پروژه ی پایانی درس شناسایی الگو

نام : حسین سیم چی

۹۸۴۴۳۱۱۹

استاد : آقای دکتر احمد علی آبین

۱۹ / ۱۱ / ۱۳۹۹

مقدمه

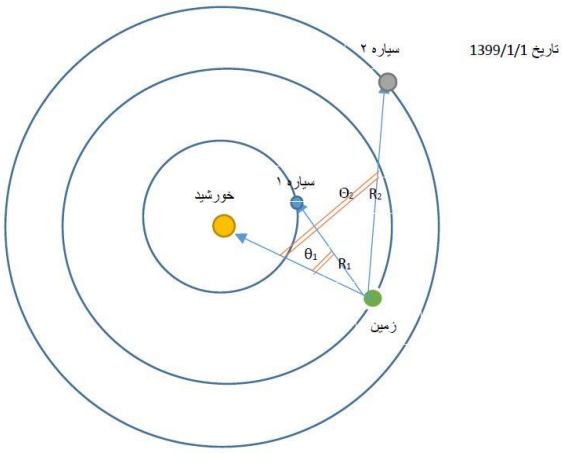
هدف از این پروژه یافتن رابطه ی وقوع زلزله بالای ۴.۵ ریشتر در ایران با چینش و وضعیت سیارات منظومه ی شمسی است. پروژه ی بیان شده دارای سه مرحله زیر است

۱. استخراج یکسری ویژگی از وضعیت سیارات با توجه به تاریخ های داده شده
۲. اضافه کردن ستون برچسب بر اساس اینکه اگر در تاریخ معین، زلزله بالای ۴.۵ ریشتر باشد برچسب یک و در غیر این صورت برچسب صفر را به آن می دهیم
۳. انتخاب یک کلسیفایر مناسب برای دسته بندی داده ها.

در این پروژه سعی شده است از تمامی کلسیفایر های مطرح استفاده کنیم تا دقت تمامی آن ها را با یکدیگر مقایسه کنیم. در این حالت می توان در انتها کلسیفایری را انتخاب نمود که بالاترین دقت را دارد. همچنین از انواع روش های استخراج ویژگی با تعداد پارامترهای متفاوت استفاده شده است.

استخراج ویژگی

در گام نخست لازم است با استفاده از تاریخ های داده شده، ویژگی های مورد نظر را استخراج کنیم که به نوعی ساخت دیتاست در این مرحله انجام می گیرد. همانطور که از فایل پروژه دریافت می شود، در این پروژه به دنبال بدست آوردن ویژگی ها بر اساس شکل زیر هستیم



و در نهایت دیتاست ما به تعداد روزها سطر و به تعداد ویژگی ها ستون دارد که در یک تاریخ خاص به صورت شکل زیر خواهد بود

	سیاره ۱		سیاره ۲		...	سیاره k	
تاریخ	θ_1	r_1	θ_2	r_2	...	θ_k	r_k
۱۳۹۹/۱/۱							

برای بدست آوردن ویژگی های فوق، می توان از کتابخانه ی Solarsystem در پایتون استفاده نمود که تاریخ خاص را به آن می دهیم و در آن تاریخ موقعیت سیارات نسبت به زمین را به ما برمی گرداند. **ازآنجایی که مقادیر ویژگی ها بسیار تاثیر گذار در نتایج حاصل هستند و در راستای بهبود مقادیر ویژگی ها و بدست آوردن آن ها نیاز دیدم تا کدی را نوشته، که ویژگی های فاصله و زاویه هر سیاره از زمین و موقعیت سیارات را به ما برگرداند. که برای بدست آوردن فرمول های نوشته شده در فایل کد از تجربیات افراد متخصص در زمینه ی نجوم و فیزیک استفاده شده است.** کد نوشته شده را در ادامه به اختصار توضیح داده و نحوه ی بدست آوردن ویژگی ها را توضیح خواهم داد.

