به نام خدا

گزارش فاز سوم

پروژه هوش محاسباتی

تینا توکلی-محمدهادی امینی

دیتاستی داریم که شامل 102 کلاس مختلف با بردارهای 512 بعدی است، در این فاز پروژه قصد داریم دیتای اموزش را با الگوریتم k_means با k برابر 50 دسته بندی کنیم تا به صورت مناسب طبقه بندی شوند(classification) تا داده جدید را لیبل گذاری کنیم.

بخش اول :

دیتای اموزش را تقسیم میکنیم تا چند کلاستر کوچکتر داشته باشیم که دیتاهای شبیه به هم را در خود دارند ،که اینجا k ما برابر 50 است یعنی ما 102 کلاس مختلف از دیتاها را به 50 کلاستر تقسیم میکنیم به طوریکه مراکز کلاسترها را پیدا کرده و سپس با توجه به فاصله داده ها از مراکز ،کلاستر بندی میکنیم.

ما برای visualize کردن کلاسترها در ابتدا نیاز به کاهش بعد داریم بنابراین ازدو روش زیر استفاده می کنیم(pca وtsne)

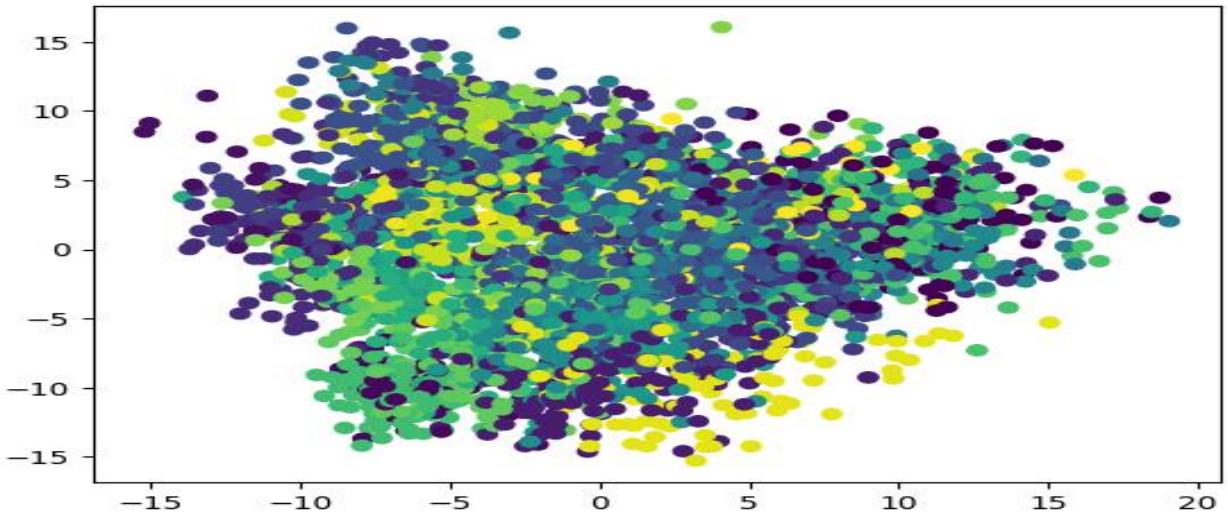
Pcaچیست؟

الگوریتمی که برای کاهش ابعاد داده‌ها و تبدیل آنها به مجموعه‌ای از مؤلفه‌های اصلی با ویژگی‌های مهم استفاده می‌شود. این مؤلفه‌های اصلی به گونه‌ای انتخاب می‌شوند که بیشترین واریانس داده‌ها را حفظ کنند، و بر این PCA اساس، ابعاد داده کاهش می‌یابد

