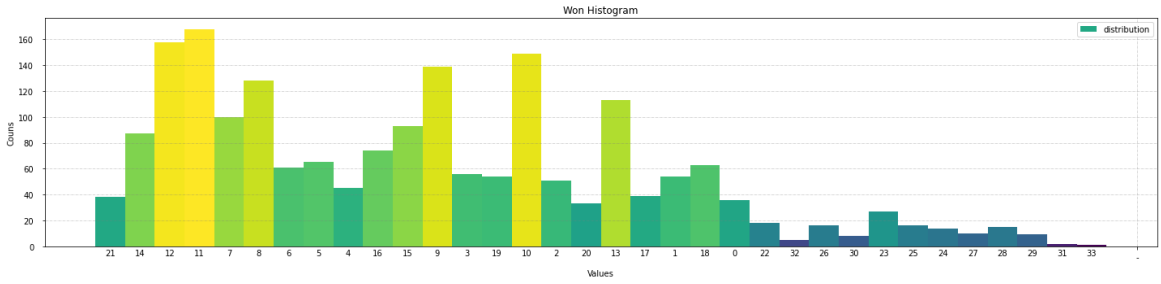
شکل 72:



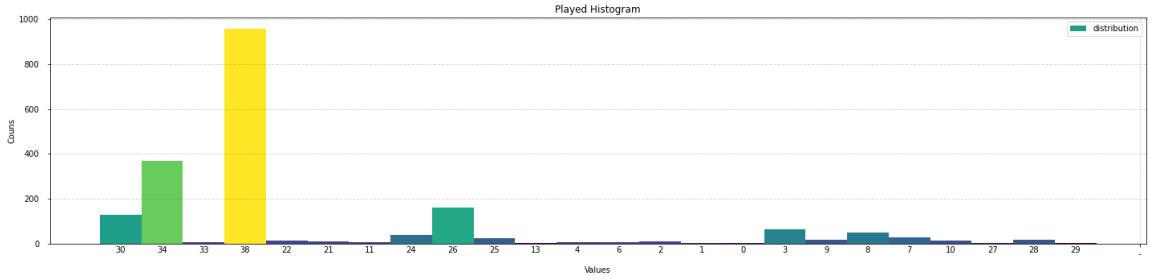
شکل 73:



شکل 74:



شکل 75



شکل 76

همان‌طور که در هیستوگرام‌های بالا نیز دقت کنید باز به همین ترتیب تقریبا تمامی داده‌ها در یک بازه مخصوصی هستند و ما نمی‌توانیم داده outlierای را مشاهده کنیم. پس بنابراین کار استخراج داده ما خوب بوده و مشکل به خصوصی در داده‌های استخراج شده ما وجود ندارد. و می‌توان که به این داده‌ها تا حد خوبی اعتماد کرد. در مورد والیبال و بسکتبال نیز برای این که خیلی زیاد نشوند من دیگر نمودارهای آن‌ها را قرار ندادم.

راه‌های دیگری و نیز نوشتن تست کیس و... نیز برای کنترل کیفیت وجود دارد ولی دیگر خیلی بحث را من تخصصی نمی‌کنم و وارد این مباحث نمی‌شوم.

❖ سوال5:

در این سوال از ما پرسیده شده است که چگونه می‌توان یک چنین پروژه‌ای را در ابعاد بزرگ‌تر انجام داد؟ چه وظایف و چه کارهایی باید برای مرتب‌سازی داده‌ها باید که تعریف شود؟

وقتی که scale و ابعاد کار را در این حوزه گسترش می‌دهیم در نتیجه ما به یک سری چالش‌هایی برخورد می‌کنیم اکنون که در حالت کار با داده‌های محدود نداشتیم!

چالش اول: استخراج داده از ساختارهای Dynamic سایت‌ها:

همان‌طور که در این پروژه دیدیم استخراج کردن داده از فایل‌های HTML ساده مخصوصا اگر کسی در آن مهارت بالایی پیدا کند چالش چندانی ندارد و می‌تواند که خیلی ساده آن را انجام دهد! اما در ابعاد بزرگتر بسیاری از سایت‌ها هستند که به شدت پویا هستند و متکی بر Javascript/AJAX هستند و مدام در حال آپدیت شدن هستند مانند دیتای مربوط به قیمت‌های سهام شرکت‌ها در بورس و ... و در این حالت استخراج داده‌ها می‌تواند که تا حدودی با مشکل مواجه شود!

چالش دوم: تکنولوژی‌های جلوگیری کننده از استخراج داده:

تکنولوژی‌هایی مانند Captcha مواردی نظارتی هستند تا spamها را دور نگه دارند و این موارد یک چالش مهم برای یک استخراج کننده هستند تا بتواند به داده‌های یک سایت دسترسی پیدا کند و بدین منظور ما باید راه‌حلی برای رفع این مشکل نیز پیدا کنیم.

