بررسی مدل های مختلف در پیشبینی نتایج بازی‌های Premier League

هدف : بررسی روش‌های مختلف طبقه‌بندی (classification) در پیش‌بینی نتایج بازی‌های لیگ‌برتر انگلیس

روش شناسی:

بررسی متدولوژی های مختلف و تغییر هایپرپارامترها برای رسیدن به بهترین حالت در کلاس‌بندی داده‌ها

کلیدواژه‌ها: کلاس بندی، ماشین بردار پشتیبان، بیز، خوشه‌بندی

مقدمه :

در دهه‌ی گذشته و بخصوص سال‌های اخیر، استفاده از داده‌ها (data) و علم داده در همه‌ی ابعاد زندگیمان مطرح شده است. فوتبال نیز با توجه به پیشرفت تکنولوژی و در دسترس قرار گرفتن پایگاه‌های داده، دچار دگرگونی شده و البته هنوز موج تغییرها در این مورد را در سال‌های پیش رو بیشتر خواهیم دید. این‌روزها، علم داده در فوتبال یکی از شاخه‌های پژوهشی بسیار مورد توجه می‌باشد.

امروزه بحث‌های مربوط به داده‌ها و یادگیری ماشین (machine learning) و همچنین یادگیری عمیق (deep learning)حسابی داغ است. همانطور که گفته شد، در دهه‌ی گذشته بخش زیادی از مشاغل و حوزه‌های سلامت و بهداشت و رسانه و موارد بسیار دیگر تحت تاثیر نفوذ داده‌ها قرار گرفتند.

تصور بر این بود که فوتبال با توجه به ویژگی‌های منحصر به فردش، از این همه‌گیری مصون باشد، موردی که با توجه به نمونه‌های موفق اخیر می‌توان به نادرست بودن آن اذعان کرد. اکنون باشگاه‌های فوتبال با توجه به مزیت‌های زیادی که سرمایه گذاری در تجزیه و تحلیل داده‌ها (data analysis) برایشان به همراه دارد، درحال ورود به این حوزه‌ی بسیار مهم هستند. باشگاه‌هایی که در این مورد تعلل کنند در خطر عقب ماندن از رقیبانشان قرار دارند.

با وجود در دسترس قرار گرفتن حجم زیادی از داده‌ها، مشکل توانایی تفسیر داده‌ها هنوز با ماست. برای مثال اگر باشگاه‌ها حجم زیادی از اعداد را در اختیار داشته باشند اما فاقد دانش برای تفسیر و استخراج اطلاعات عملی آنها باشند، داده‌ها تقریباً بی‌معنی می‌شوند. برای تصمیم‌گیری خوب، تیم‌های فوتبال به داده احتیاج دارند و البته آنها همچنین به تجزیه تحلیل برای درک متقابل آن نیز محتاج هستند. تصویر زیر نمای خوبی از این مفهوم توسط کالج امپریال لندن (Imperial College London)است:

- در ابتدا با داده‌هایی مواجهیم که در نگاه اول فاقد اطلاعات‌خاص و معناداری برای ما هستند.
- با آنالیز کردن و سازماندهی داده‌ها، دیتاهای خود را به اطلاعات و دانش تبدیل می‌کنیم.
- در نهایت، داده‌های اولیه برایمان به یک خِرد، بینش و مفهومی تبدیل می‌شوند که می‌توان از آن در تصمیم گیری‌های مختلف بهره برد.

پیشرفت‌هایی که در پنج سال گذشته در صنعت فوتبال شاهد بودیم در مقابل آنچه در پنج سال آینده رخ خواهد داد ناچیز خواهند بود. در چند سال اخیر، سرعتِ رشد تصاعدی پیشرفت در فناوری‌های پشتیبانی از جمع آوری، ذخیره‌سازی و تجزیه و تحلیل داده‌ها، همزمان با افزایش نمایی سرمایه‌گذاری‌ها در تجزیه و تحلیل‌های ورزشی، کمیت و کیفیت داده‌ها را منفجر کرده است.