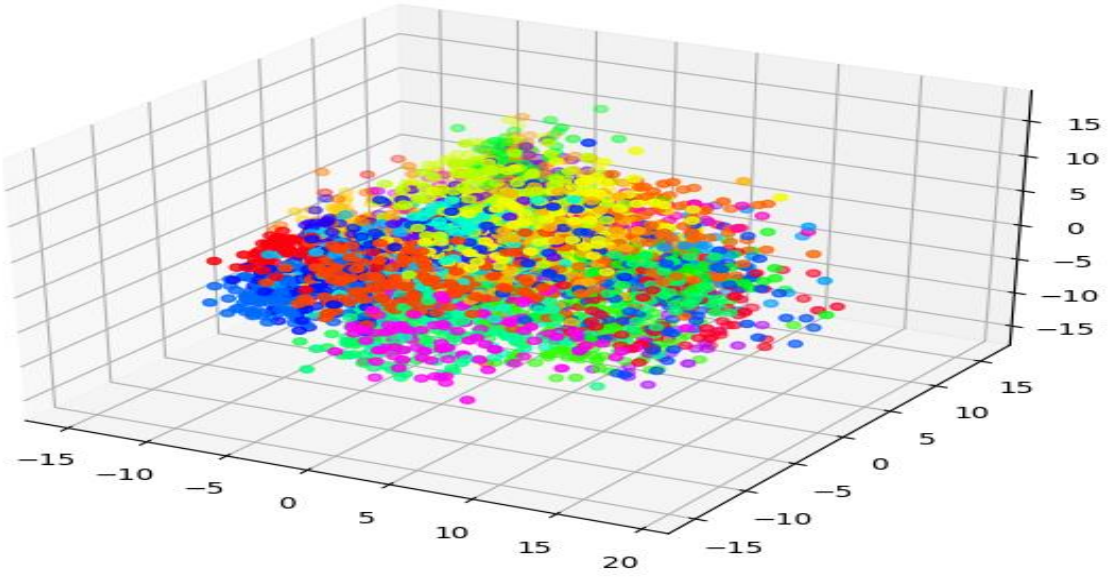
TSNEچیست؟

یک الگوریتم دیگر برای کاهش ابعاد است که برای مصورسازی داده‌های پیچیده در ابعاد بالا استفاده می‌شود. این الگوریتم با انتقال نقاط مشابه به نزدیک یکدیگر و نقاط مختلف به دور از یکدیگر، یک نقشه دوبعدی ایجاد t-SNE نسبت به فضاهای کوچکتر، می‌کند که نمایانگر ساختار داده اصلی در فضای بالاست. مهمترین ویژگی حفظ تفاوت‌ها و روابط نقاط است که به عنوان یک ویژگی مفید در تحلیل داده‌ها محسوب می‌شود.

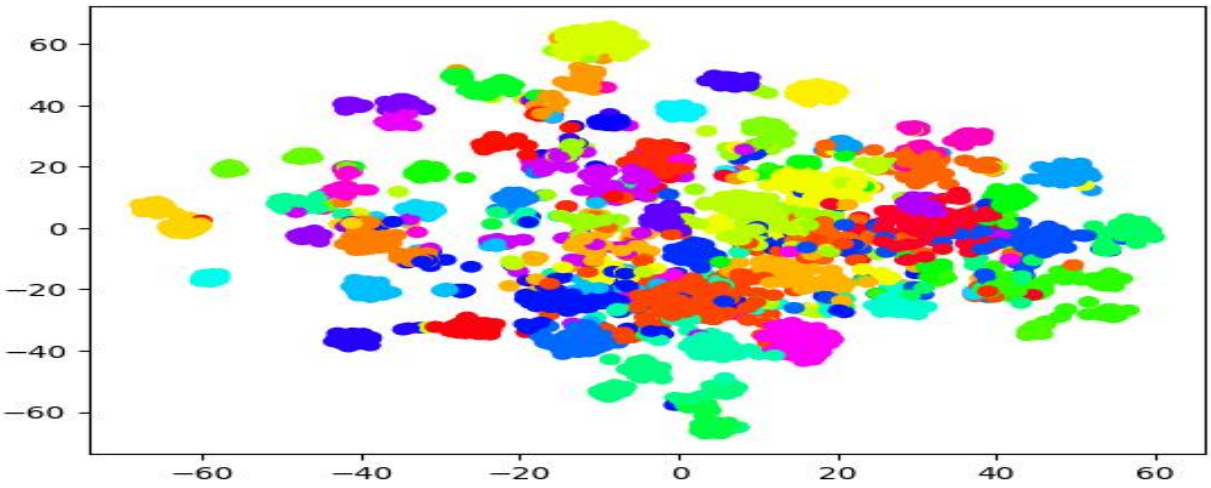
نمایش نمودارها همراه با کاهش ابعاد بعد از کلاستربندی:



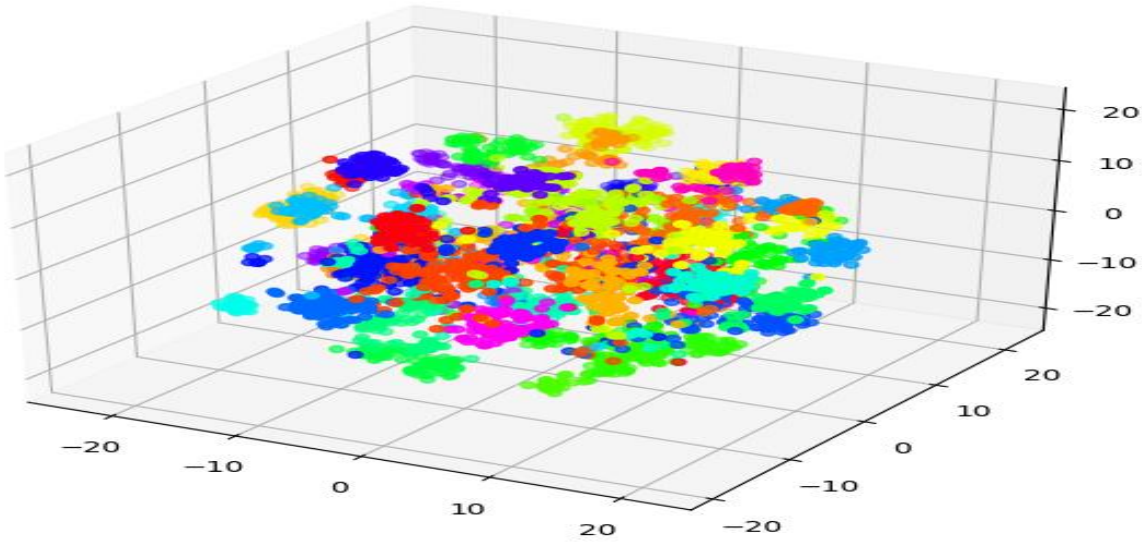
pca(n=2)



pca(n=3)



TSNE(n=2)



TSNE(n=3)

بخش دوم:

بعد از اموزش دادن دیتاست ،وقتی داده تست را میدهیم با استفاده از الگوریتم KNN فاصله ان را با مراکز کلاسترها محاسبه میکنیم ،بنابراین الان فاصله داده تست را با 50 کلاستر داریم سپس انتخاب میکنیم که چه تعداد از نزدیکترین کلاسترها را میخواهیم با توجه به اینکه زمان و امتیاز مهم است (یعنی در 2 مرحله KNN متوالی داریم که اولی روی مرکز کلاسترها و دومی روی دیتاپوینتهای انتخابی است)

حال از بین تعداد کلاسترهای نزدیکی که درنظر گرفتیم انتخاب میکنیم که چه تعداد از نزدیکترین دیتاپوینتها را در نظر بگیریم تا بیشترین امتیاز را بدهد یعنی بهترین حالتی که بیشترین لیبل گذاری درست را دارد.

معیار انتخاب هر یک از موارد فوق score بوده که توسط فرمول(Accuracy+(n*p محاسبه می‌شود.

n تعداد کلاستر های نزدیک انتخابی (بهترین k برای KNN اولیه) بوده و Accuracy میزان دقت تشخیص نهایی متناسب با لیبل واقعی داده های آموزشی ما می‌باشد.

منظور از pهمان پنالتی است که مارا از انتخاب تعداد بالاتری از n منع میکند زیرا اگر توزیع داده مناسبی داشته باشیم با افزایش n امکان تشخیص دقیق تری برای لیبل وجود دارد اما سربار زیادی برای ما دارد که با این روش تنبیه برای آن لحاظ میشود و زمان زیادی را تلف میکند (به صورت پیش فرض 0.2- درنظر گرفته می‌شود.)

برای این منظور ما خروجی امتیازهای خود را با توجه به n انتخابی الگوریتم های KNN خود حساب میکنیم وهمینطور پلات کرده ایم.

فاصله منهتن(Manhattan Distance):

این فاصله را با جمع کردن تفاوت مطلق بین مختصات مربوطه محاسبه می کند. نسبت به فاصله اقلیدوسی نسبت به نقاط پرت حساسیت کمتری دارد و برای مجموعه داده هایی با بی نظمی مناسب است.

محاسبه امتیاز براساس الگوریتم KNN با فاصله منهتن([n,k,score]):

```
[(4, 5, 83.86422466422466),
(5, 5, 83.73748473748473),
(6, 5, 83.7084249084249),
(7, 5, 83.67936507936507),
(5, 9, 83.66422466422466),
(4, 8, 83.62002442002442),
(5, 7, 83.61538461538461),
(5, 8, 83.61538461538461),
(6, 9, 83.6107448107448),
(6, 8, 83.58632478632478),
(7, 8, 83.58168498168497),
(5, 6, 83.56654456654456),
(4, 6, 83.54676434676435),
(6, 7, 83.51306471306471),
(7, 6, 83.5084249084249),
(7, 9, 83.5084249084249),
(8, 7, 83.5037851037851),
```