چالش سوم: سرعت لود پایین:

هر چه تعداد صفحاتی که باید استخراج‌کننده از آن‌ها داده استخراج کند بیشتر می‌شوند در نتیجه مدت زمان بیشتری نیز باید صرف شود تا این داده‌ها کامل استخراج شوند. و این موضوع اکنون واضح است که استخراج داده در حجم وسیع مقدار زیادی از منابع سیستم و کامپیوتر ما را تحت اشغال خودش در می‌آورد تا بتواند که این داده‌ها را استخراج کند و این حجم کار سنگین بر روی سیستم ممکن است که موجب breakdown شدن نیز بشود! بنابراین ما باید که راه حتی را برای حل این موضوع نیز داشته باشیم در کارکردن در scale و حجم بالا!

چالش چهارم: Data Warehousing:

استخراج کردن داده‌ها در حجم بالا یک حجم بسیار عظیمی از داده را تولید می‌کند. و این موضوع نیاز به یک زیرساخت بسیار قوی برای ذخیره‌سازی این داده‌ها به صورت امن دارد. و داشتن چنین دیتابیس‌ای نیاز به صرف هزینه بسیار زیاد و نیز زمان زیاد برای نگهداری از آن دارد.

این موارد تعدادی از چالش‌هایی بود که وقتی پروژه ما scale می‌شود در سطوح بالا با آن‌ها مواجه می‌شویم و ما حال باید راه‌حلی متناسب با نوع دیتایی که با آن کار می‌کنیم و نیاز آن شرکت یا فردی که پروژه را به ما داده است، ارائه دهیم تا بتوانیم که این مشکلات را به خوبی رفع کنیم.

همچنین موضوعات دیگری از قبیل زمان‌بندی استخراج داده نیز در خیلی از مواقع اهمیت پیدا می‌کند به عنوان مثال در دیتاهایی مانند دیتای قیمت سهام‌ها در بورس این استخراج داده باید در کسری از ثانیه انجام شود چون دیتا مدام در حال به روز شدن است اما در مورد مثلا دیتای بازی‌های جام جهانی که در همین پروژه نیز داشتیم! کافی است که 4 سالی یک بار که این بازی‌ها انجام می‌شود تنها به جستجو در آن بپردازیم! همچنین در صورتی که نیاز به دیتای یک بازی فوتبال داشته باشیم که بخواهیم دیتای آن را به روز داشته باشیم کافی است که به صورت ساعتی که بازی‌ها انجام می‌شود جستجو انجام دهیم. اما این موارد می‌تواند که بسته به نیاز ما متفاوت باشد!

❖ سوال6:

در این سوال از ما پرسیده شده است که نتایج حاصل از فعالیت تیم خود را چگونه به تیم بعدی ارائه می‌دهیم؟

در این حالت بسته به حجم داده‌ای که تولید کرده‌ایم تصمیم ما می‌تواند متفاوت باشد! در صورتی که حجم دیتای تولید شده توسط ما خیلی زیاد نباشد مانند مثالی که در این پروژه داشتیم که خب می‌توانیم داده‌های خروجی خود را در قالب و فرمت همین فایل‌های اکسل یا csv. را آن‌ها ارائه دهیم یا این که اگر داده‌ها مدام به روز می‌شوند در یک لینکی آن را قرار دهیم که مدام به روز شود و آن‌ها بتوانند که به آخرین نسخه دیتا در هر لحظه دسترسی داشته باشند!

اما خب مواردی هست که آنقدر حجم دیتای استخراجی بالا هست که این روش پاسخگو نیست! در نتیجه در این حالت‌ها باید به فکر یک راهکار جدید باشیم! حتی امکان دارد به دلیل زیاد بودن حجم محاسبات آن را به صورت توزیع شده انجام داده باشیم و هر قسمتی از دیتا بر روی یک سیستم باشد! در چنین حالاتی بهترین کار استفاده از یک دیتابیس هست بسته به نیاز ما می‌تواند این دیتابیس NoSQL یا SQL باشد اما معمولا برای سرعت بالاتر NoSQL را انتخاب می‌کنند و با انتخاب یک دیتابیس می‌توانیم که آن را به صورت توزیع شده (در صورتی که دیتای ما به صورت توزیع شده محاسبه شود) در بیاوریم یا حالت‌های

دیگه و سپس مستقیما خروجی آن را به تیم بعدی که می‌خواهند بر روی این دیتا تحلیل انجام دهند، ارائه دهیم.

من تمامی فایل‌های ایجاد شده را در پوشه‌ای در کنار این فایل گزارش قرار می‌دهم.

و همچنین از ما خواسته شده بود تا در مورد دیتاهایی که استخراج کرده‌ایم توضیح دهیم که من این کار را در ابتدای همین گزارش انجام دادم و در فایل دیگری آن را نمی‌گذارم.

باتشکر از زحمات شما