با این حال به نظر می‌رسد پیشرفت‌هایی که در پنج سال گذشته شاهد بودیم در مقابل آنچه در پنج سال آینده رخ خواهد داد ناچیز خواهند بود. با بزرگتر شدن و بهبود مجموعه‌ی داده‌ها (datasets) تعداد کاربردهای بالقوه‌ی تجزیه و تحلیل داده‌ها در بازی چند برابر شده است. این امر «تجزیه و تحلیل فوتبال» را به یک مفهوم کاملاً عمومی تبدیل کرده است.

جمع آوری داده :

در این مسئله ما ابتدا با **جمع‌آوری داده‌ها**ی بازی‌های لیگ برتر انگلیس در طول یک فصل کامل، دیتاست مورد نظر را تشکیل داده‌ایم.

رویکرد ما برای جمع آوری داده‌های مورد نیاز، جمع آوری آن از سطح وب (web scraping) بوده که به این منظور از وبسایب fbref استفاده کرده‌ایم.

مسئله‌ی ما یک مسئله‌ی کلاس‌بندی است که ستون تارگت داده‌های ما، Win می‌باشد که شامل مقادیر(۰-۱) برای هر تیم، در یک مسابقه‌است.

در ادامه جهت امکان استفاده از داده‌ها برای مدلسازی، ستون های کتوریکال را عددی کرده‌ایم.

از آنجایی که بعضی فیچرها فقط در پایان بازی قابل دسترسی اند، ما به صورت صریح نمی‌توانیم از آنها قبل از یک بازی برای پیش‌بینی استفاده کنیم.

به همین خاطر ستون هایی با پسوند **rolling** ایجاد کردیم که در واقع میانگین ۵ تا از ستون مشابه در سمپل های قبلی است.

مدلسازی :

ماشین بردار پشتیبانی (Support vector machines - SVMs)یکی از روش‌های یادگیری بانظارت[۱] است که از آن برای طبقه‌بندی[۲] و رگرسیون[۳] استفاده می‌کنند. مبنای کاری دسته‌بندی کننده SVM دسته‌بندی خطی داده‌ها است و در تقسیم خطی داده‌ها سعی می‌کنیم ابرصفحه ای را انتخاب کنیم که حاشیه اطمینان بیشتری داشته باشد. حل معادله پیدا کردن خط بهینه برای داده‌ها به وسیله روش‌های برنامه سازی غبرخطی که روش‌های شناخته شده‌ای در حل مسائل محدودیت‌دار هستند صورت می‌گیرد. قبل از تقسیمِ خطی برای اینکه ماشین بتواند داده‌های با پیچیدگی بالا را دسته‌بندی کند داده‌ها را به وسیله تابعِ phi به فضای با ابعاد خیلی بالاتر[۴] می‌بریم. برای اینکه بتوانیم مسئله ابعاد خیلی بالا را با استفاده از این روش‌ها حل کنیم از قضیه دوگانی لاگرانژ[۵] برای تبدیلِ مسئله مینیمم‌سازی مورد نظر به فرم دوگانی آن که در آن به جای تابع پیچیده phi که ما را به فضایی با ابعاد بالا می‌برد، تابعِ ساده‌تری به نامِ تابع هسته (کرنل) که ضرب برداری تابع phi است ظاهر می‌شود استفاده می‌کنیم. از توابع هسته مختلفی از جمله هسته‌های نمایی، چندجمله‌ای و سیگموید می‌توان استفاده نمود .

بررسی پارامترهای SVM :

تنظیم پارامترها برای الگوریتم ماشین بردار پشتیبان به طور مؤثر باعث بهبود کارایی مدل می‌شود. در ادامه لیست پارامترهای موجود برای SVM در زبان پایتون مورد بررسی قرار گرفته است:

```
sklearn.svm.SVC(C=1.0, kernel='rbf', degree=3, gamma=0.0, coef0=0.0,
shrinking=True, probability=False,tol=0.001, cache_size=200,
class_weight=None, verbose=False, max_iter=-1, random_state=None)
```

کرنل: این پارامتر پیش از این مورد بررسی قرار گرفت. گزینه‌های گوناگونی شامل«linear» ، «rbf»، «poly» برای کرنل وجود دارند و در حالت پیش‌فرض کرنل روی پارامتر rbf قرار دارد .و poly برای خط جداساز غیر راست مفید هستند.

