پروژ پایانی درس تحلیل داده

سپهر کریمی(810100447)

امین قهرمانی (810100439)

بهمن‌ماه ۱۴۰۱ – دانشگاه تهران

چشم‌انداز کلی پروژه

به فرایند تمیز کردن داده، کاوش در داده و فهمیدن داده تحلیل داده گویند. در تحلیل داده اطلاعات سودمند و مفید برای تصمیم‌گیری استخراج می‌شود. بنابرین جمع‌آوری داده و تحلیل و بررسی آن از اهمیت زیادی برخوردار است. برای این امر می توان از الگوریتم‌های یادگیری ماشین استفاده کرد. همانطور که میدانید یادگیری ماشین در حال طی مسیری سریع و روبه جلو در علم کامپیوتر و هوش مصنوعی است. با افرایش داده‌های هوشمند، تجزیه و تحلیل داده‌ها برای پیدا کردن روابط بین چندین ویژگی، یک اصل اساسی برای پیشرفت فناوری در دنیای امروز به شمار میرود بنابراین یکی از کاربردهای مهم یادگیری ماشین، داده کاوی میباشد که با کمک الگوریتم‌های یادگیری مناسب، میتوان نتایج خوبی از داده‌ها استخراج کرد پس یادگیری ماشین و داده کاوی کاملا در راستای هم عمل میکنند.

در این پروژه قصد داریم تا به صورت عملی با دیتا کار کنیم و ضمن استخراج و ذخیره اطلاعات مفید آنها، با کمک الگوریتم های یادگیری، پیش بینی مناسبی انجام دهیم.

قصد داریم تا در بخش‌های مختلف، یک پروژه یادگیری ماشین را بررسی کنیم. ابتدا وب اسکرپینگ را انجام خواهیم داد و سپس سعی خواهیم کرد پیش بینی کنیم چه کسی mvp در یک فصل مشخص nba خواهد بود. برای انجام این‌کار به اطلاعات زیادی در مورد بازیکنان nba و آمار آنها نیاز داریم. بنابراین در این بخش اول از پروژه‌ای که قرار است وب اسکرپینگ را انجام دهیم تا در واقع آن داده هارا بدست آوریم و آن را برای تجزیه و تحالیل در پانداس بارگذاری میکنیم.

هدف: پیش‌بینی کنیم که NBA mvp به چه کسی می‌رود. برای انجام این‌کار، ما به اطلاعاتی در مورد اینکه چه کسی mvps در هر فصل بوده است و همچنین آماری برای هر بازیکن برای هر فصلی که می‌خواهیم پیش‌بینی کنیم، نیاز داریم.

بنابراین به سایتی به نام https://www.basketball-reference.com/ مراجعه میکنیم. این سایت آمارهای زیادی در مورد nba از ابتدا تا به امروز دارند و داده‌های آنها به روشی ساختاریافته و کاملا قالب‌بندی شده است که باعث می‌شود ما به راحتی بتوانیم اسکرپینگ و دانلود کنیم.

از این سایت ما در واقع برای دریافت داده از سه نوع صفحه مختلف به آن نیاز داریم. که هر کدام از آنها را یکی یکی مرور میکنیم.

بخش اول – پرکردن فرم و اعضای گروه

نام و نام خانوادگی	شماره دانشجویی
سپهر کریمی	۸۱۰۱۰۰۴۴۷
امین قهرمانی	۸۱۰۱۰۰۴۳۹

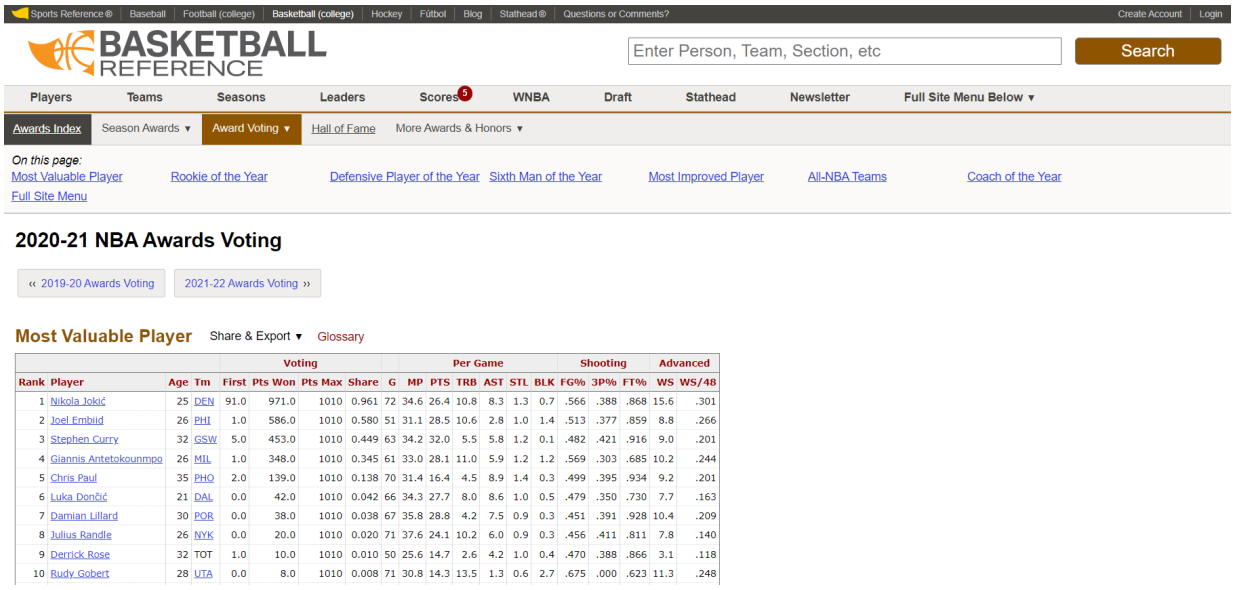
1) DOWNLOADING **MVP** VOTES WITH REQUESTS

2) Parsing The Votes Table With Beautifulsoup

3) Combining MVP Votes With Pandas

این صفحه اطلاعاتی در مورد باارزش ترین بازیکن برای هر فصل دارد. و به ما نشان میدهد برنده‌ای که نیکولا جوکیچ است ۹۷۱ امتیاز از ۱۰۱۰ امتیاز ممکن را بدست آورده است. بنابراین او در واقع جایزه mvp را برده و سپس برنده شده است.

با تغییر آدرس میتوانیم آمار فصل‌های مختلف را دریافت کنیم. بنابراین ما در واقع برای حدود ۲۰ سال داده‌ها را دریافت خواهیم کرد. داده‌ها را تا سال ۱۹۹۱ در یافت خواهیم کرد.



Sports Reference® | Baseball | Football (college) | Basketball (college) | Hockey | Fútbol | Blog | Stathead® | Questions or Comments? | Create Account | Login

BASKETBALL REFERENCE

Enter Person, Team, Section, etc — Search

Players | Teams | Seasons | Leaders | Scores | WNBA | Draft | Stathead | Newsletter | Full Site Menu Below

Awards Index | Season Awards | Award Voting | Hall of Fame | More Awards & Honors

On this page:
Most Valuable Player | Rookie of the Year | Defensive Player of the Year | Sixth Man of the Year | Most Improved Player | All-NBA Teams | Coach of the Year
Full Site Menu

2020-21 NBA Awards Voting

‹‹ 2019-20 Awards Voting | 2021-22 Awards Voting ››

Most Valuable Player Share & Export ▾ Glossary

				Voting					Per Game						Shooting			Advanced	
Rank	Player	Age	Tm	First	Pts Won	Pts Max	Share	G	MP	PTS	TRB	AST	STL	BLK	FG%	3P%	FT%	WS	WS/48
1	Nikola Jokić	25	DEN	91.0	971.0	1010	0.961	72	34.6	26.4	10.8	8.3	1.3	0.7	.566	.388	.868	15.6	.301
2	Joel Embiid	26	PHI	1.0	586.0	1010	0.580	51	31.1	28.5	10.6	2.8	1.0	1.4	.513	.377	.859	8.8	.266
3	Stephen Curry	32	GSW	5.0	453.0	1010	0.449	63	34.2	32.0	5.5	5.8	1.2	0.1	.482	.421	.916	9.0	.201
4	Giannis Antetokounmpo	26	MIL	1.0	348.0	1010	0.345	61	33.0	28.1	11.0	5.9	1.2	1.2	.569	.303	.685	10.2	.244
5	Chris Paul	35	PHO	2.0	139.0	1010	0.138	70	31.4	16.4	4.5	8.9	1.4	0.3	.499	.395	.934	9.2	.201
6	Luka Dončić	21	DAL	0.0	42.0	1010	0.042	66	34.3	27.7	8.0	8.6	1.0	0.5	.479	.350	.730	7.7	.163
7	Damian Lillard	30	POR	0.0	38.0	1010	0.038	67	35.8	28.8	4.2	7.5	0.9	0.3	.451	.391	.928	10.4	.209
8	Julius Randle	26	NYK	0.0	20.0	1010	0.020	71	37.6	24.1	10.2	6.0	0.9	0.3	.456	.411	.811	7.8	.140
9	Derrick Rose	32	TOT	1.0	10.0	1010	0.010	50	25.6	14.7	2.6	4.2	1.0	0.4	.470	.388	.866	3.1	.118
10	Rudy Gobert	28	UTA	0.0	8.0	1010	0.008	71	30.8	14.3	13.5	1.3	0.6	2.7	.675	.000	.623	11.3	.248

Baasketball Reference – شکل ۱

https://www.basketball-reference.com/awards/awards_2021.html

1) DOWNLOADING **PLAYER** STATS

2) Parsing The Stats With Beautifulsoup

3) Combining Player Stats With Pandas

کار بعدی که می خواهیم انجام دهیم این است که اگر می‌خواهیم پیش بینی کنیم چه کسی mvp می شود، تنها داشتن رای mvp کافی نیست، ما به آمار برای همه بازیکنان نیاز داریم، زیرا اگر میخواهید در طول فصل یا نسبت به فصل پیش‌بینی کنیم باید به همه بازیکنان نگاه کنیم و بتوانیم بگوییم آیا آماری وجود دارد که mvp شایسته است یا خیر. بنابراین انچه باید بدست بیاوریم تمام آمار بازیکنان از سال ۱۹۹۱ تا ۲۰۲۱ است تا یک مدل یادگیری ماشین را اموزش دهیم.

2020-21 NBA Season | Standings | Schedule and Results | Leaders | Coaches | Player Stats ▾ | Other ▾ | 2021 Playoffs Summary | Back to top ▲

Totals | Per Game | Per 36 Min | Per 100 Poss | Advanced | Play-by-Play | Shooting | Adjusted Shooting

Player Per Game Share & Export ▾ When table is sorted, hide non-qualifiers for rate stats Glossary Hide Partial Rows

Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	eFG%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
1	Precious Achiuwa	PF	21	MIA	61	4	12.1	2.0	3.7	.544	0.0	0.0	.000	2.0	3.7	.546	.544	0.9	1.8	.509	1.2	2.2	3.4	0.5	0.3	0.5	0.7	1.5	5.0
2	Jaylen Adams	PG	24	MIL	7	0	2.6	0.1	1.1	.125	0.0	0.3	.000	0.1	0.9	.167	.125	0.0	0.0		0.0	0.4	0.4	0.3	0.0	0.0	0.0	0.1	0.3
3	Steven Adams	C	27	NOP	58	58	27.7	3.3	5.3	.614	0.0	0.1	.000	3.3	5.3	.620	.614	1.0	2.3	.444	3.7	5.2	8.9	1.9	0.9	0.7	1.3	1.9	7.6
4	Bam Adebayo	C	23	MIA	64	64	33.5	7.1	12.5	.570	0.0	0.1	.250	7.1	12.4	.573	.571	4.4	5.5	.799	2.2	6.7	9.0	5.4	1.2	1.0	2.6	2.3	18.7
5	LaMarcus Aldridge	C	35	TOT	26	23	25.9	5.4	11.4	.473	1.2	3.1	.388	4.2	8.3	.505	.525	1.6	1.8	.872	0.7	3.8	4.5	1.9	0.4	1.1	1.0	1.8	13.5
5	LaMarcus Aldridge	C	35	SAS	21	18	25.9	5.5	11.8	.464	1.3	3.6	.360	4.2	8.2	.509	.518	1.5	1.8	.838	0.8	3.7	4.5	1.7	0.4	0.9	1.0	1.7	13.7
5	LaMarcus Aldridge	C	35	BRK	5	5	26.0	5.0	9.6	.521	0.8	1.0	.800	4.2	8.6	.488	.563	2.0	2.0	1.000	0.4	4.4	4.8	2.6	0.6	2.2	1.4	2.2	12.8
6	Ty-Shon Alexander	SG	22	PHO	15	0	3.1	0.2	0.8	.250	0.1	0.6	.222	0.1	0.2	.333	.333	0.1	0.1	.500	0.1	0.5	0.7	0.4	0.0	0.1	0.2	0.1	0.6
7	Nickeil Alexander-Walker	SG	22	NOP	46	13	21.9	4.2	10.0	.419	1.7	4.8	.347	2.5	5.2	.485	.502	1.0	1.4	.727	0.3	2.8	3.1	2.2	1.0	0.5	1.5	1.9	11.0
8	Grayson Allen	SG	25	MEM	50	38	25.2	3.5	8.3	.418	2.1	5.5	.391	1.3	2.8	.471	.547	1.6	1.8	.868	0.4	2.8	3.2	2.2	0.9	0.2	1.0	1.4	10.6
9	Jarrett Allen	C	22	TOT	63	45	29.6	4.7	7.7	.618	0.1	0.3	.316	4.6	7.3	.631	.624	3.2	4.6	.703	3.1	6.9	10.0	1.7	0.5	1.4	1.6	1.5	12.8
9	Jarrett Allen	C	22	BRK	12	5	26.7	3.7	5.4	.677	0.0	0.0		3.7	5.4	.677	.677	3.8	5.1	.754	3.2	7.3	10.4	1.7	0.6	1.6	1.8	1.8	11.2
9	Jarrett Allen	C	22	CLE	51	40	30.3	5.0	8.2	.609	0.1	0.4	.316	4.9	7.8	.623	.616	3.1	4.5	.690	3.1	6.8	9.9	1.7	0.5	1.4	1.5	1.5	13.2
10	Al-Farouq Aminu	PF	30	TOT	23	14	18.9	1.7	4.3	.384	0.3	1.6	.216	1.3	2.7	.484	.424	0.8	1.0	.818	1.0	3.8	4.8	1.3	0.8	0.4	1.2	1.3	4.4
10	Al-Farouq Aminu	PF	30	ORL	17	14	21.6	2.1	5.2	.404	0.4	1.8	.226	1.7	3.4	.500	.444	0.8	1.0	.824	1.2	4.2	5.4	1.7	1.0	0.5	1.5	1.3	5.5
10	Al-Farouq Aminu	PF	30	CHI	6	0	11.2	0.3	1.7	.200	0.2	1.0	.167	0.2	0.7	.250	.250	0.7	0.8	.800	0.3	2.8	3.2	0.3	0.3	0.0	0.5	1.2	1.5
11	Kyle Anderson	PF	27	MEM	69	69	27.3	4.5	9.5	.468	1.4	3.8	.360	3.1	5.8	.539	.540	2.1	2.7	.783	0.8	5.0	5.7	3.6	1.2	0.8	1.2	1.7	12.4
12	Giannis Antetokounmpo	PF	26	MIL	61	61	33.0	10.3	18.0	.569	1.1	3.6	.303	9.2	14.4	.636	.600	6.5	9.5	.685	1.6	9.4	11.0	5.9	1.2	1.2	3.4	2.8	28.1
13	Kostas Antetokounmpo	PF	23	LAL	15	0	3.7	0.2	0.7	.300	0.0	0.0		0.2	0.7	.300	.300	0.4	0.9	.462	0.3	1.0	1.3	0.1	0.1	0.3	0.7	0.5	0.8
14	Thanasis Antetokounmpo	SF	28	MIL	57	3	9.7	1.2	2.4	.489	0.1	0.5	.241	1.1	1.9	.556	.515	0.4	0.9	.510	0.9	1.2	2.2	0.8	0.4	0.2	0.8	1.3	2.9
15	Carmelo Anthony	PF	36	POR	69	3	24.5	4.7	11.3	.421	1.9	4.7	.409	2.8	6.6	.429	.506	2.0	2.2	.890	0.5	2.6	3.1	1.5	0.7	0.6	0.9	2.1	13.4
16	Cole Anthony	PG	20	ORL	47	34	27.1	4.7	11.7	.397	1.2	3.7	.337	3.4	8.1	.424	.449	2.3	2.8	.832	0.8	3.9	4.7	4.1	0.6	0.4	2.3	2.1	12.9
17	OG Anunoby	SF	23	TOR	43	43	33.3	5.8	12.1	.480	2.4	6.1	.398	3.4	6.0	.562	.580	1.9	2.4	.784	1.3	4.3	5.5	2.2	1.5	0.7	1.7	2.7	15.9

Baasketball Reference – شکل ۲

https://www.basketball-reference.com/leagues/NBA_2021_per_game.html

1) DOWNLOADING **TEAM** DATA

2) Parsing The Stats With Beautifulsoup

3) Combining Team Stats With Pandas

و آخرین چیزی که ما باید پیش بینی کنیم تیم است. بنابراین وقتی پیش‌بینی میکنیم چه کسی برنده MVP میشود، میخواهیم مطمئن شویم که رکورد تیم آنها را درج میکنیم تا مدل یادگیری

ماشـین بتوانـد ان رکـورد را ببینـد و از آن بـه عنـوان پیش‌بینـی کننـده اسـتفاده کنـد. و چیـزی کـه بایـد اسکرپ کنیم رکوردهای تیم در سال است.

Division Standings

* Playoff teams

Eastern Conference	W	L	W/L%	GB	PS/G	PA/G	SRS
Atlantic Division							
Philadelphia 76ers*	49	23	.681	—	113.6	108.1	5.28
Brooklyn Nets*	48	24	.667	1.0	118.6	114.1	4.24
New York Knicks*	41	31	.569	8.0	107.0	104.7	2.13
Boston Celtics*	36	36	.500	13.0	112.6	111.2	1.32
Toronto Raptors	27	45	.375	22.0	111.3	111.7	-0.54
Central Division							
Milwaukee Bucks*	46	26	.639	—	120.1	114.2	5.57
Indiana Pacers	34	38	.472	12.0	115.3	115.3	-0.13
Chicago Bulls	31	41	.431	15.0	110.7	111.6	-0.94
Cleveland Cavaliers	22	50	.306	24.0	103.8	112.3	-8.19
Detroit Pistons	20	52	.278	26.0	106.6	111.1	-4.38
Southeast Division							
Atlanta Hawks*	41	31	.569	—	113.7	111.4	2.14
Miami Heat*	40	32	.556	1.0	108.1	108.0	-0.06
Washington Wizards*	34	38	.472	7.0	116.6	118.5	-1.85
Charlotte Hornets	33	39	.458	8.0	109.5	111.4	-1.94
Orlando Magic	21	51	.292	20.0	104.0	113.3	-9.02

Western Conference	W	L	W/L%	GB	PS/G	PA/G	SRS
Northwest Division							
Utah Jazz*	52	20	.722	—	116.4	107.2	8.97
Denver Nuggets*	47	25	.653	5.0	115.1	110.1	4.82
Portland Trail Blazers*	42	30	.583	10.0	116.1	114.3	1.81
Minnesota Timberwolves	23	49	.319	29.0	112.1	117.7	-5.25
Oklahoma City Thunder	22	50	.306	30.0	105.0	115.6	-10.13
Pacific Division							
Phoenix Suns*	51	21	.708	—	115.3	109.5	5.67
Los Angeles Clippers*	47	25	.653	4.0	114.0	107.8	6.02
Los Angeles Lakers*	42	30	.583	9.0	109.5	106.8	2.77
Golden State Warriors	39	33	.542	12.0	113.7	112.7	1.10
Sacramento Kings	31	41	.431	20.0	113.7	117.4	-3.45
Southwest Division							
Dallas Mavericks*	42	30	.583	—	112.4	110.2	2.26
Memphis Grizzlies*	38	34	.528	4.0	113.3	112.3	1.07
San Antonio Spurs	33	39	.458	9.0	111.1	112.8	-1.58
New Orleans Pelicans	31	41	.431	11.0	114.6	114.9	-0.20
Houston Rockets	17	55	.236	25.0	108.8	116.7	-7.50

Baasketball Reference – ۳ شکل

https://www.basketball-reference.com/leagues/NBA_2021_standings.html

```
import requests
import os
import shutil
from bs4 import BeautifulSoup
import pandas as pd
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
```

```
years = list(range(1991,2022))

url_start = "https://www.basketball-reference.com/awards/awards_{}.html"

for year in years:
    url = url_start.format(year)

    data = requests.get(url)

    with open("mvp/{}.html".format(year), "w+", encoding="utf-8") as f:
        f.write(data.text)
```

```
dfs = []
for year in years:
    with open("mvp/{}.html".format(year), encoding="utf-8") as f:
        page = f.read()

    soup = BeautifulSoup(page, 'html.parser')
    soup.find('tr', class_="over_header").decompose()
    mvp_table = soup.find_all(id="mvp")[0]
    mvp_df = pd.read_html(str(mvp_table))[0]
    mvp_df["Year"] = year
    dfs.append(mvp_df)
```

DOWNLOADING TEAM DATA - شکل ۴

بخش سوم - *Cleaning NBA Stats Data With Python And Pandas*

سوال اول:

1) Cleaning The MVP Vote Data

تمیز کردن بخش نادیده گرفته شده در علم داده است. اگر شما دانشمند داده یا یک تحلیلگر داده باشید، ۸۰ درصد از زمان خود را صرف میکنید و این یک تخمین محافظه‌کارانه برای انجام این کار است و برای انجام این کار ما قصد داریم از کتابخانه‌ای به نام Pandas استفاده کنیم

برای نصب کتابخانه پانداس از دستور زیر در Jupyter Notebook استفاده میکنیم

!Pip install pandas

ابتدا فایلMVP.csv خود را بارگذاری می‌کنیم:

```
import pandas as pd
```

```
mvps = pd.read_csv("mvps.csv")
mvps
```

	Unnamed: 0	Rank	Player	Age	Tm	First	Pts Won	Pts Max	Share	G	...	TRB	AST	STL	BLK	FG%	3P%	FT%	WS	WS/48	Year
0	0	1	Michael Jordan	27	CHI	77.0	891.0	960	0.928	82	...	6.0	5.5	2.7	1.0	0.539	0.312	0.851	20.3	0.321	1991
1	1	2	Magic Johnson	31	LAL	10.0	497.0	960	0.518	79	...	7.0	12.5	1.3	0.2	0.477	0.320	0.906	15.4	0.251	1991
2	2	3	David Robinson	25	SAS	6.0	476.0	960	0.496	82	...	13.0	2.5	1.5	3.9	0.552	0.143	0.762	17.0	0.264	1991
3	3	4	Charles Barkley	27	PHI	2.0	222.0	960	0.231	67	...	10.1	4.2	1.6	0.5	0.570	0.284	0.722	13.4	0.258	1991
4	4	5	Karl Malone	27	UTA	0.0	142.0	960	0.148	82	...	11.8	3.3	1.1	1.0	0.527	0.286	0.770	15.5	0.225	1991
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
469	10	11	Russell Westbrook	32	WAS	0.0	5.0	1010	0.005	65	...	11.5	11.7	1.4	0.4	0.439	0.315	0.656	3.7	0.075	2021
470	11	12	Ben Simmons	24	PHI	0.0	3.0	1010	0.003	58	...	7.2	6.9	1.6	0.6	0.557	0.300	0.613	6.0	0.153	2021
471	12	13T	James Harden	31	TOT	0.0	1.0	1010	0.001	44	...	7.9	10.8	1.2	0.8	0.466	0.362	0.861	7.0	0.208	2021
472	13	13T	LeBron James	36	LAL	0.0	1.0	1010	0.001	45	...	7.7	7.8	1.1	0.6	0.513	0.365	0.698	5.6	0.179	2021
473	14	13T	Kawhi Leonard	29	LAC	0.0	1.0	1010	0.001	52	...	6.5	5.2	1.6	0.4	0.512	0.398	0.885	8.8	0.238	2021

474 rows × 22 columns

شکل ۵ - بارگذاری دیتای mvp

باید از شر برخی از ستوت‌های اضافی خلاص شویم(مثلا: g، TRB و ...). ستون‌های اضافی را که در پیش‌بینی به ما کمکی نمی‌کنند را حذف میکنیم. فیچر یا ستون‌های مهم ما در این دیتاست موارد زیر است.

```
mvps = mvps[["Player", "Year", "Pts Won", "Pts Max", "Share"]]
```

```
mvps.head()
```

	Player	Year	Pts Won	Pts Max	Share
0	Michael Jordan	1991	891.0	960	0.928
1	Magic Johnson	1991	497.0	960	0.518
2	David Robinson	1991	476.0	960	0.496
3	Charles Barkley	1991	222.0	960	0.231
4	Karl Malone	1991	142.0	960	0.148

شکل ۶ - فیچر های مهم دیتاست NBA

با استفاده از این فیچرهای مهم تشخیص اینکه چه اتفاقی در حال رخ دادن است بسیار آسان‌تر است. و از شر ستون‌های تکراری که به آن‌ها نیاز نداشتیم خلاص شدیم.