(8, 7, 83.5037851037851),
(4, 7, 83.4979242979243),
(4, 4, 83.4979242979243),
(5, 4, 83.46886446886447),
(7, 7, 83.45958485958485),
(8, 8, 83.45494505494506),
(4, 9, 83.44908424908425),
(7, 5, 83.43516483516483),
(8, 5, 83.40610500610501),
(3, 5, 83.40488400488401),
(9, 7, 83.37704517704518),
(6, 6, 83.36654456654456),
(8, 9, 83.35726495726496),
(6, 4, 83.34212454212454),
(7, 4, 83.33748473748473),
(8, 7, 83.33284493284494),
(9, 8, 83.32820512820513),
(7, 9, 83.3130647130647),
(3, 7, 83.30720390720391),
(9, 7, 83.3037851037851),
(7, 8, 83.28864468864468),
(8, 5, 83.28400488400489),
(3, 6, 83.28278388278389),
(3, 4, 83.28278388278389),
(9, 5, 83.27936507936508),
(7, 6, 83.26422466422466),
(8, 6, 83.25958485958486),
(9, 9, 83.23052503052503),
(9, 9, 83.23052503052503),
(9, 9, 83.23052503052503),
(7, 4, 83.21538461538461),
(9, 5, 83.206105006105),
(9, 6, 83.206105006105),
(9, 6, 83.206105006105),
(9, 6, 83.15726495726496),
(8, 6, 83.13748473748474),
(8, 9, 83.13748473748474),
(8, 4, 83.06422466422467),
(9, 9, 83.05958485958486),
(8, 8, 83.01538461538462),
(8, 4, 82.9909645909646),
(3, 8, 82.96532356532357),
(9, 4, 82.96190476190476),
(9, 4, 82.96190476190476),
(4, 3, 82.96068376068376),
(9, 4, 82.88864468864469),

(9, 8, 82.88864468864469),
(5, 3, 82.88278388278388),
(3, 3, 82.81880341880343),
(3, 9, 82.7943833943834),
(6, 3, 82.78046398046398),
(5, 1, 82.76068376068376),
(7, 3, 82.75140415140415),
(3, 1, 82.72112332112333),
(4, 1, 82.71648351648352),
(7, 3, 82.55604395604395),
(6, 1, 82.48742368742369),
(8, 3, 82.45372405372406),
(3, 9, 82.42808302808304),
(7, 1, 82.4095238095238),
(2, 5, 82.40830280830279),
(9, 3, 82.32698412698413),
(9, 3, 82.32698412698413),
(2, 1, 82.28620268620267),
(8, 3, 82.25836385836386),
(8, 1, 82.23394383394384),
(2, 4, 82.2129426129426),
(9, 3, 82.20488400488401),
(9, 1, 82.05836385836386),
(7, 1, 81.96996336996337),
(2, 7, 81.96874236874235),
(2, 6, 81.96874236874235),
(2, 8, 81.79780219780218),
(8, 1, 81.7943833943834),
(2, 3, 81.77338217338216),
(9, 1, 81.66764346764347),
(2, 9, 81.45592185592184),
(3, 2, 81.35360195360195),
(5, 2, 81.27106227106226),
(4, 2, 81.25128205128205),
(6, 2, 81.07106227106226),
(7, 2, 81.04200244200243),
(2, 2, 80.77216117216116),
(8, 2, 80.76874236874237),
(9, 2, 80.59316239316239),
(1, 1, 80.4105006105006),
(1, 4, 79.48253968253968),
(1, 3, 79.40927960927961),
(1, 4, 79.36043956043956),
(1, 2, 79.01855921855922),
(1, 5, 78.92087912087912),
(1, 6, 78.7010989010989),

```
(1, 8, 78.0905982905983),
(1, 9, 77.87081807081807])
```

(محاسبه کردن امتیاز براساس تعداد کلاسترهای مختلف و تعداد دیتاپوینت های متفاوت)