گاما (gamma): ضریب کرنل برای rbf ، poly و sigmoid است. هرچه مقدار گاما بیشتر باشد، الگوریتم تلاش می‌کند برازش را دقیقاً بر اساس مجموعه داده‌های تمرینی انجام دهد و این امر موجب تعمیم یافتن خطا و وقوع مشکل بیش برازش (Over-Fitting) می‌شود.

C: پارامتر جریمه C ، برای جمله خطا است. این پارامتر همچنین برقراری تعادل بین مرزهای تصمیم‌گیری هموار و طبقه‌بندی نقاط داده تمرینی را کنترل می‌کند.

بررسی SVM:

- **مزایا**
  - حاشیه جداسازی برای دسته‌های مختلف کاملاً واضح است.
  - در فضاهای با ابعاد بالاتر کارایی بیشتری دارد.
  - در شرایطی که تعداد ابعاد بیش از تعداد نمونه‌ها باشد نیز کار می‌کند.
  - یک زیر مجموعه از نقاط تمرینی را در تابع تصمیم‌گیری استفاده می‌کند (که به آن‌ها بردارهای پشتیبان گفته می‌شود)، بنابراین در مصرف حافظه نیز به صورت بهینه عمل می‌کند.

- معایب
  - هنگامی که مجموعه داده‌ها بسیار بزرگ باشد، عملکرد خوبی ندارد، زیرا نیازمند زمان آموزش بسیار زیاد است.
  - هنگامی که مجموعه داده نوفه (نویز) زیادی داشته باشد، عملکرد خوبی ندارد و کلاس‌های هدف دچار همپوشانی می‌شوند.
  - ماشین بردار پشتیبان به طور مستقیم تخمین‌های احتمالاتی را فراهم نمی‌کند و این موارد با استفاده از یک اعتبارسنجی متقابل (Cross Validation) پرهزینه پنج‌گانه انجام می‌شوند.

: Naive Bayes classifier

دسته‌بندی‌کننده بیز ساده (به انگلیسی: Naive Bayes classifier) در یادگیری ماشین به گروهی از دسته‌بندی‌کننده‌های ساده بر پایه احتمالات گفته می‌شود که با فرض استقلال متغیرهای تصادفی و براساس قضیه بیز ساخته می‌شوند. به‌طور ساده روش بیز روشی برای دسته‌بندی پدیده‌ها، بر پایه احتمال وقوع یا عدم وقوع یک پدیده‌است .

اگر nمتغیر ورودی داشته باشیم یعنی $x=(x_1, x_2, ..., x_n)$ و خروجی y از یک مجموعه k عضوی باشد، هدف از مدل‌سازی پیدا کردن احتمال مشروط هر کدام از این k دسته است یعنی :

$p(C_k| x_1, x_2, ..., x_n)$.

طبق قانون بیز این احتمال برابر است :

$$p(C_k \mid X) = \frac{p(C_k)\, p(X \mid C_k)}{p(X)}$$

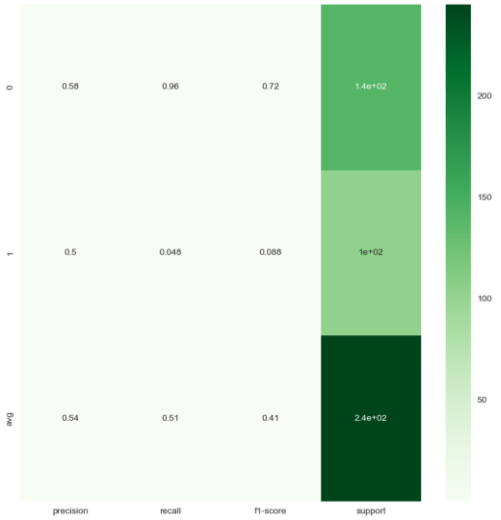
$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