۲) Cleaning The Player Data

در این قسمت، دادهای Player.csv را بارگذاری و سپس پاکسازی می‌کنیم.

```
players = pd.read_csv("players.csv")
players
```

	Unnamed: 0	Rk	Player	Pos	Age	Tm	G	GS	MP	FG	...	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	Year
0	0	1	Alaa Abdelnaby	PF	22	POR	43	0	6.7	1.3	...	0.6	1.4	2.1	0.3	0.1	0.3	0.5	0.9	3.1	1991
1	1	2	Mahmoud Abdul-Rauf	PG	21	DEN	67	19	22.5	6.2	...	0.5	1.3	1.8	3.1	0.8	0.1	1.6	2.2	14.1	1991
2	2	3	Mark Acres	C	28	ORL	68	0	19.3	1.6	...	2.1	3.2	5.3	0.4	0.4	0.4	0.6	3.2	4.2	1991
3	3	4	Michael Adams	PG	28	DEN	66	66	35.5	8.5	...	0.9	3.0	3.9	10.5	2.2	0.1	3.6	2.5	26.5	1991
4	4	5	Mark Aguirre	SF	31	DET	78	13	25.7	5.4	...	1.7	3.1	4.8	1.8	0.6	0.3	1.6	2.7	14.2	1991
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18039	725	536	Delon Wright	PG	28	SAC	27	8	25.8	3.9	...	1.0	2.9	3.9	3.6	1.6	0.4	1.3	1.1	10.0	2021
18040	726	537	Thaddeus Young	PF	32	CHI	68	23	24.3	5.4	...	2.5	3.8	6.2	4.3	1.1	0.6	2.0	2.2	12.1	2021
18041	727	538	Trae Young	PG	22	ATL	63	63	33.7	7.7	...	0.6	3.3	3.9	9.4	0.8	0.2	4.1	1.8	25.3	2021
18042	728	539	Cody Zeller	C	28	CHO	48	21	20.9	3.8	...	2.5	4.4	6.8	1.8	0.6	0.4	1.1	2.5	9.4	2021
18043	729	540	Ivica Zubac	C	23	LAC	72	33	22.3	3.6	...	2.6	4.6	7.2	1.3	0.3	0.9	1.1	2.6	9.0	2021

18044 rows × 32 columns

شکل ۷ – بارگذاری دیتای player

بنابراین کاری که ما باید انجام دهیم این است که این دیتا را پاکسازی کنیم و سپس آن را با داده‌های MVP ترکیب کنیم. ما برای هر بازیکن vote داریم و می‌توانیم بفهمیم که هر بازیکن چند رای mvp بدست آورده است. ستون‌های اضافی را دراینجا حذف میکنیم. دو ستون اول را حذف میکنیم.

```
del players["Unnamed: 0"]
```

```
del players["Rk"]
```

کاری که باید انجام دهیم این است که برخی از نام های Player را اصلاح کنیم زیرا اگر به داده های بازیکنان نگاه کنیم میبینم که در بعضی از اسامی بازیکنان * وجود دارد.(زیرا بازیکنانی که علامت * در آخر اسم آنها وجود دارد به این معنی است که جز بهترین بازیکنان هستند که ستاره را با نام خود داشته‌اند و باید از شر این ستاره ها هم خلاص شویم زیرا این امر ادغام براساس نام بازیکن را غیر ممکن میکند)

2	Mark Acres
3	Michael Adams
4	Mark Aguirre
5	Danny Ainge
6	Mark Alarie
7	Steve Alford
8	Greg Anderson
9	Greg Anderson
10	Greg Anderson
11	Greg Anderson
12	Nick Anderson
13	Ron Anderson
14	Willie Anderson
15	Michael Ansley
16	B.J. Armstrong
17	Vincent Askew
18	Keith Askins
19	Miloš Babić
20	Thurl Bailey
21	Cedric Ball
22	Ken Bannister
23	Charles Barkley*
24	Dana Barros
25	John Battle
26	Kenny Battle
27	Kenny Battle
28	Kenny Battle
29	William Bedford
30	Benoit Benjamin
31	Benoit Benjamin
32	Benoit Benjamin
33	Winston Bennett
34	Larry Bird*
35	Rolando Blackman
36	Lance Blanks
37	Mookie Blaylock
38	Muggsy Bogues
39	Manute Bol

شکل ۸ – لیست بازیکنان

برای این حذف این ستاره‌ها ما از روش str.replace استفاده خواهیم کرد.

```
players["Player"] = players["Player"].str.replace("*","", regex=False)
```

بنابراین ما با استفاده از حذف ستاره از نام بازیکنان به راحتی میتوانیم دیتاست MVP و Player را با هم ادغام کنیم.

یک مشکل دیگر این دیتا این است که وقتی دیتایی Player را بررسی میکنیم متوجه میشویم که Greg Anderson چندین ردیف برای سال ۱۹۹۱ دارد. اما ما نمیخواهیم که هر بازیکن یک ردیف در داده‌ها داشته باشد. و دلیل اینکه او چندین ردیف دارد این است که او برای چندین تیم مختلف بازی کرده است(MIL، NJN، DEN). بنابراین باید از شر این سطرهای تکراری هم خلاص شویم زیرا می خواهیم مطمئن شویم که هر بازیکن قفقط یک ردیف در داده‌ها دارد. بنابراین میتوانیم پیش بینی کنیم که هر بازیکن چقدر Vote می‌آورد.

	Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	...	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	Year
0	1	Alaa Abdelnaby	PF	22	POR	43	0	6.7	1.3	2.7	...	0.6	1.4	2.1	0.3	0.1	0.3	0.5	0.9	3.1	1991
1	2	Mahmoud Abdul-Rauf	PG	21	DEN	67	19	22.5	6.2	15.1	...	0.5	1.3	1.8	3.1	0.8	0.1	1.6	2.2	14.1	1991
2	3	Mark Acres	C	28	ORL	68	0	19.3	1.6	3.1	...	2.1	3.2	5.3	0.4	0.4	0.4	0.6	3.2	4.2	1991
3	4	Michael Adams	PG	28	DEN	66	66	35.5	8.5	21.5	...	0.9	3.0	3.9	10.5	2.2	0.1	3.6	2.5	26.5	1991
4	5	Mark Aguirre	SF	31	DET	78	13	25.7	5.4	11.7	...	1.7	3.1	4.8	1.8	0.6	0.3	1.6	2.7	14.2	1991
5	6	Danny Ainge	SG	31	POR	80	0	21.4	4.2	8.9	...	0.6	2.0	2.6	3.6	0.8	0.2	1.3	2.4	11.1	1991
6	7	Mark Alarie	PF	27	WSB	42	1	14.0	2.4	5.4	...	1.0	1.8	2.8	1.1	0.4	0.2	1.0	2.1	5.8	1991
7	8	Steve Alford	PG	26	DAL	34	0	6.9	1.7	3.4	...	0.3	0.4	0.7	0.6	0.2	0.0	0.5	0.3	4.4	1991
8	9	Greg Anderson	PF	26	TOT	68	2	13.6	1.7	4.0	...	1.4	3.3	4.7	0.2	0.5	0.7	1.2	2.1	4.3	1991
9	9	Greg Anderson	PF	26	MIL	26	0	9.5	1.0	2.8	...	1.0	1.9	2.9	0.1	0.3	0.3	0.8	1.1	2.7	1991
10	9	Greg Anderson	PF	26	NJN	1	0	18.0	4.0	4.0	...	4.0	2.0	6.0	1.0	2.0	0.0	1.0	4.0	8.0	1991
11	9	Greg Anderson	PF	26	DEN	41	2	16.1	2.1	4.7	...	1.6	4.1	5.8	0.3	0.6	0.9	1.5	2.6	5.2	1991
12	10	Nick Anderson	SG	23	ORL	70	42	28.2	5.7	12.2	...	1.3	4.2	5.5	1.5	1.1	0.6	1.6	2.1	14.1	1991
13	11	Ron Anderson	SF	32	PHI	82	13	28.5	6.2	12.9	...	1.3	3.2	4.5	1.4	0.8	0.2	1.2	2.0	14.6	1991
14	12	Willie Anderson	SG	24	SAS	75	75	34.6	6.0	13.2	...	0.9	3.8	4.7	4.8	1.1	0.6	2.2	3.0	14.4	1991
15	13	Michael Ansley	SF	23	ORL	67	1	13.1	2.1	3.9	...	1.8	2.0	3.8	0.4	0.4	0.1	0.5	1.9	5.7	1991
16	14	B.J. Armstrong	PG	23	CHI	82	0	21.1	3.7	7.7	...	0.3	1.5	1.8	3.7	0.9	0.0	1.3	1.4	8.8	1991
17	15	Vincent Askew	SG	24	GSW	7	0	12.1	1.7	3.6	...	1.0	0.6	1.6	1.9	0.3	0.0	0.9	3.0	4.7	1991
18	16	Keith Askins	SF	23	MIA	39	1	6.8	0.9	2.1	...	0.8	1.0	1.7	0.5	0.4	0.3	0.3	1.2	2.2	1991
19	17	Miloš Babić	PF	22	CLE	12	0	4.3	0.5	1.6	...	0.5	0.3	0.8	0.3	0.1	0.1	0.4	0.6	1.6	1991

شکل ۹ – دیتاست player

برای اینکار از دستور groupby استفاده میکنیم تا بتوانیم بگوییم بر اساس بازیکن گروه‌بندی شده است.

```
def single_team(df):
    if df.shape[0]==1:
        return df
    else:
        row = df[df["Tm"]=="TOT"]
        row["Tm"] = df.iloc[-1,:]["Tm"]
        return row

players = players.groupby(["Player", "Year"]).apply(single_team)
```

			Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	...	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	Year
Player	**Year**																						
A.C. Green	**1991**	**164**	A.C. Green	PF	27	LAL	82	21	26.4	3.1	6.6	.476	...	2.5	3.8	6.3	0.9	0.7	0.3	1.2	1.4	9.1	1991
	1992	**633**	A.C. Green	PF	28	LAL	82	53	35.4	4.7	9.8	.476	...	3.7	5.6	9.3	1.4	1.1	0.4	1.4	1.7	13.6	1992
	1993	**1092**	A.C. Green	PF	29	LAL	82	55	34.4	4.6	8.6	.537	...	3.5	5.2	8.7	1.4	1.1	0.5	1.4	1.8	12.8	1993
	1994	**1579**	A.C. Green	PF	30	PHO	82	55	34.5	5.7	11.3	.502	...	3.4	5.8	9.2	1.7	0.9	0.5	1.2	1.7	14.7	1994
	1995	**2067**	A.C. Green	SF	31	PHO	82	52	32.8	3.8	7.5	.504	...	2.4	5.8	8.2	1.5	0.7	0.4	1.4	1.8	11.2	1995
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Željko Rebrača	**2002**	**6095**	Željko Rebrača	C	29	DET	74	4	15.9	2.6	5.1	.505	...	1.1	2.8	3.9	0.5	0.4	1.0	1.1	2.6	6.9	2002
	2003	**6595**	Željko Rebrača	C	30	DET	30	12	16.3	2.7	4.8	.552	...	0.9	2.2	3.1	0.3	0.2	0.6	1.0	2.6	6.6	2003
	2004	**7176**	Željko Rebrača	C	31	ATL	24	2	11.4	1.4	3.2	.442	...	1.0	1.5	2.4	0.3	0.2	0.5	0.7	2.2	3.8	2004
	2005	**7776**	Željko Rebrača	C	32	LAC	58	2	16.0	2.3	4.0	.568	...	0.8	2.3	3.2	0.4	0.2	0.7	0.8	2.2	5.8	2005
	2006	**8370**	Željko Rebrača	C	33	LAC	29	2	14.2	1.8	3.3	.542	...	0.4	1.8	2.2	0.3	0.2	0.7	0.8	2.0	4.7	2006

14092 rows × 30 columns

groupby for player dataset – ۱۰ شکل

Combining The Player And MVP Data

اکنون کاری که میتوانیم انجام دهیم این است که در واقع دو فریم داده‌های خود(MVP و Player) را با هم ادغام کنیم. قبلا اشاره کردیم که میخواهیم نام بازیکن و سال را ادغام کنیم.

توجه: داده‌های MVP فقط برای افرادی که برنده MVP شده‌اند است. اما داده‌های Player داده‌هایی برای هر بازیکن در هر فصل دارد.

```
combined = players.merge(mvps, how="outer", on=["Player", "Year"])
```

```
combined[combined["Pts Won"] > 0]
```

	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	...	AST	STL	BLK	TOV	PF	PTS	Year	Pts Won	Pts Max	Share
187	Al Jefferson	C	29	CHA	73	73	35.0	9.6	18.8	.509	...	2.1	0.9	1.1	1.7	2.4	21.8	2014	34.0	1250.0	0.027
329	Allen Iverson	PG	21	PHI	76	74	40.1	8.2	19.8	.416	...	7.5	2.1	0.3	4.4	3.1	23.5	1997	1.0	1150.0	0.001
331	Allen Iverson	SG	23	PHI	48	48	41.5	9.1	22.0	.412	...	4.6	2.3	0.1	3.5	2.0	26.8	1999	319.0	1180.0	0.270
332	Allen Iverson	SG	24	PHI	70	70	40.8	10.4	24.8	.421	...	4.7	2.1	0.1	3.3	2.3	28.4	2000	132.0	1210.0	0.109
333	Allen Iverson	SG	25	PHI	71	71	42.0	10.7	25.5	.420	...	4.6	2.5	0.3	3.3	2.1	31.1	2001	1121.0	1240.0	0.904
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13587	Vince Carter	SF	23	TOR	82	82	38.1	9.6	20.7	.465	...	3.9	1.3	1.1	2.2	3.2	25.7	2000	51.0	1210.0	0.042
13588	Vince Carter	SF	24	TOR	75	75	39.7	10.2	22.1	.460	...	3.9	1.5	1.1	2.2	2.7	27.6	2001	7.0	1240.0	0.006
13592	Vince Carter	SF-SG	28	NJN	77	76	36.7	9.0	20.0	.452	...	4.2	1.4	0.6	2.2	3.2	24.5	2005	3.0	1270.0	0.002
13952	Yao Ming	C	23	HOU	82	82	32.8	6.5	12.5	.522	...	1.5	0.3	1.9	2.5	3.3	17.5	2004	1.0	1230.0	0.001
13957	Yao Ming	C	28	HOU	77	77	33.6	7.4	13.4	.548	...	1.8	0.4	1.9	3.0	3.3	19.7	2009	1.0	1210.0	0.001