نحوه ی کار با کد نوشته شده به این صورت است که باید تاریخ، ساعت و دقیقه را به عنوان ورودی به آن بدهیم و درمقابل به عنوان خروجی ویژگی های بدست آمده در آن تاریخ را در فایلی جداگانه ذخیره می کند.

که در نهایت دیتاست حاصل مانند زیر ذخیره خواهد شد

Year	Month	Day	Hour	Minute	Lat	Long	Mag	R_mercury	theta_mercury	R_venus	theta_venus	R_mars	theta_mars	R_jupiter	theta_jupiter	R_saturn	theta_saturn	R_uranus	theta_uranus	R_neptun	theta_neptun	R_moon	theta_moon
1900	6	1	0	0	38.5	43.3	5	192.5159302	13.15093763	65.27180013	40.12308218	354.0427812	25.27602055	629.4010441	175.3319296	1292.576756	159.3061887	2724.691696	170.8840655	4639.340352	15.13615691	0.384	103.598582
1900	7	12	0	0	40.28	43.1	5.9	91.87882145	2.222311082	41.7415198	6.231324518	328.5080817	37.13372318	672.446117	130.8709456	1294.39091	157.2280688	2775.778809	129.1127687	4631.614385	23.76144273	0.384	4.178578939
1902	2	13	9	39	40.72	48.71	6	91.75137583	1.192602563	41.44199979	2.193678849	364.2532181	18.8781458	915.9008157	21.46164424	1559.733508	30.94982729	2915.364742	71.93596224	4408.990202	124.3506285	0.384	30.64451608
1902	2	21	0	0	41.8	48.8	5.6	100.4706741	14.3562741	43.17772083	13.95953185	367.0535322	16.73688077	908.5148151	27.29510495	1549.101831	37.60910376	2896.666053	79.24051059	4425.711856	116.7795719	0.384	115.7819526
1902	7	9	3	38	27.08	56.34	6.4	164.5801012	20.53185626	185.6895318	35.63715118	359.7034003	21.94060546	643.5128062	151.8275663	1285.216964	171.9624215	2754.027176	141.4260519	4638.478587	16.31610043	0.384	136.1811814
1902	9	5	4	33	39.5	48	4.8	155.9189432	21.72140897	230.9403197	22.13153118	324.5765749	38.69856384	652.4313193	144.1225361	1338.146021	127.1716153	2884.808663	83.79820157	4539.343397	71.89366498	0.384	154.1580251
1902	10	3	23	5	41.9	45.6	5.2	91.93601789	2.55172338	245.6611809	14.92444507	300.9501367	47.34471671	704.4464723	114.5561494	1403.415245	98.62450623	2952.888163	56.26374048	4466.672223	100.013703	0.384	116.5613037
1902	10	4	1	46	41.9	45.6	4.9	91.97809246	2.768749553	245.7074896	14.89607444	300.8510044	47.37894635	704.6862122	114.4463591	1403.690092	98.51636206	2953.124437	56.15792549	4466.390655	100.1238698	0.384	117.8141258
1902	10	17	7	21	41.9	45.9	5.2	116.5698392	20.74575253	250.5752298	11.52357267	288.7407563	51.4718879	734.2739398	101.7341656	1436.496445	85.86653802	2978.816784	43.6888133	4434.035821	113.2113635	0.384	93.90931321
1902	10	26	11	37	39.7	47.8	4.7	144.7587232	22.61055717	253.2286963	9.170150025	279.9174225	54.36468657	755.6980369	93.23471116	1459.043837	77.26046046	2993.612149	35.09665055	4413.287622	122.3435233	0.384	8.93104519
1902	12	2	0	47	40.7	48.7	4.9	207.1439559	2.086806552	257.7961663	0.2657385	241.713266	66.47470489	838.6010023	61.58046453	1537.075356	44.13626913	3022.067796	1.159068018	4355.046948	159.0770153	0.384	58.47160685
1902	12	4	22	18	37.8	65.5	4.7	205.7779788	4.578182273	257.7440478	1.015194435	238.5002306	67.48600718	844.5345646	59.19593151	1541.972549	41.57483899	3022.044465	1.522114092	4352.57923	162.0063909	0.384	90.92826052
1902	1	1	0	0	39	43.3	5	193.7059308	12.65105889	70.49829937	42.13092039	342.151972	31.25992944	924.6344314	11.48518469	1582.002236	6.60966409	2999.739635	30.97083595	4348.636797	168.0471434	0.384	95.69170971
1903	1	2	7	15	36.5	54.9	5	146.1845787	22.53883914	254.0189561	8.340924305	205.9079043	102.0293481	893.7151681	36.52232227	1576.060643	16.80251386	3003.936203	27.83893621	4348.036574	169.2547497	0.384	48.85717421
1903	2	9	5	18	36.58	47.65	5.6	113.694438	20.05754734	240.1091764	17.99792626	160.4787937	94.5449756	926.6991935	7.44076927	1576.766205	15.93083812	2936.156174	63.48883944	4395.485059	131.0261326	0.384	113.7506435