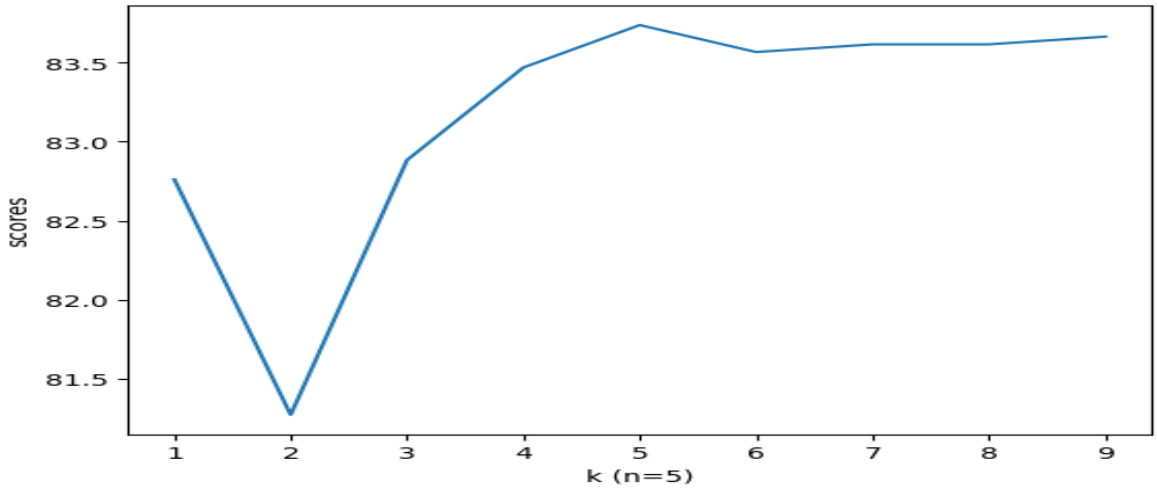
محاسبه امتیاز براساس الگوریتم KNN با فاصله اقلیدوسی([n,k,score]):

```
[(4, 5, 83.86422466422466),
(5, 9, 83.66422466422466),
(5, 5, 83.63980463980464),
(4, 8, 83.62002442002442),
(5, 7, 83.61538461538461),
(5, 8, 83.61538461538461),
(6, 9, 83.6107448107448),
(6, 8, 83.58632478632478),
(7, 8, 83.58168498168497),
(5, 6, 83.56654456654456),
(6, 5, 83.56190476190476),
(4, 6, 83.54676434676435),
(6, 7, 83.51306471306471),
(7, 6, 83.5084249084249),
(7, 9, 83.5084249084249),
(8, 7, 83.5037851037851),
(8, 7, 83.5037851037851),
(4, 7, 83.4979242979243),
(7, 7, 83.45958485958485),
(7, 7, 83.45958485958485),
(8, 8, 83.45494505494506),
(8, 8, 83.45494505494506),
(4, 9, 83.44908424908425),
(7, 5, 83.43516483516483),
(3, 5, 83.40488400488401),
(9, 7, 83.37704517704518),
(8, 9, 83.35726495726496),
(9, 8, 83.32820512820513),
(4, 4, 83.32698412698413),
(5, 4, 83.32234432234432),
(8, 5, 83.28400488400489),
(3, 6, 83.28278388278389),
(8, 6, 83.25958485958486),
(9, 9, 83.23052503052503),
(9, 9, 83.23052503052503),
(9, 9, 83.23052503052503),
(7, 4, 83.21538461538461),
(9, 5, 83.206105006105),
(6, 4, 83.19560439560439),
```

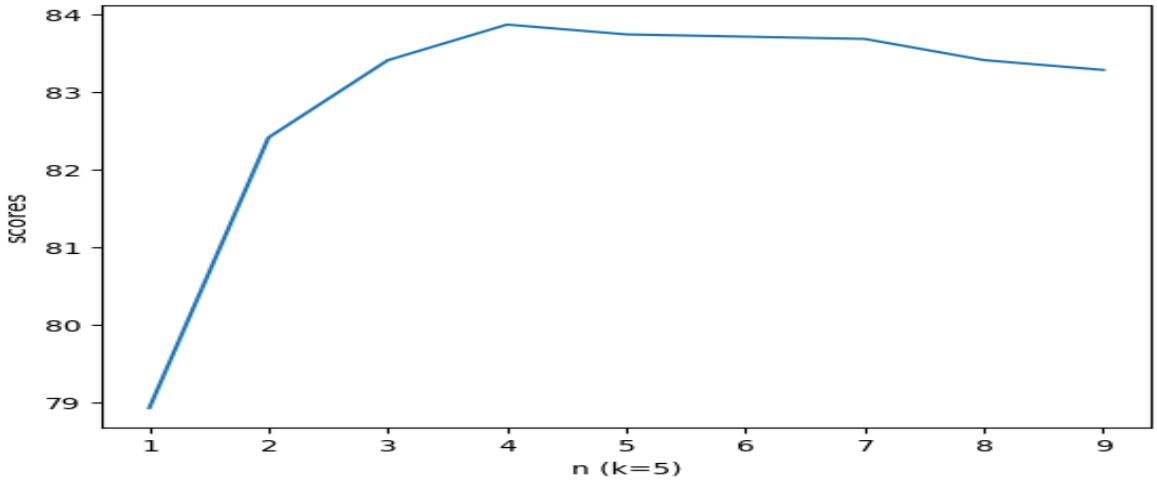
```
(3, 7, 83.18510378510379),
(9, 6, 83.15726495726496),
(8, 4, 82.9909645909646),
(3, 4, 82.9897435897436),
(3, 8, 82.96532356532357),
(4, 3, 82.96068376068376),
(9, 4, 82.88864468864469),
(3, 3, 82.81880341880343),
(3, 9, 82.7943833943834),
(5, 3, 82.73626373626374),
(6, 3, 82.60952380952381),
(7, 3, 82.55604395604395),
(5, 1, 82.36996336996337),
(4, 1, 82.32576312576313),
(3, 1, 82.28156288156289),
(8, 3, 82.25836385836386),
(2, 4, 82.2129426129426),
(9, 3, 82.20488400488401),
(2, 5, 82.1152625152625),
(7, 1, 81.96996336996337),
(2, 7, 81.96874236874235),
(2, 1, 81.82222222222221),
(2, 8, 81.79780219780218),
(8, 1, 81.7943833943834),
(2, 3, 81.77338217338216),
(2, 6, 81.70012210012209),
(9, 1, 81.66764346764347),
(2, 9, 81.45592185592184),
(5, 2, 80.97802197802197),
(4, 2, 80.93382173382173),
(3, 2, 80.81636141636142),
(7, 2, 80.70012210012209),
(8, 2, 80.52454212454212),
(2, 2, 80.43028083028082),
(9, 2, 80.34896214896214),
(1, 1, 80.04420024420024),
(1, 4, 79.36043956043956),
(1, 3, 79.06739926739927),
(1, 2, 78.798778998779),
(1, 5, 78.798778998779),
(1, 6, 78.65225885225885),
(1, 7, 78.23711843711844),
(1, 8, 78.04175824175825),
(1, 9, 77.87081807081807)]
```