474 rows × 33 columns

	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	...	AST	STL	BLK	TOV	PF	PTS	Year	Pts Won	Pts Max	Share
0	A.C. Green	PF	27	LAL	82	21	26.4	3.1	6.6	.476	...	0.9	0.7	0.3	1.2	1.4	9.1	1991	NaN	NaN	NaN
1	A.C. Green	PF	28	LAL	82	53	35.4	4.7	9.8	.476	...	1.4	1.1	0.4	1.4	1.7	13.6	1992	NaN	NaN	NaN
2	A.C. Green	PF	29	LAL	82	55	34.4	4.6	8.6	.537	...	1.4	1.1	0.5	1.4	1.8	12.8	1993	NaN	NaN	NaN
3	A.C. Green	PF	30	PHO	82	55	34.5	5.7	11.3	.502	...	1.7	0.9	0.5	1.2	1.7	14.7	1994	NaN	NaN	NaN
4	A.C. Green	SF	31	PHO	82	52	32.8	3.8	7.5	.504	...	1.5	0.7	0.4	1.4	1.8	11.2	1995	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
14087	Željko Rebrača	C	29	DET	74	4	15.9	2.6	5.1	.505	...	0.5	0.4	1.0	1.1	2.6	6.9	2002	NaN	NaN	NaN
14088	Željko Rebrača	C	30	DET	30	12	16.3	2.7	4.8	.552	...	0.3	0.2	0.6	1.0	2.6	6.6	2003	NaN	NaN	NaN
14089	Željko Rebrača	C	31	ATL	24	2	11.4	1.4	3.2	.442	...	0.3	0.2	0.5	0.7	2.2	3.8	2004	NaN	NaN	NaN
14090	Željko Rebrača	C	32	LAC	58	2	16.0	2.3	4.0	.568	...	0.4	0.2	0.7	0.8	2.2	5.8	2005	NaN	NaN	NaN
14091	Željko Rebrača	C	33	LAC	29	2	14.2	1.8	3.3	.542	...	0.3	0.2	0.7	0.8	2.0	4.7	2006	NaN	NaN	NaN

14092 rows × 33 columns

شکل ۱۱ – مقادیر NaN در دیتاست player

هدف بعدی ما این است که باید مقادیر NaN را حل کنیم. فیچرهای "Pts Won", "Pts Max", "Share" را با استفاده از دستور زیر با مقدار صفرر پر می‌کنیم(زیرا هیچ Voteای نگرفته‌اند).

```
combined[["Pts Won", "Pts Max", "Share"]] = combined[["Pts Won", "Pts Max", "Share"]].fillna(0)
```

```
combined
```

	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	...	AST	STL	BLK	TOV	PF	PTS	Year	Pts Won	Pts Max	Share
0	A.C. Green	PF	27	LAL	82	21	26.4	3.1	6.6	.476	...	0.9	0.7	0.3	1.2	1.4	9.1	1991	0.0	0.0	0.0
1	A.C. Green	PF	28	LAL	82	53	35.4	4.7	9.8	.476	...	1.4	1.1	0.4	1.4	1.7	13.6	1992	0.0	0.0	0.0
2	A.C. Green	PF	29	LAL	82	55	34.4	4.6	8.6	.537	...	1.4	1.1	0.5	1.4	1.8	12.8	1993	0.0	0.0	0.0
3	A.C. Green	PF	30	PHO	82	55	34.5	5.7	11.3	.502	...	1.7	0.9	0.5	1.2	1.7	14.7	1994	0.0	0.0	0.0
4	A.C. Green	SF	31	PHO	82	52	32.8	3.8	7.5	.504	...	1.5	0.7	0.4	1.4	1.8	11.2	1995	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
14087	Željko Rebrača	C	29	DET	74	4	15.9	2.6	5.1	.505	...	0.5	0.4	1.0	1.1	2.6	6.9	2002	0.0	0.0	0.0
14088	Željko Rebrača	C	30	DET	30	12	16.3	2.7	4.8	.552	...	0.3	0.2	0.6	1.0	2.6	6.6	2003	0.0	0.0	0.0
14089	Željko Rebrača	C	31	ATL	24	2	11.4	1.4	3.2	.442	...	0.3	0.2	0.5	0.7	2.2	3.8	2004	0.0	0.0	0.0
14090	Željko Rebrača	C	32	LAC	58	2	16.0	2.3	4.0	.568	...	0.4	0.2	0.7	0.8	2.2	5.8	2005	0.0	0.0	0.0
14091	Željko Rebrača	C	33	LAC	29	2	14.2	1.8	3.3	.542	...	0.3	0.2	0.7	0.8	2.0	4.7	2006	0.0	0.0	0.0

شکل ۱۲ – دیتاست plater بعد از مقداردهی NaN ها

Cleaning The Team Data

در این قسمت، دادهای Team.csv را بارگذاری و سپس پاکسازی می‌کنیم.

```
teams = pd.read_csv("teams.csv")
teams
```

	Unnamed: 0	W	L	W/L%	GB	PS/G	PA/G	SRS	Year	Team
0	0	56	26	.683	—	111.5	105.7	5.22	1991	Boston Celtics*
1	1	44	38	.537	12.0	105.4	105.6	-0.39	1991	Philadelphia 76ers*
2	2	39	43	.476	17.0	103.1	103.3	-0.43	1991	New York Knicks*
3	3	30	52	.366	26.0	101.4	106.4	-4.84	1991	Washington Bullets
4	4	26	56	.317	30.0	102.9	107.5	-4.53	1991	New Jersey Nets
...	...	...	...	...	...	...	...	...	...	...
993	13	44	28	.611	—	117.8	114.8	3.13	2020	Houston Rockets*
994	14	43	32	.573	2.5	117.0	112.1	4.87	2020	Dallas Mavericks*
995	15	34	39	.466	10.5	112.6	113.7	-0.91	2020	Memphis Grizzlies
996	16	32	39	.451	11.5	114.1	115.2	-0.65	2020	San Antonio Spurs
997	17	30	42	.417	14.0	115.8	117.1	-0.55	2020	New Orleans Pelicans

998 rows × 10 columns

شکل ۱۳ - بارگذاری دیتاست Team

همانطور که مشخص است ما در این دیتاست، رکوردهای شکست تیم را داریم.

```
teams.head(20)
```

	Unnamed: 0	W	L	W/L%	GB	PS/G	PA/G	SRS	Year	Team
0	0	56	26	.683	—	111.5	105.7	5.22	1991	Boston Celtics*
1	1	44	38	.537	12.0	105.4	105.6	-0.39	1991	Philadelphia 76ers*
2	2	39	43	.476	17.0	103.1	103.3	-0.43	1991	New York Knicks*
3	3	30	52	.366	26.0	101.4	106.4	-4.84	1991	Washington Bullets
4	4	26	56	.317	30.0	102.9	107.5	-4.53	1991	New Jersey Nets
5	5	24	58	.293	32.0	101.8	107.8	-5.91	1991	Miami Heat
6	6	Central Division	Central Division	Central Division	Central Division	Central Division	Central Division	Central Division	1991	Central Division
7	7	61	21	.744	—	110.0	101.0	8.57	1991	Chicago Bulls*
8	8	50	32	.610	11.0	100.1	96.8	3.08	1991	Detroit Pistons*
9	9	48	34	.585	13.0	106.4	104.0	2.33	1991	Milwaukee Bucks*
10	10	43	39	.524	18.0	109.8	109.0	0.72	1991	Atlanta Hawks*
11	11	41	41	.500	20.0	111.7	112.1	-0.37	1991	Indiana Pacers*
12	12	33	49	.402	28.0	101.7	104.2	-2.33	1991	Cleveland Cavaliers
13	13	26	56	.317	35.0	102.8	108.0	-4.95	1991	Charlotte Hornets
14	0	Midwest Division	Midwest Division	Midwest Division	Midwest Division	Midwest Division	Midwest Division	Midwest Division	1991	Midwest Division
15	1	55	27	.671	—	107.1	102.6	4.30	1991	San Antonio Spurs*
16	2	54	28	.659	1.0	104.0	100.7	3.18	1991	Utah Jazz*
17	3	52	30	.634	3.0	106.7	103.2	3.27	1991	Houston Rockets*
18	4	31	51	.378	24.0	105.9	109.9	-3.79	1991	Orlando Magic
19	5	29	53	.354	26.0	99.6	103.5	-3.75	1991	Minnesota Timberwolves

شکل ۱۴ – دیتاست player داریا تقسیم بندی در بعضی از ردیف‌ها

برای پاکسازی این قسمت، همانطور که مشاهده میکنیم بعضی از ردیف‌ها داری تقسیم بندی است که باید حذف شوند. و روشی که ما انجام میدهیم این است که میگوییم تیم‌ها برابر با تیم‌ها هستند نه تین‌هایی که در ستون w قرار دارند. و str.contains چک میکند که آیا این رشته دارای تقسیم است یا خیر.

```
teams = teams[~teams["W"].str.contains("Division")].copy()
teams.head(20)
```

	Unnamed: 0	W	L	W/L%	GB	PS/G	PA/G	SRS	Year	Team
0	0	56	26	.683	—	111.5	105.7	5.22	1991	Boston Celtics*
1	1	44	38	.537	12.0	105.4	105.6	-0.39	1991	Philadelphia 76ers*
2	2	39	43	.476	17.0	103.1	103.3	-0.43	1991	New York Knicks*
3	3	30	52	.366	26.0	101.4	106.4	-4.84	1991	Washington Bullets
4	4	26	56	.317	30.0	102.9	107.5	-4.53	1991	New Jersey Nets
5	5	24	58	.293	32.0	101.8	107.8	-5.91	1991	Miami Heat
7	7	61	21	.744	—	110.0	101.0	8.57	1991	Chicago Bulls*
8	8	50	32	.610	11.0	100.1	96.8	3.08	1991	Detroit Pistons*
9	9	48	34	.585	13.0	106.4	104.0	2.33	1991	Milwaukee Bucks*
10	10	43	39	.524	18.0	109.8	109.0	0.72	1991	Atlanta Hawks*
11	11	41	41	.500	20.0	111.7	112.1	-0.37	1991	Indiana Pacers*
12	12	33	49	.402	28.0	101.7	104.2	-2.33	1991	Cleveland Cavaliers
13	13	26	56	.317	35.0	102.8	108.0	-4.95	1991	Charlotte Hornets
15	1	55	27	.671	—	107.1	102.6	4.30	1991	San Antonio Spurs*
16	2	54	28	.659	1.0	104.0	100.7	3.18	1991	Utah Jazz*
17	3	52	30	.634	3.0	106.7	103.2	3.27	1991	Houston Rockets*
18	4	31	51	.378	24.0	105.9	109.9	-3.79	1991	Orlando Magic
19	5	29	53	.354	26.0	99.6	103.5	-3.75	1991	Minnesota Timberwolves
20	6	28	54	.341	27.0	99.9	104.5	-4.27	1991	Dallas Mavericks
21	7	20	62	.244	35.0	119.9	130.8	-10.31	1991	Denver Nuggets

شکل ۱۵ – دیتاست player با تیم های ستاره دار

همانطور که مشاهده میکنید مشکل تقسیم در بعضی از ردیف های دیتای ما اکنون از بین رفته است. ام در این دیتا هم همانطور که مشخص است نام تبعضی از تیم‌های ما همچنان دارای علامت ستاره است که باید این مشکل را هم مثل تکنیک‌هایی که قبلا استفاده کردیم، حل کنیم. بنابراین از روش جایگزینی رشته استفاده میکنیم و از ستاره بصورت زیر استفاده میکنیم. چیزی که قرار است جایگزین کنیم، یک جالی خالی است.

```
teams["Team"] = teams["Team"].str.replace("*", "", regex=False)
teams
```

	Unnamed: 0	W	L	W/L%	GB	PS/G	PA/G	SRS	Year	Team
0	0	56	26	.683	—	111.5	105.7	5.22	1991	Boston Celtics
1	1	44	38	.537	12.0	105.4	105.6	-0.39	1991	Philadelphia 76ers
2	2	39	43	.476	17.0	103.1	103.3	-0.43	1991	New York Knicks
3	3	30	52	.366	26.0	101.4	106.4	-4.84	1991	Washington Bullets
4	4	26	56	.317	30.0	102.9	107.5	-4.53	1991	New Jersey Nets
...	...	...	...	...	...	...	...	...	...	...
993	13	44	28	.611	—	117.8	114.8	3.13	2020	Houston Rockets
994	14	43	32	.573	2.5	117.0	112.1	4.87	2020	Dallas Mavericks
995	15	34	39	.466	10.5	112.6	113.7	-0.91	2020	Memphis Grizzlies
996	16	32	39	.451	11.5	114.1	115.2	-0.65	2020	San Antonio Spurs
997	17	30	42	.417	14.0	115.8	117.1	-0.55	2020	New Orleans Pelicans

876 rows × 10 columns

شکل ۱۶ – دیتاست player بعد از پاکسازی ستاره‌ها از نام تیم‌ها

می‌توانیم ببینیم که نام آن تیم‌ها اکنون پاک شده‌اند و دیگر علامت ستاره را ندارند.