1903	3	22	14	35	33.16	59.71	6.2	206.9683853	2.54914326	213.9917335	28.12651287	112.9718978	119.7947028	912.8819764	24.00251909	1520.210194	52.24493067	2833.733311	103.5628522	4492.571119	90.01485866	0.384	142.4950557
1903	4	19	0	0	39.1	42.4	6.8	173.0579074	18.9358175	191.1009282	34.37402099	88.51275598	145.1137903	876.4258771	45.35048294	1459.155612	77.21756134	2772.353765	130.8845471	4560.133557	63.38331189	0.384	164.4463742
1903	4	28	23	39	39.14	42.65	7	143.8421673	22.64987196	181.7792507	36.5067892	82.65773149	156.7684733	858.590984	53.35102825	1434.607362	86.58664818	2754.716656	140.9867255	4581.901472	53.76824326	0.384	83.6633781
1903	5	28	3	58	40.9	42.75	5.8	95.98764045	10.46945856	151.9918556	42.0391169	80.10210397	164.9262257	796.5410214	77.73215386	1364.034692	114.9183671	2724.722577	170.8057007	4629.211584	25.88915961	0.384	116.6923566
1903	6	3	11	29	41.8	46.3	4.9	109.1886488	18.67116374	145.1199361	43.04366692	82.5895998	156.9454023	781.8746918	83.25150484	1350.193636	121.241089	2723.058653	177.2895356	4635.489101	19.88862403	0.384	45.9507027
1903	6	24	16	56	37.48	48.96	5.9	172.1774552	19.12270744	121.1927858	45.6216652	97.22433648	133.7224522	732.1516556	102.6014322	1311.240811	142.9675784	2730.700788	160.9218238	4644.698545	0.248577835	0.384	168.0934664
1903	7	5	0	0	41.799	48.7	5.5	194.901279	12.12175532	109.2778059	46.23810307	106.8510337	124.5722206	709.0458197	112.4727785	1297.826314	153.7248644	2741.475779	150.3977112	4642.347534	10.00990614	0.384	52.74044934
1903	7	8	4	43	41.4	44.5	4.9	199.7846968	9.58165471	105.5590083	46.31177064	110.0720668	121.9800229	702.1674142	115.6067983	1294.519455	157.0877374	2745.68958	147.1396245	4640.714057	13.04236684	0.384	16.92218589
1903	7	9	13	21	41.4	44.5	5.2	201.5142881	8.468111973	103.9761127	46.32333029	111.4702205	120.9101483	699.294483	116.9499419	1293.242505	158.5209596	2747.600468	145.7553444	4639.891125	14.33263719	0.384	1.686000828
1903	9	25	1	20	35.18	58.23	5.9	103.4578332	16.16159518	42.22914627	9.648828563	202.0679154	100.7463228	636.8133828	159.3614201	1352.123643	120.331889	2922.288687	69.17141506	4495.700377	88.81669673	0.384	146.7220771
1903	11	2	22	12	41.1	47.2	5.2	202.025324	8.106493622	73.67409629	43.07863978	246.4922785	64.9726612	696.1665611	118.4377534	1445.197751	82.55151731	2997.539938	32.50449298	4402.903493	127.2828686	0.384	137.7315745
1904	3	20	9	59	36.6	59.4	5.6	197.205253	11.01059907	216.3011682	27.39597462	358.3086705	22.80326213	928.0072258	2.664342529	1545.666911	39.55919948	2848.358684	97.80462543	4483.148833	93.62978098	0.384	25.88955303
1904	4	28	15	25	40.6	48.6	5.1	96.02509514	10.51138793	240.5637648	17.76681114	372.3184304	11.75157875	908.6631365	27.1891734	1465.085898	74.93499536	2761.405856	136.92482	4578.067844	55.53440944	0.384	53.6564885
1904	7	5	21	54	41	49	5.5	206.18675	4.000976695	257.7961785	0.265316023	375.4949678	7.296907347	782.941114	82.84956754	1312.592527	142.0401392	2738.170144	153.2062115	4642.599262	9.457782776	0.384	98.6483767
1904	8	27	16	9	42	42	4.9	91.74541688	1.121370957	248.139026	13.32016018	359.4138631	22.12220871	668.1272578	133.4364798	1290.318634	162.2182938	2842.149381	100.2325362	4568.475449	59.80716703	0.384	30.15184804
1904	11	9	3	28	36.94	59.77	6.4	194.1269312	12.46791321	203.1368855	31.29139745	311.848014	43.4911391	649.9709393	146.0799351	1430.632462	91.89668337	3000.916722	30.121895	4393.600714	132.0117733	0.384	73.12350599
1905	1	9	6	17	37	48.68	6.2	126.8278966	22.28320806	143.0808041	43.32097984	253.3128454	62.82962084	774.5318721	93.9743385	1557.156844	32.66940285	3005.526612	26.56543728	4349.648385	166.2470062	0.384	37.9659018