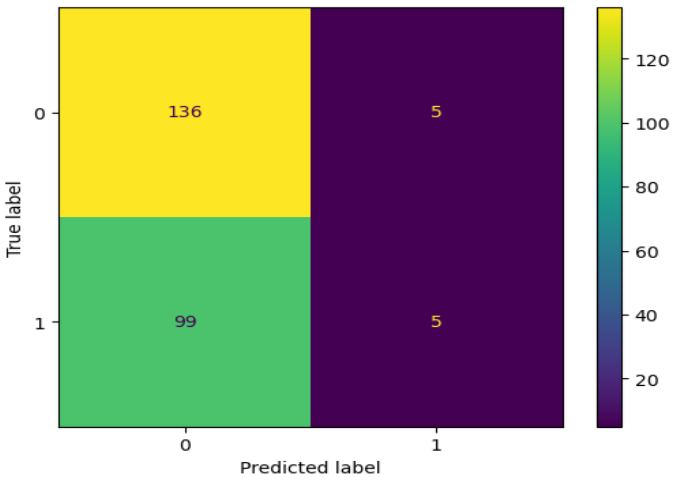
نتایج محاسباتی :

در حالت اولیه‌ی svm، با پارامتر های

```
C=1.0,
kernel='rbf',
degree=3,
gamma='scale',
```

ماتریس درهم‌ریختگی و نتایج زیر، حاصل شده است :





تنظیم پارامترها :

با تغییر مقادیر مختلف در حالت‌های محدود داریم:

۱-

با مجموعه مقادیر :

Kernel : poly

Degree : [1, 2, 3, 4]

حالت best_estimator_، مربوط به پارامتر های kernel : poly با degree:4 است.

که **Accuaracy : 0.57** است.

۲-

Kernel : rbf

gamma : [1, 0.1, 0.01, 0.001, 0.0001]

حالت best_estimator_، مربوط به پارامتر های kernel : rbf با gamma:0.01 است.

که **Accuaracy : 0.56** است.

۳-

kernel : sigmoid

gamma: [1, 0.1, 0.01, 0.001, 0.0001]

حالت best_estimator_، مربوط به پارامتر های kernel : sigmoid با gamma:1 است.

که **Accuaracy : 0.58** است.

همچنین برای بررسی حالت‌های مختلف که به صورت صریح تعریف نکردیم، با رویکرد

RandomizedSearchCv

پارامترهای زیر را بررسی کردیم :

```
{'C': 6.17022004702574, 'gamma': 0.8203244934421581},
{'C': 2.001143748173449, 'gamma': 0.40233257263183975},
{'C': 3.4675589081711307, 'gamma': 0.1923385947687978},
{'C': 3.862602113776709, 'gamma': 0.4455607270430477},
{'C': 5.967674742306699, 'gamma': 0.6388167340033569},
{'C': 6.191945144032948, 'gamma': 0.7852195003967595},
{'C': 4.0445224973151745, 'gamma': 0.9781174363909454},
{'C': 2.2738759319792616, 'gamma': 0.7704675101784022},
{'C': 6.17304802367127, 'gamma': 0.6586898284457516},
{'C': 3.403869385952338, 'gamma': 0.29810148908487877},
{'C': 10.007445686755366, 'gamma': 1.0682615757193976},
{'C': 5.134241781592428, 'gamma': 0.7923226156693141},
{'C': 10.763891522960384, 'gamma': 0.9946066635038473},
{'C': 2.8504421136977793, 'gamma': 0.13905478323288237},
{'C': 3.6983041956456892, 'gamma': 0.9781425034294131},
{'C': 2.9834683383305007, 'gamma': 0.5211076250050521},
{'C': 11.57889530150502, 'gamma': 0.633165284973017},
{'C': 8.918771139504734, 'gamma': 0.4155156310060629},
 {'C': 8.865009276815837, 'gamma': 0.9346256718973729},
{'C': 2.182882773441918, 'gamma': 0.8501443149449674}
```