مشکل دیگری که داریم نام‌های موجود در این مجموعه داده است.

```
teams["Team"].unique()
```

```
array(['Boston Celtics', 'Philadelphia 76ers', 'New York Knicks',
       'Washington Bullets', 'New Jersey Nets', 'Miami Heat',
       'Chicago Bulls', 'Detroit Pistons', 'Milwaukee Bucks',
       'Atlanta Hawks', 'Indiana Pacers', 'Cleveland Cavaliers',
       'Charlotte Hornets', 'San Antonio Spurs', 'Utah Jazz',
       'Houston Rockets', 'Orlando Magic', 'Minnesota Timberwolves',
       'Dallas Mavericks', 'Denver Nuggets', 'Portland Trail Blazers',
       'Los Angeles Lakers', 'Phoenix Suns', 'Golden State Warriors',
       'Seattle SuperSonics', 'Los Angeles Clippers', 'Sacramento Kings',
       'Toronto Raptors', 'Vancouver Grizzlies', 'Washington Wizards',
       'Memphis Grizzlies', 'New Orleans Hornets', 'Charlotte Bobcats',
       'New Orleans/Oklahoma City Hornets', 'Oklahoma City Thunder',
       'Brooklyn Nets', 'New Orleans Pelicans'], dtype=object)
```

شکل ۱۷ – نام‌تیم‌های منحصربفرد

```
combined["Tm"].unique()
```

```
array(['LAL', 'PHO', 'DAL', 'MIA', 'CLE', 'WSB', 'CHI', 'GSW', 'IND',
       'WAS', 'MIN', 'BOS', 'HOU', 'DEN', 'ORL', 'NOH', 'TOR', 'SAC',
       'CHO', 'POR', 'DET', 'PHI', 'UTA', 'MIL', 'VAN', 'SEA', 'NJN',
       'NOK', 'LAC', 'OKC', 'ATL', 'CHA', 'MEM', 'NYK', 'NOP', 'BRK',
       'SAS', 'CHH'], dtype=object)
```

شکل ۱۸ – نام مخفف تیم‌ها(سه حرفی)

نام تیم‌ها سه حرفی است، اما در مجموعه داده‌های تیم‌های ما، ما اسامی کامل تیم‌ها را داریم که مشکل ساز است. زیرا اگر بخواهیم این دو را ادغام کنیم، هیچ راهی وجود ندارد که چه چیزی را ادغام کنیم. چون نام تیم‌ها یکسان نیست. بنابراین باید راهی برای اضافه کردن نام مستعار به تیم پیدا کنیم. (ستون یا نام کامل تیم را به فریم داده ترکیبی اضافه کنیم). بنابراین برای انجام این‌کار از فایلی به نام nichnames.csv استفاده میکنیم. این فایل بین مخفف و نام کامل تیم نگاشت می شود. بنابراین یک ستون برای مخفف و یک ستون برای نام کامل تیم خواهیم داشت.

	A	B	C	D	E
1	Abbreviati	Name			
2	ATL	Atlanta Hawks			
3	BRK	Brooklyn Nets			
4	BKN	Brooklyn Nets			
5	BOS	Boston Celtics			
6	CHA	Charlotte Bobcats			
7	CHH	Charlotte Hornets			
8	CHO	Charlotte Hornets			
9	CHI	Chicago Bulls			
10	CLE	Cleveland Cavaliers			
11	DAL	Dallas Mavericks			
12	DEN	Denver Nuggets			
13	DET	Detroit Pistons			
14	GSW	Golden State Warriors			
15	HOU	Houston Rockets			
16	IND	Indiana Pacers			
17	LAC	Los Angeles Clippers			
18	LAL	Los Angeles Lakers			
19	MEM	Memphis Grizzlies			
20	MIA	Miami Heat			
21	MIL	Milwaukee Bucks			
22	MIN	Minnesota Timberwolves			
23	NJN	New Jersey Nets			
24	NOH	New Orleans Hornets			
25	NOP	New Orleans Pelicans			
26	NOK	New Orleans/Oklahoma City Hornets			
27	NYK	New York Knicks			
28	OKC	Oklahoma City Thunder			
29	ORL	Orlando Magic			

nicknames

شکل ۱۹ – دیتاست نام مستعار تیم‌ها و نام کامل تیم‌ها

```
nicknames = {}
with open("nicknames.csv") as f:
    lines = f.readlines()
    for line in lines[1:]:
        abbrev,name = line.replace("\n","").split(",")
        nicknames[abbrev] = name
```

```
combined["Team"] = combined["Tm"].map(nicknames)
```

```
combined.head()
```

	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	...	STL	BLK	TOV	PF	PTS	Year	Pts Won	Pts Max	Share	Team
0	A.C. Green	PF	27	LAL	82	21	26.4	3.1	6.6	.476	...	0.7	0.3	1.2	1.4	9.1	1991	0.0	0.0	0.0	Los Angeles Lakers
1	A.C. Green	PF	28	LAL	82	53	35.4	4.7	9.8	.476	...	1.1	0.4	1.4	1.7	13.6	1992	0.0	0.0	0.0	Los Angeles Lakers
2	A.C. Green	PF	29	LAL	82	55	34.4	4.6	8.6	.537	...	1.1	0.5	1.4	1.8	12.8	1993	0.0	0.0	0.0	Los Angeles Lakers
3	A.C. Green	PF	30	PHO	82	55	34.5	5.7	11.3	.502	...	0.9	0.5	1.2	1.7	14.7	1994	0.0	0.0	0.0	Phoenix Suns
4	A.C. Green	SF	31	PHO	82	52	32.8	3.8	7.5	.504	...	0.7	0.4	1.4	1.8	11.2	1995	0.0	0.0	0.0	Phoenix Suns

5 rows × 34 columns

شکل ۲۰ – دیتاست player بعد از پاکسازی و ترکیب نام مخفف تیم‌ها و نام کامل آنها

بنابراین اکنون همه چیزهایی را داریم که برای ادغام چارچوب داده ترکیبی و تیم خود نیاز داریم.

```
train = combined.merge(teams, how="outer",on=["Team", "Year"])
```

```
train
```

	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	...	Share	Team	Unnamed: 0	W	L	W/L%	GB	PS/G	PA/G	SRS
0	A.C. Green	PF	27	LAL	82	21	26.4	3.1	6.6	.476	...	0.0	Los Angeles Lakers	10.0	58	24	.707	5.0	106.3	99.6	6.73
1	Byron Scott	SG	29	LAL	82	82	32.1	6.1	12.8	.477	...	0.0	Los Angeles Lakers	10.0	58	24	.707	5.0	106.3	99.6	6.73
2	Elden Campbell	PF	22	LAL	52	0	7.3	1.1	2.4	.455	...	0.0	Los Angeles Lakers	10.0	58	24	.707	5.0	106.3	99.6	6.73
3	Irving Thomas	PF	25	LAL	26	0	4.2	0.7	1.9	.340	...	0.0	Los Angeles Lakers	10.0	58	24	.707	5.0	106.3	99.6	6.73
4	James Worthy	SF	29	LAL	78	74	38.6	9.2	18.7	.492	...	0.0	Los Angeles Lakers	10.0	58	24	.707	5.0	106.3	99.6	6.73
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
14087	Spencer Hawes	PF	28	MIL	54	1	14.8	2.5	5.1	.484	...	0.0	Milwaukee Bucks	7.0	42	40	.512	9.0	103.6	103.8	-0.45
14088	Steve Novak	PF	33	MIL	8	0	2.8	0.3	0.9	.286	...	0.0	Milwaukee Bucks	7.0	42	40	.512	9.0	103.6	103.8	-0.45
14089	Terrence Jones	PF	25	MIL	54	12	23.5	4.3	9.1	.470	...	0.0	Milwaukee Bucks	7.0	42	40	.512	9.0	103.6	103.8	-0.45
14090	Thon Maker	C	19	MIL	57	34	9.9	1.5	3.2	.459	...	0.0	Milwaukee Bucks	7.0	42	40	.512	9.0	103.6	103.8	-0.45
14091	Tony Snell	SG	25	MIL	80	80	29.2	3.1	6.8	.455	...	0.0	Milwaukee Bucks	7.0	42	40	.512	9.0	103.6	103.8	-0.45

14092 rows × 42 columns

شکل ۲۱ – ترکیب نام تیم‌ها و سال

```
del train["Unnamed: 0"]
```

اکنون ستون‌های اضافی ما از مجموعه داده‌های تیم‌ها را میبینید که به خوبی همه داده‌های ما باهم ادغام شده است.

```
train.dtypes
```

```
Player      object
Pos         object
Age         object
Tm          object
G           object
GS          object
MP          object
FG          object
FGA         object
FG%         object
3P          object
3PA         object
3P%         object
2P          object
2PA         object
2P%         object
eFG%        object
FT          object
FTA         object
FT%         object
ORB         object
DRB         object
TRB         object
AST         object
STL         object
BLK         object
TOV         object
PF          object
PTS         object
Year         int64
```

شکل ۲۱ – نوع دیتاست که از جنس object است

نکته: بسیاری از انواع داده‌های ما، انواع داده‌های object هستند. ولی اکثرا ستون‌های ما عددی هستند. اما پانداس انها را به عنوان رشته ذخیره کرده است. که باید انها را به ستون‌های عددی تبدیل کنیم. و راهی که میتوانیم انجام دهیم استفاده از تابع عددی pd.to_numeric است.

```
train = train.apply(pd.to_numeric, errors='ignore')
```

```
train.dtypes
```

```
Player      object
Pos         object
Age          int64
Tm          object
G            int64
GS           int64
MP         float64
FG         float64
FGA        float64
FG%        float64
3P         float64
3PA        float64
3P%        float64
2P         float64
2PA        float64
2P%        float64
eFG%       float64
FT         float64
FTA        float64
FT%        float64
ORB        float64
DRB        float64
TRB        float64
AST        float64
STL        float64
BLK        float64
TOV        float64
PF         float64
PTS        float64
Year         int64
Pts Won    float64
Pts Max    float64
Share      float64
Team        object
W          float64
```

شکل ۲۲- تغییر دادن نوع دیتاست از object به float64 و int64

سوال دوم:

اگر در مجموعه داده ها نمونه های ستون کلاس نامتوازن باشند، یعنی یکی از کلاس ها دارای majority باشد، پس معیارaccuracy دیگر نمی تواند به خوبی نشانگر دقت مدل باشد و باید از معیار های دیگری مانند presicion و recall و f1score استفاده کرد. همچنین در صورتی که تولید سمپل در دسترس باشد، می توان از این روش استفاده کرد. اما در موضوع دیتاست ما فقط می توان با استفاده از بررسی دیگر معیار های متریک، مدل مناسب را تشخیص داد.

بخش چهارم: مصورسازی و تحلیل EDA

سوال اول:

در حل سوال اول این قسمت، با استفاده از الگوریتم randomforestregressor می خواهیم بررسی کنیم که کدام ستون ها بیشترین نقش را دارند؟ پس ابتدا ستون‌ها را نرمالایز کرده و سپس الگوریتم randomforestregressor را پیاده سازی کردیم. که نتایج بدست امده به شرح زیر است:

	feature	importance
0	Pts Won	0.806346
1	Pts Max	0.050632
8	FG%	0.021626
5	AST	0.021534
9	3P%	0.019718
11	WS	0.018662
10	FT%	0.012775
4	TRB	0.012063
12	WS/48	0.010690
7	BLK	0.007891
3	MP	0.006615
6	STL	0.006090
2	G	0.005359

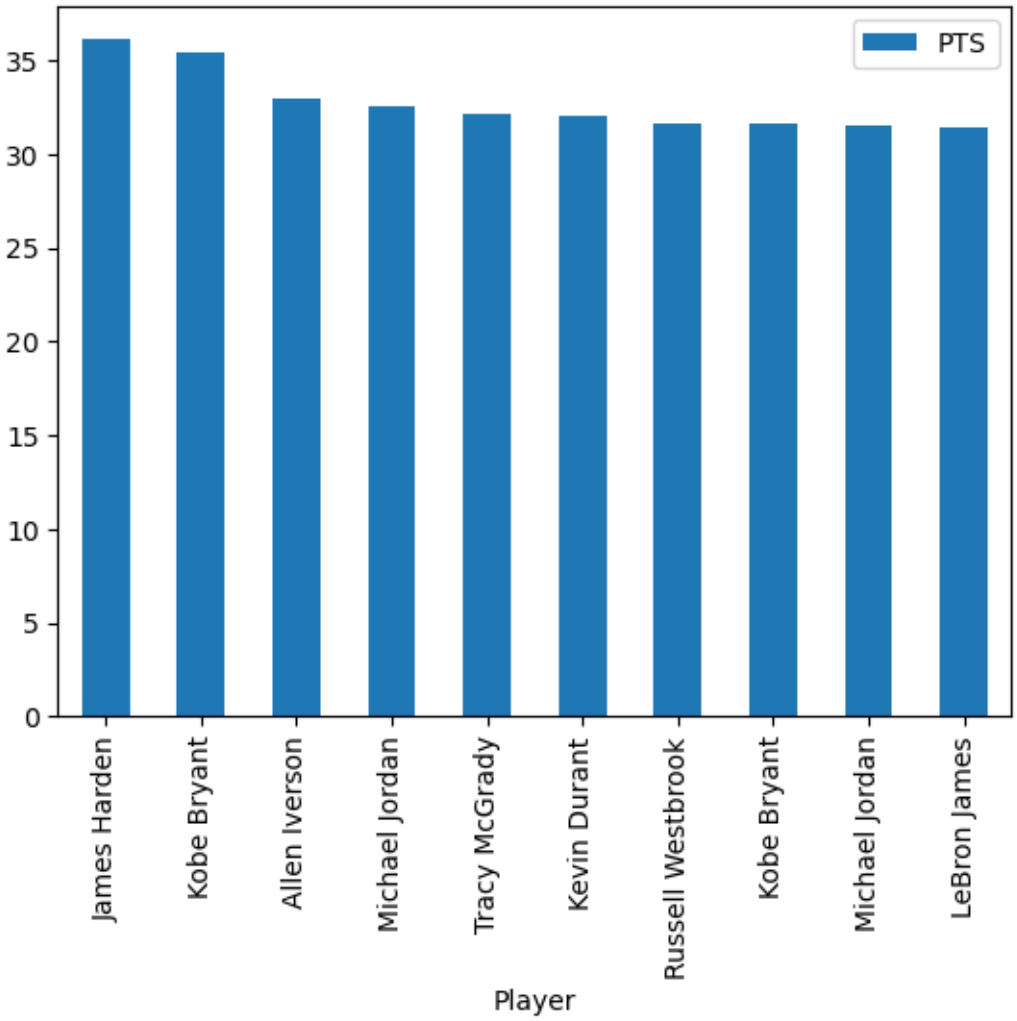
شکل ۲۳ : ستون هایی که بیشترین نقش را دارند

همانطور که می دانستیم، دو ستون اول Pts Won و ستون Pts Max ، بیشترین نقش را دارند. و این نتایج مطابق با انتظارات بود. زیرا این دو ستون بسیار با ستون MVP کوریلیشن دارند. دلیل آن نیز این است که این ستون ها به معنی روش های گرفتن امتیاز در بازی است که مهم ترین عامل را در تعیین MVPمی شازد. که روش حل آن در ادامه آمده است.