(محاسبه کردن امتیاز براساس تعداد کلاسترهای مختلف و تعداد دیتاپوینت های متفاوت)

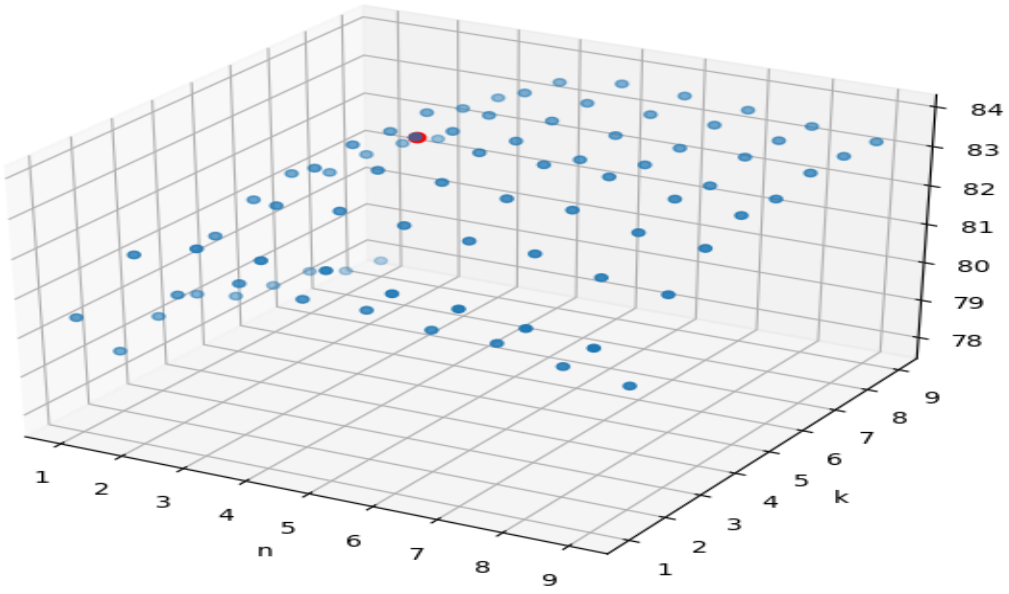
نمایش نتایج به دست امده امتیاز براساس تعداد کلاستر و دیتاپوینت انتخابی:



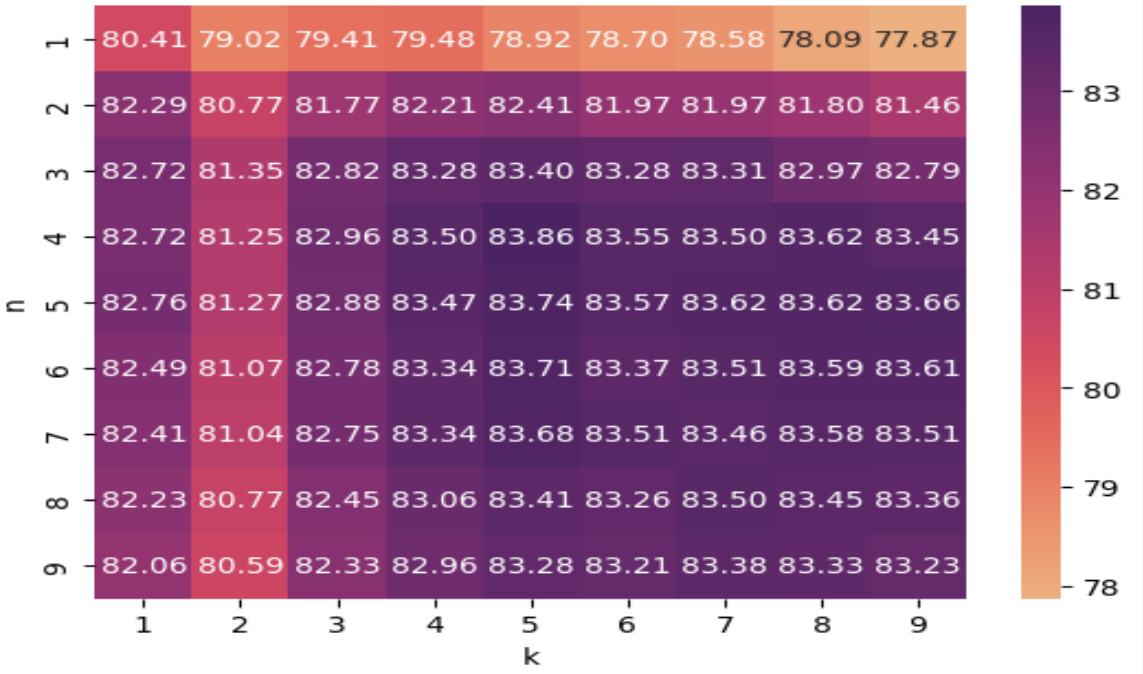
نمودار score براساس تعداد دیتاپوینتهای انتخابی



نمودار score براساس تعداد کلاسترهای نزدیک



نمایش 3 بعدی براساس تعداد کلاستر و دیتاپوینت و امتیاز



نمودار heatmap براساس تعداد کلاستر و دیتاپوینت و امتیاز به دست امده

همانطور که در فرمول مشخص است برای انتخاب هر کلاستر بیشتر، درصد بیشتری از دقت به دست امده کم میشود و از انجاییکه علاوه بر امتیاز ، هزینه زمانی نیز مهم است بنابراین تعداد کلاسترها و تعداد دیتاپوینت های انتخابی باید تعداد محدودی باشند

پارامتر n نشان‌دهنده تعداد نزدیک‌ترین مراکز کلاستری است که هر نقطه داده تست با استفاده از KNN پس از خوشه‌بندی انتخاب شده‌اند. حال اگر مقدار n بزرگ باشد باز زمانی و محاسباتی زیادی را دارد و همچنین ممکن است باعث overfiitشدن شود زیرا تعداد بیشتری از کلاسترها و همچنین داده ها را درنظر میگیرد و از سمت دیگر اگر n کوچک باشد بار محاسباتی و زمانی کمتری را دارد ولی چون تعداد محدودی از کلاستر داده ها را در نظر میگیرد ممکن است برای تشخیص لیبل درست کافی نباشد بنابراین باید در انتخاب n توجه کنیم

تعداد داده ها و توزیع داده ها در هر خوشه:در هر خوشه بندی تعداد متوازن داده ها در هر خوشه امکان نمایش بهتری را به ما میدهد و همچنین اگر توزیع نابرابر باشد به nبزرگتری نیاز داریم که سربار اضافی ایجاد میکند بنابراین باید به توزیع داده ها داخل هر کلاستر هم توجه کرد.

به دست اوردن لیبل هر کلاستر: لیبلی که بیشترین تکرار را دارد به عنوان لیبل کلاستر انتخاب میشود.

لیبل های به دست امده برای هر کلاستر (50 کلاستر):

[43, 54, 77, 72, 42, 29, 98, 16, 1, 79, 36, 7, 76, 50, 57, 100, 35, 69, 73, 75, 80, 88, 71, 49, 64, 93, 48, 68, 28, 34, 87, 45, 74, 22, 50, 2, 56, 60, 21, 40, 41, 55, 58, 70, 59, 53, 12, 46, 27, 65]

سپس برای ارزیابی کردن کلاسترها از 2 تابع purity و rand_indexاستفاده میکنیم .