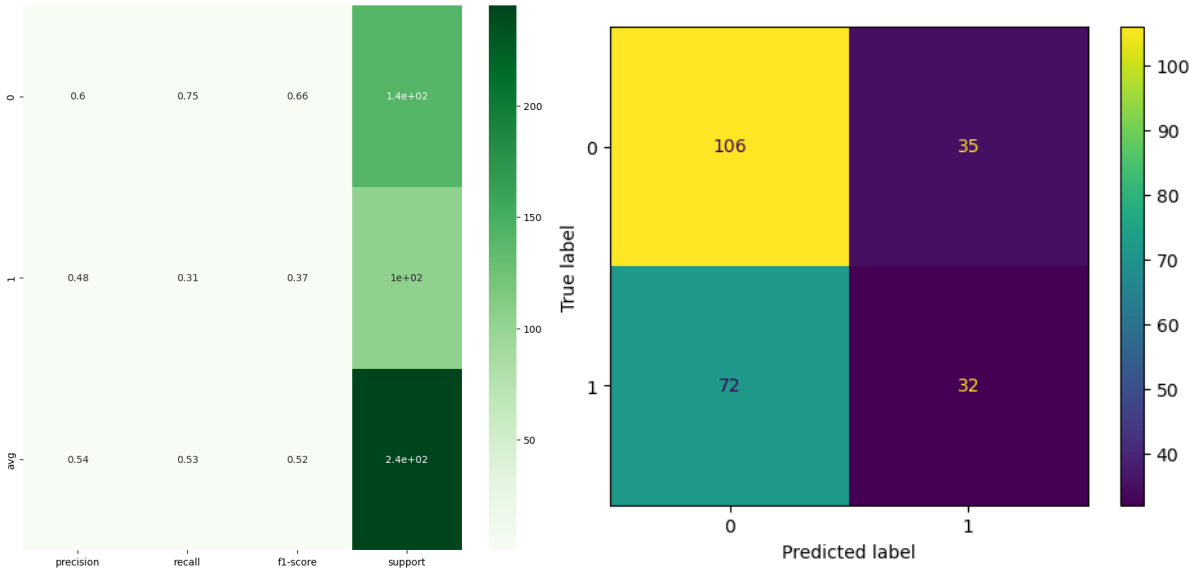
حالت best_estimator_، مربوط به پارامتر های

gamma : 0.13905478323288237 ، kernel : rbf

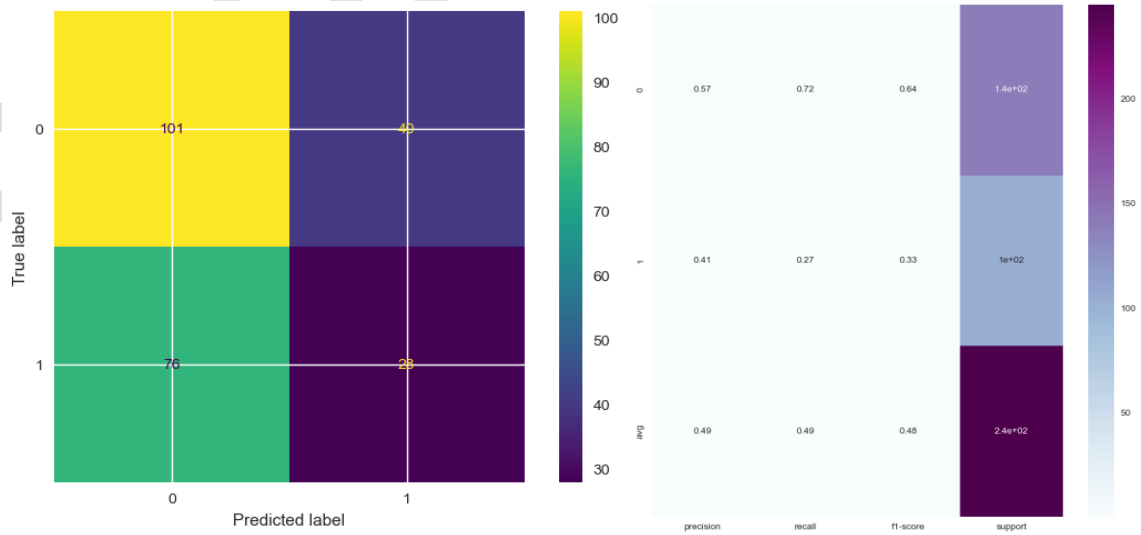
و c : 2.8504421136977793 می‌باشد.