سوال دوم:

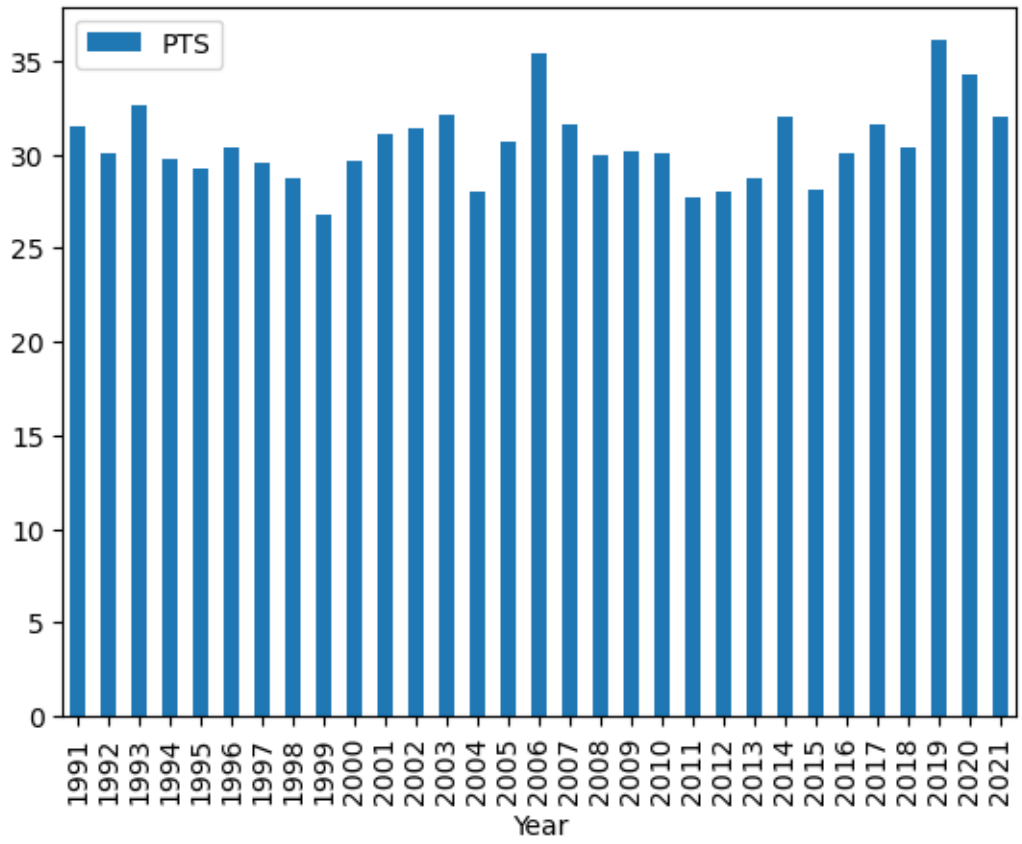
فیچرهای مهم ما هستند که تاثیر مستقیمی در پیش‌بینی ما در یادگیری "Pts Won", "PTS", "Share" ماشین دارند. بنابراین از این فیچرها در پیش‌بینی مدلمان استفاده نخواهیم کرد تا مدل اورفیت نشود.

اولین کاری که میتوانیم انجام دهیم این است که در واقع ببینیم چه کسی بیشترین امتیاز را در کل مجموعه داده کسب کرده است.



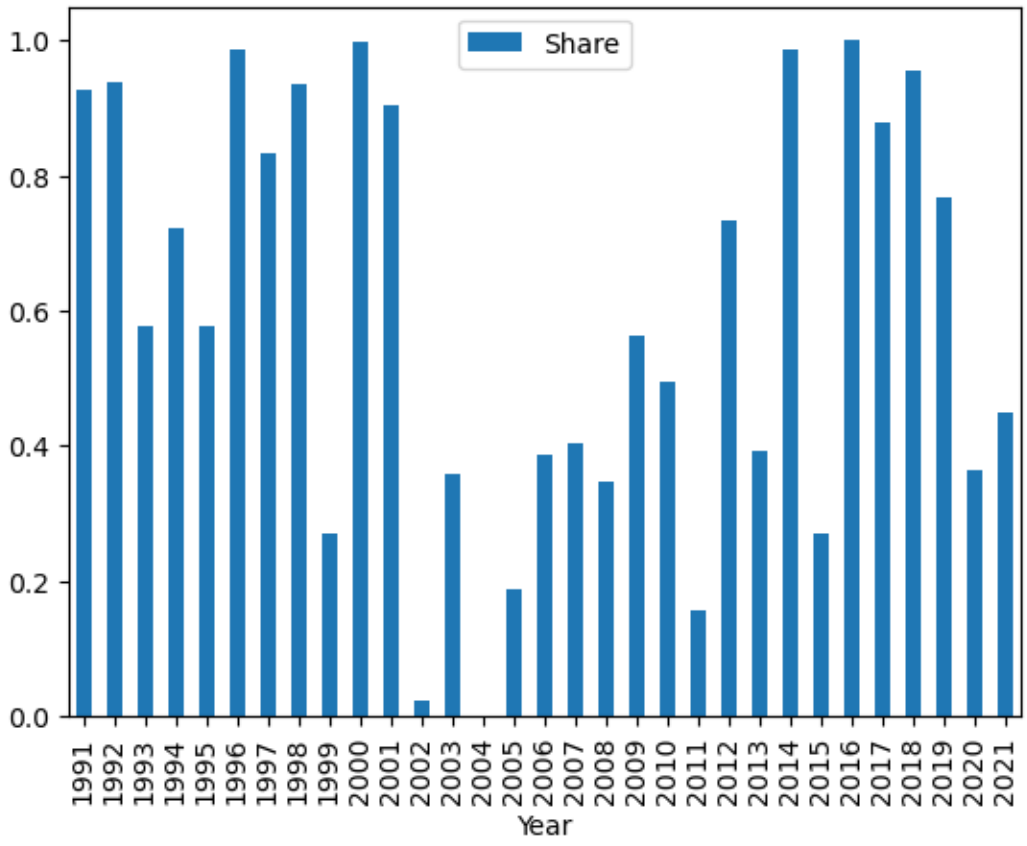
شکل ۲۴ – بیشترین امتیاز کسب شده در کل مجموعه داده

این نمودار نشان میدهد که چه کسی در فصل‌های فردی بالاترین امتیاز را داشته است. که james بالاترین امتیاز را کسب کرده است.



شکل ۲۵ – بالا ترین امتیاز کسب شده در هر بازی

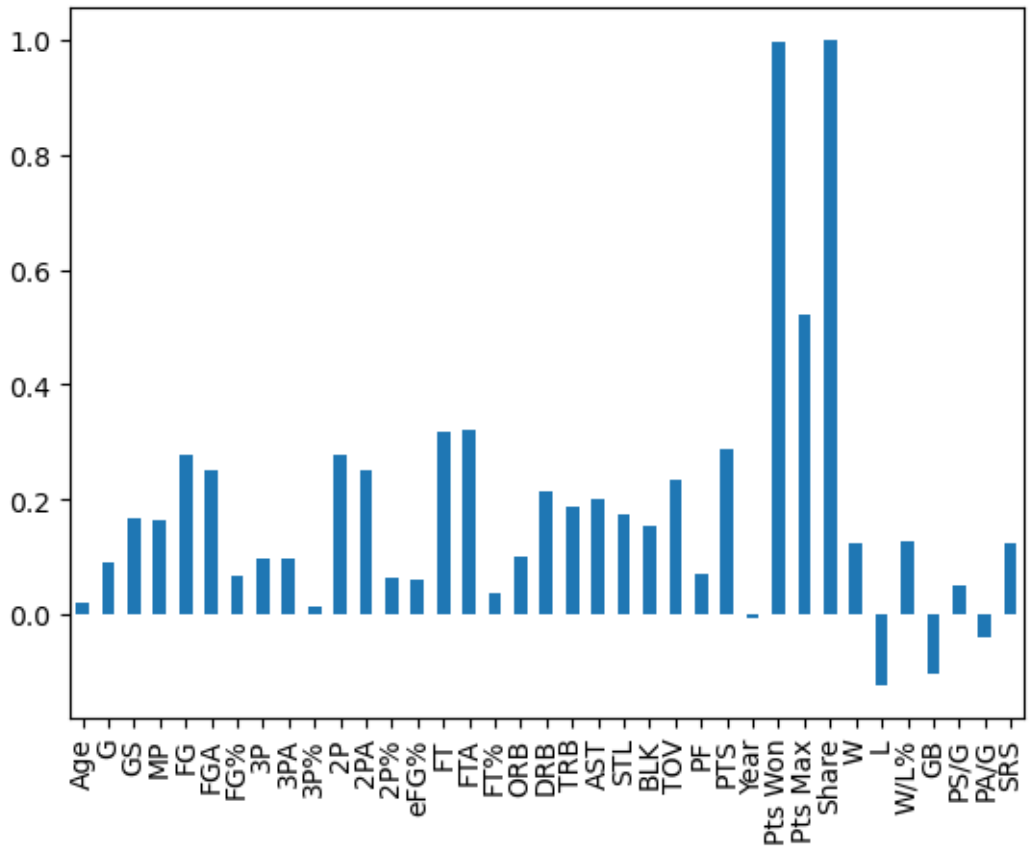
این نمودار بالاترین امتیاز کسب شده در هر بازی را نشان میدهد که چند امتیاز بوده است. همانطور که در داده‌های قبل از اینکه مایکل جردن ۹۱ تا ۹۳ باشد، تغییر نکرده است و میتوانیم ببینیم چند امتیاز بالاترین میانگین گلزن برای هر سال گلزنی در هر بازی در اینجا از سال ۱۹۹۱ تا ۲۰۲۱ بسیار خوب است.



شکل ۲۶ – بیشترین امتیاز مربوط به هر سال

این نمودار نشان میدهد که بیشترین امتیاز مربوط در چه سالی بوده است. که بیشترین امتیاز مربوط به سال ۲۰۱۶ و سال ۲۰۰۰ است با توجه به نمودار بالا.

سپس آخرین تحلیلی که می‌توانیم انجام دهیم این است به همبستگی هایی که ذکر کردیم نگاه کنیم. ما از این داده‌ها برای یادگیری ماشین استفاده خواهی کرد. بنابراین ما از روش همبستگی برای یافتن همبستگی‌ها در برابر این ستون‌ها استفاده می‌کنیم.



شکل ۲۷ – نمودار همبستگی بین داده‌ها

در این نمودار می‌توانیم انهایی را ببینیم که همبستگی بسیار زیادی با هم دارند. فقط آنها متغییرهایی هستند که سعی می‌کنیم انها را پیش بینی کنیم. بنابراین انها را باید نادیده بگیریم.

بخش پنجم: انتخاب ویژگی و کاهش ابعاد

در این بخش از پروژه باید ببینیم که آیا با نحسی ابعاد مواجه هستیم یا خیر. ما زمانی نحسی ابعاد خواهیم داشت که به دلیل وجود تعداد زیادی از ابعاد، نتوانیم یک پترن میان داده های با ابعاد بالای خود پیدا کنیم و این باعث شود که مدل ما دقت پایینی داشته باشد. همگنی که همچین اتفاقی می افتد، ما با روش های کاهش ابعاد می توانیم ابعاد مساله را کاهش دهیم.

در دیتاست این پروژه از انجایی که هرچه تعداد ابعاد بیشتر شود، نیاز به مقدار بیشتری دیتا داریم و دیتا ما در این پروژه ثابت است، ما نحسی ابعاد داریم و باید آن را با روش های نحسی ابعاد کاهش دهیم. دلیلی دیگری که برای نحسی ابعاد در این دیتاست داریم این است که تعداد بالای ابعاد قابل Visualize و بررسی نیست، و این نیز یکی از دلایلی است که ما در این پروژه نحسی ابعاد داریم.