Purity چیست؟

یک معیار ارزیابی است که برای اندازه گیری کیفیت خوشه بندی استفاده میشود ، معیار ساده و قابل فهمی است البته معمولا برای مجموعه هایی با خوشه های زیاد اندازه های متفاوت خوب کار نمیکند .

درصد خلوص بالا نشان می دهد که یک خوشه عمدتاً حاوی داده هایی از یک کلاس است.

برای خوشه های با خلوص بالا، یک 'n' کوچکتر ممکن است کافی باشد، زیرا خود خوشه به شدت نماینده یک کلاس خاص است.

purity کل داده ها (تمام 50 کلاستر) برابراست با 53.468490473864

purityهر کلاستر همراه با تعداد داده های مربوط به همان کلاستر:

[(57, 0.8947368421052632), (87, 0.4827586206896552), (45, 0.9555555555555556),

(93, 0.9247311827956989), (139, 0.2014388489208633), (158, 0.26582278481012656),

(143, 0.18181818181818182), (44, 0.8863636363636364), (27, 0.9629629629629),

(96, 0.4895833333333333), (60, 0.8833333333333333), (41, 0.975609756097561), (125, 0.96),
(150, 0.32), (67, 0.8805970149253731), (176, 0.1590909090909091),

(113, 0.3274336283185841), (41, 0.9024390243902439), (104, 0.7980769230769231),

(72, 0.7083333333333334), (162, 0.35802469135802467), (144, 0.4861111111111111),

(90, 0.5222222222222223), (93, 0.4946236559139785), (73, 0.6164383561643836),

(90, 0.26666666666666666), (47, 0.5531914893617021), (43, 0.6976744186046512),

(61, 0.5901639344262295), (69, 0.3188405797101449), (135, 0.2962962962962963),

(107, 0.897196261682243), (76, 0.7763157894736842), (46, 0.8913043478260869),

(79, 0.7215189873417721), (125, 0.184), (41, 0.8292682926829268),

(36, 0.8611111111111112), (69, 0.36231884057971014), (102, 0.5588235294117647),

(116, 0.25862068965517243), (100, 0.48), (43, 0.7209302325581395),

(42, 0.9285714285714286), (60, 0.8666666666666667), (33, 0.8181818181818182),

(26, 0.8076923076923077), (33, 1.0), (58, 0.46551724137931033),

(57, 0.49122807017543857)]

Rand_index:

معیار دیگری برای اندازه گیری شباهت بین خوشه بندی های مختلف از دیتاست است
برای محاسبه این معیار مجموع تعداد tp و tn را محاسبه میکنیم به نسبت n*(n-1)/2

TP:69212,

TN:8138045

total=8378371.0

rand index= 0.9795766981433504

انتخاب n بهینه با توجه به چند معیار:

مجموعه داده برای هر کلاستر: هرچه تعداد داده ها بیشتر باشد ممکن است به nبزرکتری برای پوشش داده ها نیاز داشته باشیم البته تعداد داده ها نیز نباید کم باشد.

خوشه بندی: باید به میزان اثربخشی هر خوشه در انتخاب nبهینه توجه کرد

محدودیت منابع :باتوجه به محدودیت منابع ،کم استفاده کردن حافظه و زمان برای ما اهمیت دارد.

بنابراین در اتخاب nباید به معیارهای بالا توجه کرد زیرا حتما باید اعتدالی بین دقت و سرعت برقرار باشد

در راستای پیدا کردن ارتباط میان لیبل های KNN و کلاستر ها به دو صورت تاثیر کلاستر ها در KNN را بررسی کردیم:

ابتدا داده هایی که 3 کلاستر با درصد خلوص زیر 40 درصد برای انها انتخاب شده را بررسی کرده و تشخیص نهایی را با لیبل واقعی داده مقایسه کرده ایم و در نهایت 21 درصد انها تشخیص اشتباه داشته(از بین 305 داده 21 درصد تشخیص اشتباه داشته) که نشان دهنده تاثیر کمی منفی کلاستر هایی است با درصد خلوص پایین در انتخاب نهایی و تشخیص غلط است.

همچنین داده هایی که 3 کلاستر با درصد خلوص بالای 70 درصدبرای انها انتخاب شده را بررسی میکنیم و نتیجه لیبل نهایی را با لیبل واقعی داده مقایسه میکنیم و میزان تاثیر درصد خلوص و تشخیص درست KNN را نیز بررسی کرده ودرنهایت 84 درصد انها تشخیص درست داشته(از بین 2660داده 84درصد تشخیص درست داشته) که نشاندهنده تاثیر مثبت کلاسترهایی است با درصد خلوص بالا است.

Selected_good_data_in_cluster=2660

Best, =[84.39]

Selected_bad_data_in_cluster=305

Worth, =[21.32]