که **Accuaracy : 0.6253** است.

در مجموع، **بهترین نتیجه برای دسته‌بندی با ماشین بردار پشتیبان**، در حالت تنظیم پارامتر با رویکرد RandomizedSearchCv بدست آمده که نتایج آن به‌صورت زیر است :



با مدل **دسته‌بندی‌کننده بیز ساده** هم نتایج بصورت زیر حاصل شد که نسبت به **ماشین بردار پشتیبان**، نتایج ضعیف‌تری برای این مسئله بدست آمده است :



که میزان **accuracy**، **0.526** است.

خوشه‌بندی با KMEANS :

الگوریتم K-Means یک الگوریتم بر پایه ی تکراری است که سعی می‌کند مجموعه داده ها را به زیرگروه‌های متمایز بدون همپوشانی تعریف کند که به این زیر گروه ها خوشه گفته می‌شود؛ که در این گروه ها هر نقطه داده فقط به یک گروه تعلق دارد. در این الگوریتم سعی می‌شود نقاط داده درون خوشه ای را تا حد ممکن شبیه به هم ساخت و در عین حال خوشه ها بیشترین فاصله را از هم داشته باشند.

این الگوریتم داده ها را به یک خوشه اختصاص می‌دهد به طوری که مجموع فاصله مربع شده بین نقاط داده و مرکز گروه (میانگین محاسبه تمام نقاط داده ای که به آن خوشه تعلق دارند) در حداقل باشد. هرچه تنوع کمتری در خوشه ها داشته باشیم، نقاط داده در یک خوشه همگن (مشابه) هستند.

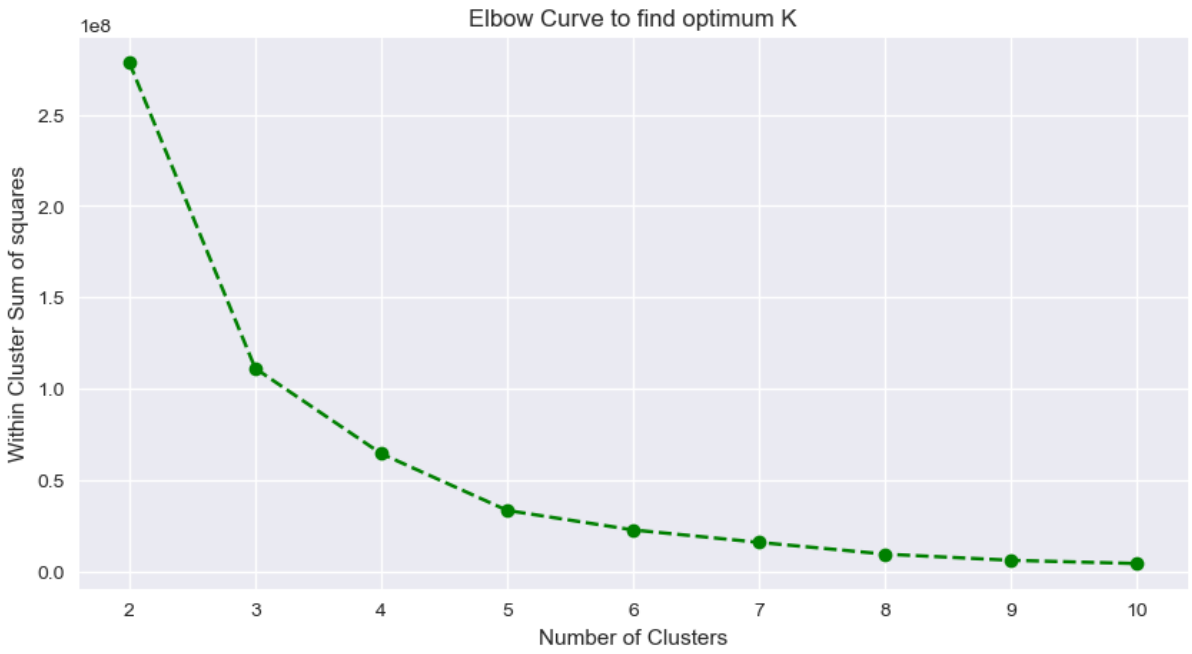
الگوریتم K-Means ، مجموعه داده بدون برچسب را به عنوان ورودی می‌گیرد، مجموعه داده را به تعداد k خوشه تقسیم می‌کند و روند را تکرار می‌کند تا زمانی که بهترین خوشه ها را پیدا نکند الگوریتم ادامه پیدا می‌کند. مقدار k باید در این الگوریتم از پیش تعیین شده باشد. عملکرد الگوریتم خوشه بندی k-mean به صورت زیر است:

با استفاده از یک فرایند تکرار، بهترین مقدار را برای نقاط مرکز تعیین می‌کند. هر نقطه داده را به نزدیکترین مرکز k خود اختصاص می‌دهد. آن نقاط داده ای که نزدیک مرکز k هستند، خوشه ای را ایجاد می‌کنند. از این رو هر خوشه دارای نقاط داده با برخی نقاط مشترک است و از خوشه های دیگر دور است.

در این مسئله با توجه به داده‌های اولیه، در صدد **خوشه‌بندی تیم ها بر اساس عملکرد** کلیشان هستیم.

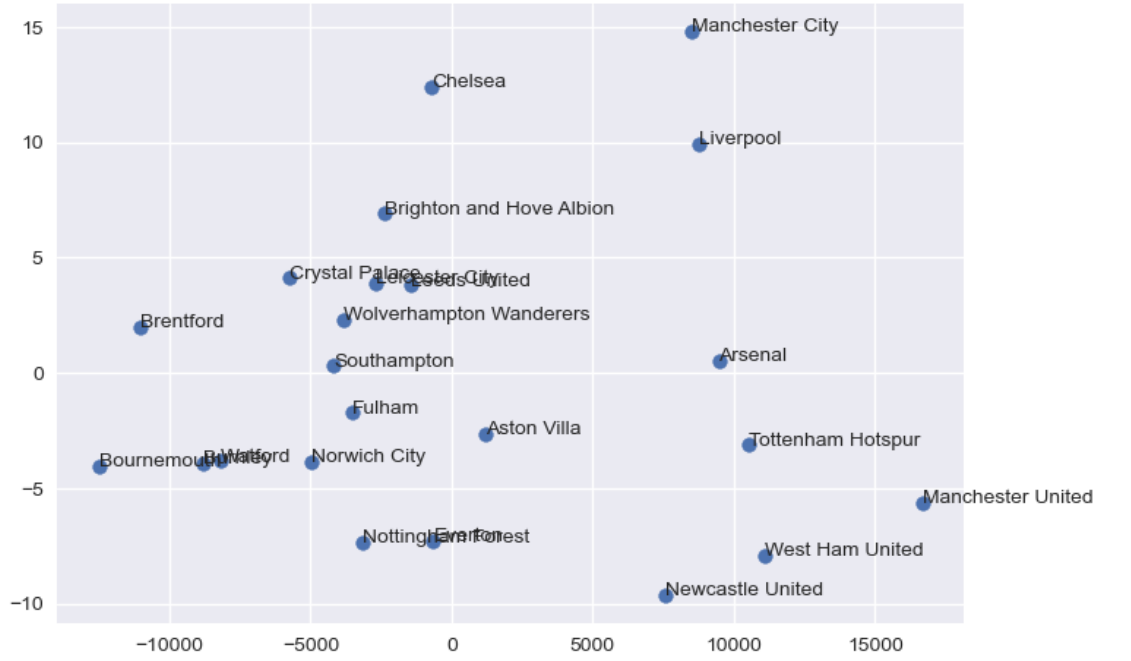
از آنجایی که در KMEANS ، **تعداد خوشه‌ها** باید از قبل مشخص شود، ابتدا با روش elbow، به دنبال پیدا کردن تعداد خوشه‌های مورد نظر هستیم و پس از آن خوشه بندی را انجام می‌دهیم.