البته کاهش ابعاد در دیتاست نباید بدون توجه به ساختار دیتا باشد. به همین دلیل با توجه به کوریلیتد بودن بعضی از فیچر ها ما در این دیتاست از pca استفاده می کنیم. زیرا PCA کمک می کند که نویز و پیچیدگی در دیتا کاهش یابد و می تواند می تواند باعث شود ما یک lower-dimensional representation با بیشترین حفظ اطلاعات دیتاست داشته باشیم.

```
#Standardize the features
#Create an object of StandardScaler which is present in sklearn.preprocessing
scalar = StandardScaler()
scaled_data = pd.DataFrame(scalar.fit_transform(class_mvp)) #scaling the data
scaled_data

#Applying PCA
pca = PCA(n_components = 5)
pca.fit(scaled_data)
data_pca = pca.transform(scaled_data)
data_pca = pd.DataFrame(data_pca,columns=['PC1','PC2','PC3','PCA4','PCA5'])
```

بخش ششم: روش های طبقه‌بندی

در این بخش می‌خواهیم با الگوریتم های discriminative داده شده، پیشبینی انجام دهیم و بر روی نمونه های جدید پیشبینی انجام دهیم.
همانطور که گفتیم در این تمرین می‌خواهیم با استفاده از داده های سال های قبل، که استخراج کردیم، MVP را در سال بعد پیدا کنیم. حال از آنجایی که داده هایی گه از سال های گذشته داریم پیوسته هستند. ستون هدف ما که ستون share است، داده های ببین ۰ تا ۱ دارد. پس ما یک ستون جدید به نام MVP می سازیم. که در آن در هر سال کسی که بیشترین مقدار را در ستون share داشته باشد، MVP آن برابر ۱ و بقیه در آن سال برابر ۰ می شوند. به این ترتیب مساله را تبدیل به یک مساله classification کردیم که می توانیم از متریک های classification در آن استفاده کنیم..

در این تمرین از ۴ متریک برای سنجش روش‌ها استفاده کردیم که هرکدام به شرح زیر می‌باشند.:

Accuracy : درصدی که مدل ما درست پیشبینی کرده است. اما در مدل‌های یادگیری ماشین،accuracy همیشه بهترین پارامتر برای سنجش نیست.زیرا دید کامل از مدل به ما نمی دهد.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

مثلا در مدلی که کلاس ها imbalance باشند، مانند مدلی که ما در حال انجام یادگیری بر روی آن هستیم، ممکن است مدلaccuracy بالا ولی recall پایینی داشته باشد.
پس معیار های زیر را نیز در مدل خود استخراج خواهیم کرد:

Recall : این معیار دید کامل تری از مدل با محاسبه false negative ها به ما می دهد.

Percision : این معیار دید کامل تری از مدل با محاسبه false Possitive ها به ما می دهد.

F1 score : این عدد ترکیبی از percision و recall می باشد. که در آن میانگین هارمونیک این دو معیار بدست می آید و یک بالانس میان این دو معیار به ما می دهد.

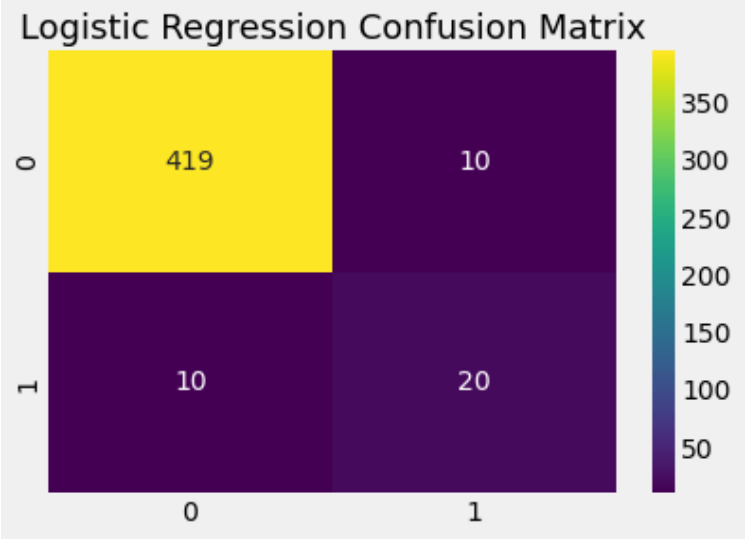
$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

AUC: یکی از مهمترین معیار ها برای سنجش عملکرد مدل، Area Under Curve می باشد

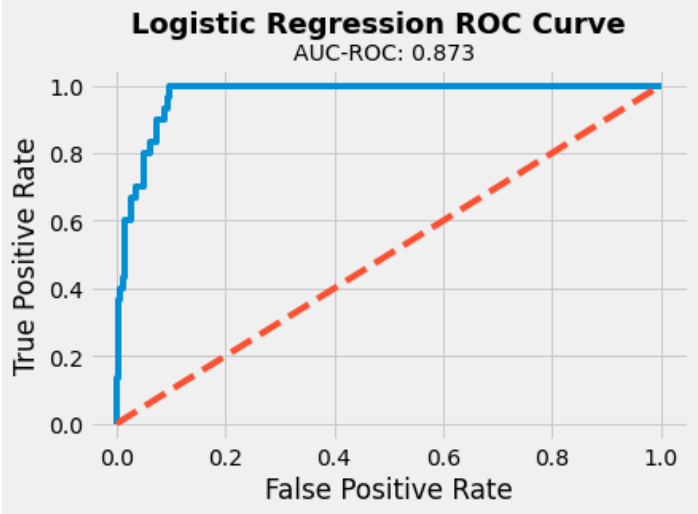
- **روش اول: logistic regression**

پیاده سازی اول ما با روش logistic regression انجام می دهیم. که نتایج آن به شرح زیر می باشد:

```
The metrics:
Accuracy: 0.956
Recall: 0.667
Precision: 0.667
F1: 0.667
Area under ROC curve: 0.972
```



شکل ۲۸ –نمودار confusion matrix برای logistic regression

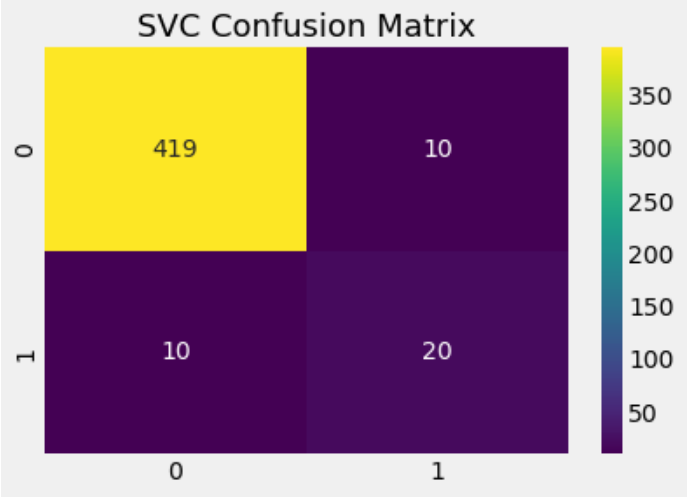


شکل ۲۹ –نمودار ROC curve برای logistic regression

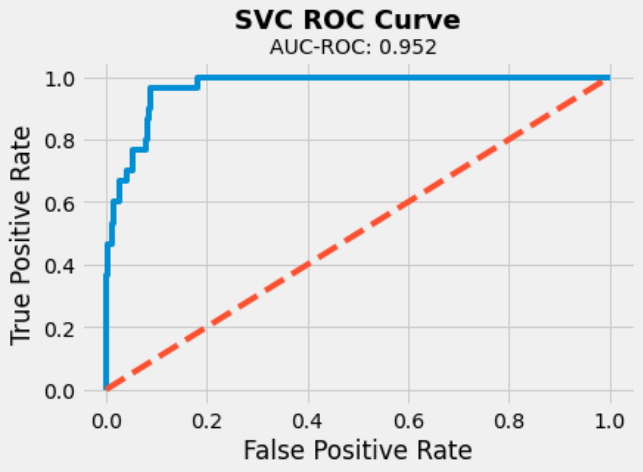
- روش دوم : SVM

پیاده سازی دوم ما با روش SVM می باشد که نتایج آن به شرح زیر است:

```
The metrics:
Accuracy: 0.956
Recall: 0.667
Precision: 0.667
F1: 0.667
Area under ROC curve: 0.968
```



شکل ۳۰ –نمودار confusion matrix برای SVM

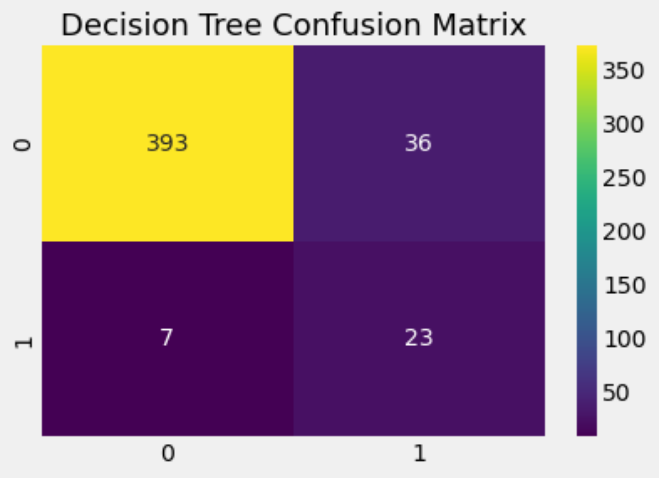


شکل ۳۱ –نمودار ROC curve برای SVM

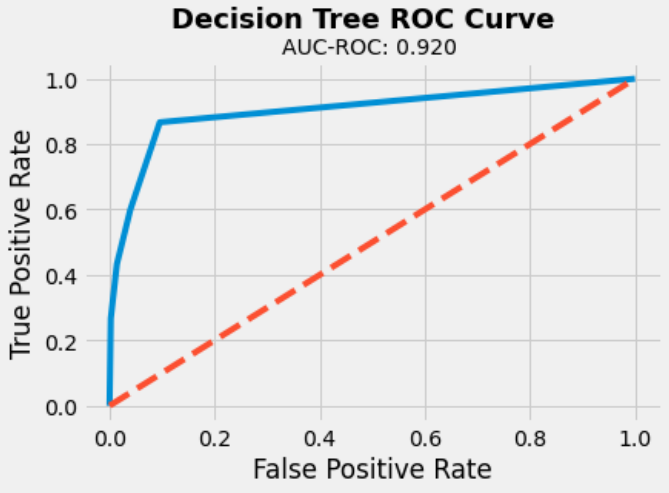
- روش سوم: Decision Tree

پیاده سازی سوم ما با روش decision tree خواهد بود که نتایج آن به شرح زیر می‌باشد:

```
The metrics:
Accuracy: 0.906
Recall: 0.767
Precision: 0.390
F1: 0.517
Area under ROC curve: 0.903
```



شکل ۳۲ –نمودار confusion matrix برای Decision Tree

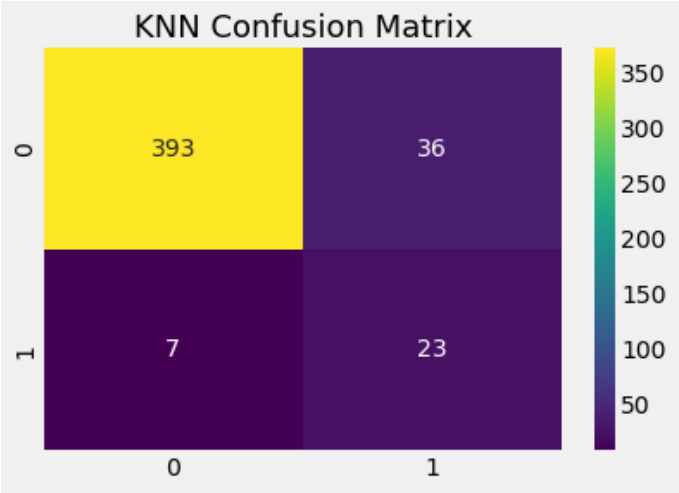


شکل ۳۳ –نمودار ROC curve برای decision tree

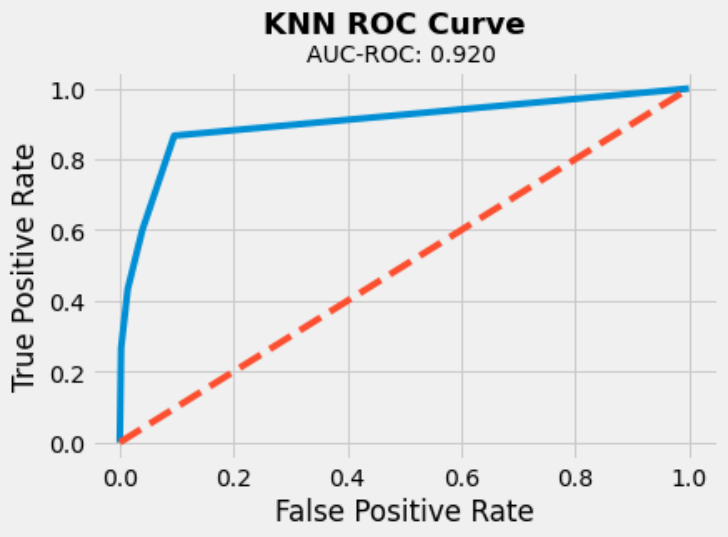
- روش چهارم : KNN

پیاده سازی سوم ما با روش KNNخواهد بود که نتایج آن به شرح زیر می‌باشد:

```
The metrics:
Accuracy: 0.906
Recall: 0.767
Precision: 0.390
F1: 0.517
Area under ROC curve: 0.903
```