بررسی تعداد خوشه‌ها با روش آرنج:

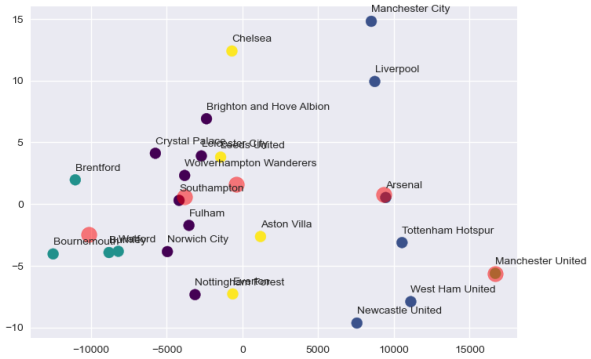


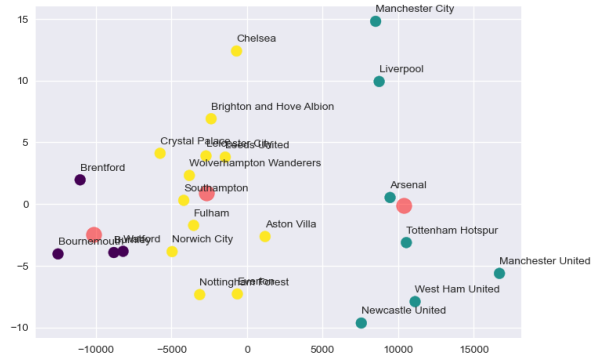
با توجه به نگاه ما به مسئله، انتخاب ۳ یا ۵ میتواند موردنظر باشد.

که هر دو حالت مورد بررسی قرار گرفته‌است:


Manchester City
Chelsea
Liverpool
Brighton and Hove Albion
Crystal Palace
Leicester City
Leeds United
Brentford
Wolverhampton Wanderers
Southampton
Arsenal
Fulham
Aston Villa
Tottenham Hotspur
Bournemouth
Watford
Norwich City
Manchester United
Nottingham Forest
West Ham United
Newcastle United
15
10
5
0
−5
−10
−10000
−5000
0
5000
10000
15000

...


Manchester City
Chelsea
Liverpool
Brighton and Hove Albion
Brentford
Wolverhampton Wanderers
Southampton
Arsenal
Fulham
Aston Villa
Tottenham Hotspur
Norwich City
Manchester United
West Ham United
Newcastle United


Manchester City
Chelsea
Liverpool
Brighton and Hove Albion
Brentford
Wolverhampton Wanderers
Southampton
Arsenal
Fulham
Aston Villa
Tottenham Hotspur
Norwich City
Manchester United
West Ham United
Newcastle United

نتیجه‌گیری:

به طور کلی، استفاده از داده‌ها و نگاه داده‌محورانه به مسئله، همواره می‌تواند همواره می‌تواند دانشی را در پی داشته باشد که در نگاه اولیه قابل مشاهده نیست.

همچنین هر مسئله می‌تواند از دیدگاه های مختلف مورد بررسی قرار گیرد و با جمع‌بندی آن، اطلاعات بسیار ارزشمندی پیرامونش استخراج شود.

همانگونه که مشاهده شد، مدل‌های مورد نظر که از آنها در مسائل مختلف (در اینجا کلاس‌بندی) استفاده میکنیم، دارای هایپر پارامترهایی هستند که تنظیم آنها می‌تواند در رسیدن به‌دقت‌های بالاتر در حل مسائل موردنظر منجر شود و نکته‌ایست که باید همواره مورد بررسی قرار گیرد.

پس بهتراست در استفاده از یک مدل، هایپر پارامترها و مقادیر مختلف آن بررسی شود.

منابع رفرنس داده شده یا مورد استفاده :

- Data : https://fbref.com/en/
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems(Book by Aurélien Géron)
- https://github.com/ageron/handson-ml3
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
- https://scikit-learn.org : 1.4. Support Vector Machines