شکل 34–نمودار confusion matrix برای KNN

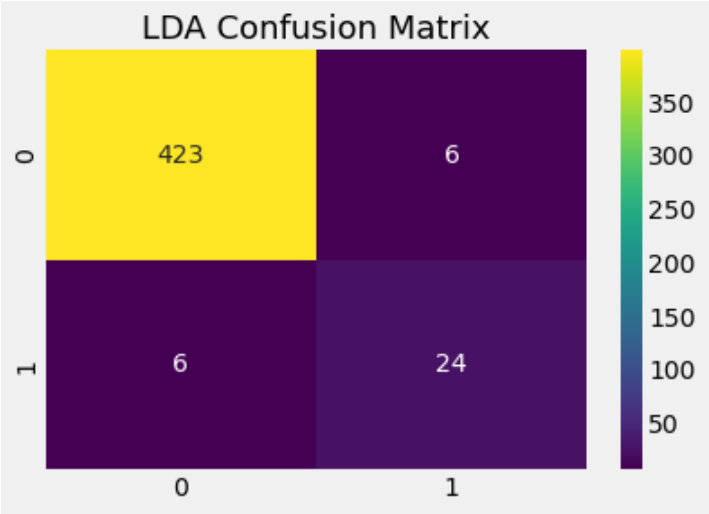


- *شکل 35–نمودار ROC curve برای KNN*

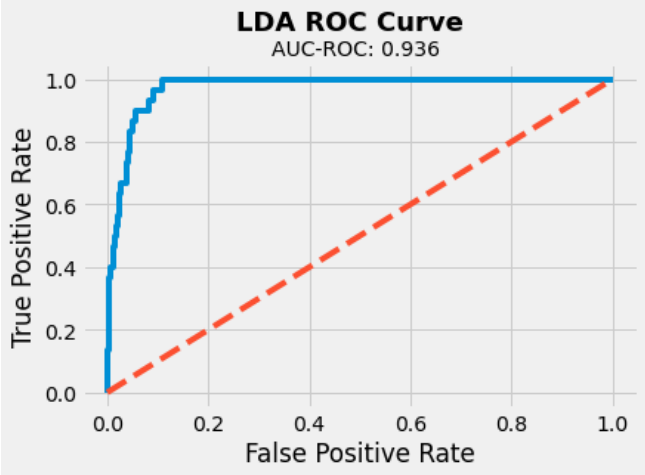
- روش پنجم : LDA

پیاده سازی پنجم ما با روش LDA خواهد بود که نتایج آن به شرح زیر می باشد:

```
The metrics:
Accuracy: 0.974
Recall: 0.800
Precision: 0.800
F1: 0.800
Area under ROC curve: 0.974
```



شکل ۳۶ –نمودار confusion matrix برای LDA

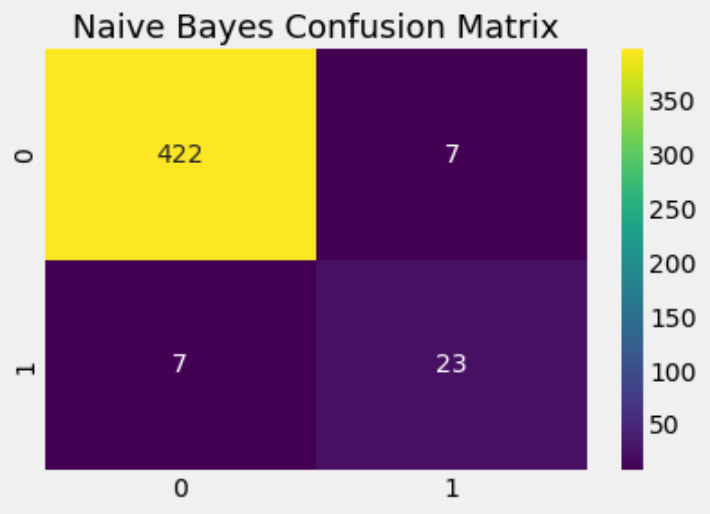


شکل 37 –نمودار ROC curve برای LDA

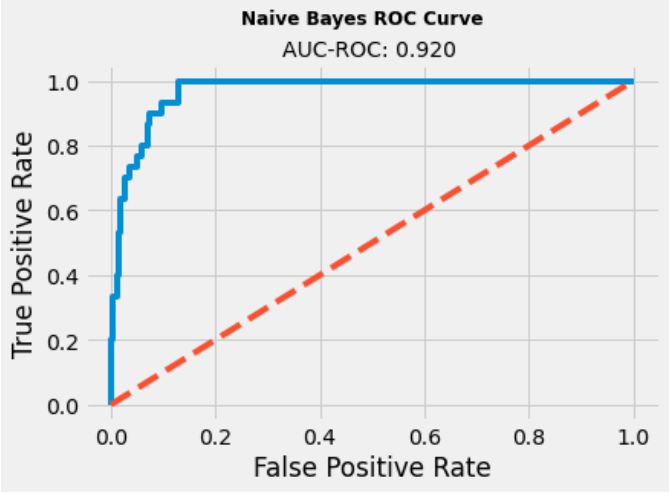
- روش ششم: naïve bayes

پیاده سازی ششم ما با روش naïve bayes خواهد بود که نتایج آن به شرح زیر می باشد:

```
The metrics:
Accuracy: 0.969
Recall: 0.767
Precision: 0.767
F1: 0.767
Area under ROC curve: 0.970
```



شکل ۳۸ –نمودار confusion matrix برای naive bayes



شکل 39 –نمودار ROC curve برای naïve bayes

- مقایسه مدل ها:
در این قسمت می‌خواهیم تمام مدل هایی که پیاده سازی کردیم را از نظر پاذامتر های مختلف مقایسه کنیم و بهترین مدل را انتخاب کنیم.

	accuracy	presicion	recall	F1
Logistic regression	0.95	0.66	0.66	0.66
SVM	0.956	0.667	0.66	0.667
Naïve Bayes	0.969	0.767	0.76	0.767
LDA	0.974	0.80	0.80	0.80
KNN	0.906	0.390	0.767	0.517
Desicion Tree	0.695	0.160	0.867	0.781

همانطور که از مقایسه تمامی متریک های گفته شده می توان فهمید، در 3 متریک accuracy، precision و F1 کلسیفایر LDA بهترین نتیجه را داده است و در recall نیز نتیجه بسیار خوبی گرفته است. پس این مدل را به عنوان بهترین مدل برای دیتاست انتخاب می کنیم.

سوال دوم:

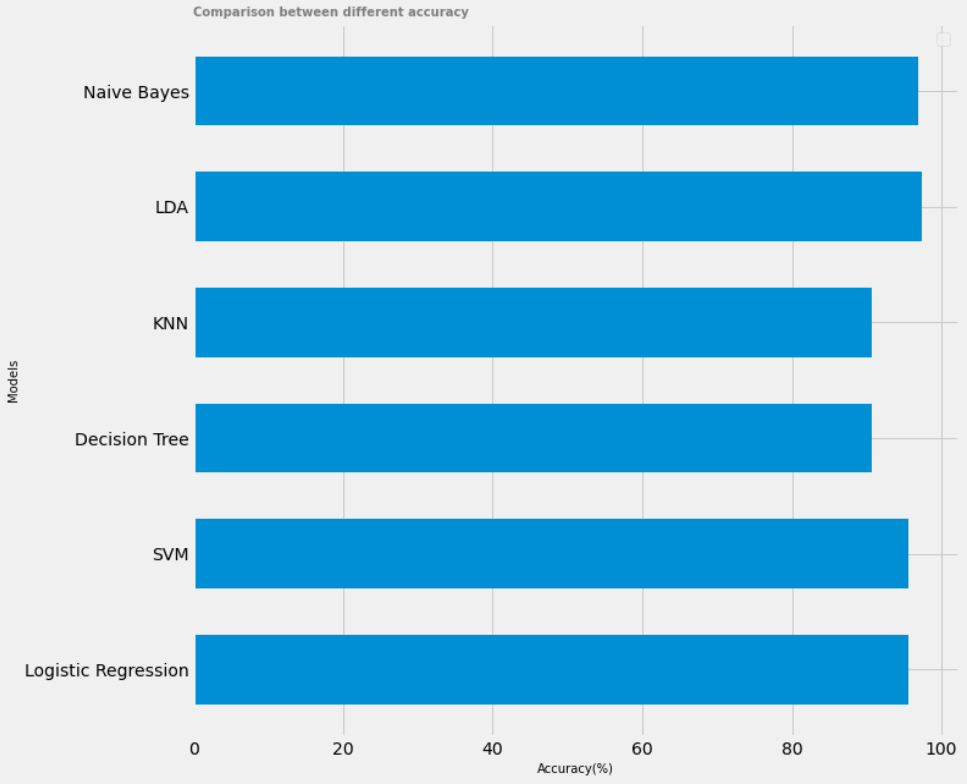
در بسیاری از اوقات می توان از ویژگی های مهم خروجی randomforestregressor استفاده کرد. اما در اینجا، از آنجایی که دیتا ها با هم کوریلیشن بالایی داشتند، و تعداد ویژگی ها بالا بود ما از روش های کاهش ابعاد استفاده کردیم. **حال اگر از مهم ترین ویژگی های این مدل استفاده کنیم، دقت مدل ما به احتمال زیاد بسیار بالا می شود،** اما مدل ما به داده های train ، overfit می شود.

سوال سوم:

نمودار accuracy مدل‌ها به شرح زیر می باشد:

Model	Accuracy
Logistic Regression	95.642702
SVM	95.642702
Decision Tree	90.631808
KNN	90.631808
LDA	97.385621
Naive Bayes	96.949891

همانطور که از نمودار معلوم است، طبقه بند LDA بهترین عملکرد را از نظر معیار accuracy دارد.



شکل 40 –نمودار مقایسه accuracy مدل های استفاده شده

Accuracy در بیشتر موارد به عنوان مهم ترین معیار برای سنجش یک مدل استفاده می شود. اما همیشه استفاده از accuracy بهترین کار نیست. زیر مثلا اگر در مدل ما بین کلاس ها imbalance از نظر تعداد وجود داشته باشد، و فقط از accuracy استفاده کنیم، مدل ما اطلاع دقیقی از کلاس miority به ما نخواهد داد. در این موارد با بررسی معیار های دیگر گفته شده در قسمت بالا (percision, recall, F۱-score, AUC) می توانیم به درک دقیقی از مدل برسیم.

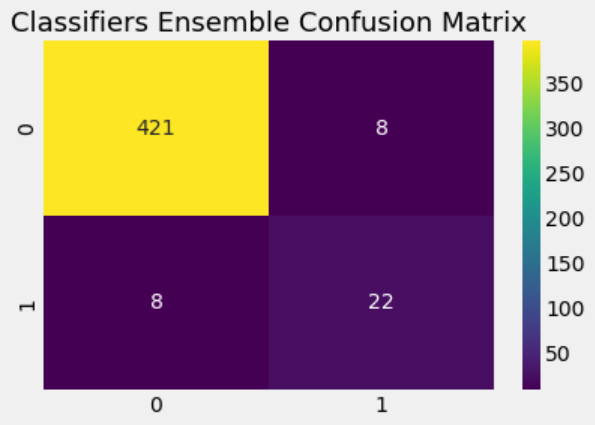
سوال چهارم:

در این قسمت می‌خواهیم یک مدل ensemble بسازیم و دقت آن را گزارش دهیم:

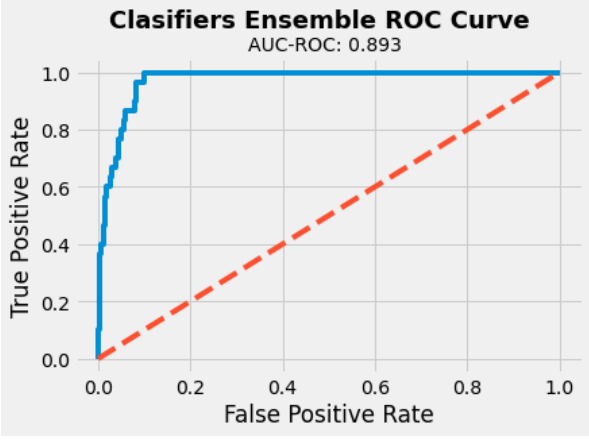
در ساختن مدل ensemble، در level0 از classifier های Bayes، Logistic regression، LDA و SVM استفاده کردیم. و سپس برای level1 از logistic regression استفاده می کنیم.که نتایج پیاده سازی آن به شرح زیر است:

همانطور که می بینیم، این مدل بهترین accuracy را در میان مدل های ما ارائه می کند. و همچنین در بقیه پارامتر ها نیز در رده دوم بهترین مدل قرار می گیرد.

```
The metrics:
Accuracy: 0.965
Recall: 0.733
Precision: 0.733
F1: 0.733
Area under ROC curve: 0.973
```



شکل ۴۱ –نمودار confusion برای ensemble



شکل 42 –نمودار ROC curve برای ensemble