*** فایل دیتاست در پوشه ی مربوط به پروژه قرار داده شده است.

نحوه ی بدست آوردن ویژگی ها

منظومه شمسی را در یک مختصات دایروی به مبدا خورشید فرض کرده ایم. موقعیت هر سیاره با دو پارامتر فاصله شعاعی از خورشید (R_i) و زاویه (θ_i) می توان به دست آورد. فاصله هر سیاره با خورشید ثابت فرض شده است. هر سیاره با یک سرعت زاویه ای ثابت به دور خورشید به صورت پادساعتگرد در حال چرخش هست. لذا می توان موقعیت هر سیاره در زمان دلخواه t طبق رابطه زیر بدست آورد:

$$\theta_i = \omega_i(t - t_0) + \theta_{i,0}$$

که در رابطه بالا ω_i سرعت زاویه ای سیاره i ام و θ_i زاویه این سیاره در زمان t می باشد. همان طور که مشاهده می شود در این رابطه نیاز است تا زاویه سیاره ($\theta_{i,0}$) در یک زمان مبدا (t_0) معلوم باشد. در مراجع معتبر فاصله سیارات تا خورشید به همراه تناوب مداری (Orbital Period)، مدت زمانی که طول می کشد تا سیاره یک دور کامل (۳۶۰ درجه) به دور خورشید بگردد) آن ها ذکر شده است. سرعت زاویه ای هر سیاره را می توان از روی تناوب مداری آن به صورت زیر محاسبه کرد:

$$\omega_i = \frac{360°}{Orbital\ Period}$$

برای داشتن زاویه سیاره ها ($\theta_{i,0}$) در یک زمان مبدا (t_0)، از مراجع معتبر موقعیت سیاره ها در تاریخ ۱۵ ژانویه ۲۰۲۱ در ساعت ۴ و ۴۵ دقیقه به وقت تهران به دست آمد. این تاریخ به عنوان زمان مبدا (t_0) انتخاب شده و موقعیت سیارات در این زمان مرجع محاسبات بعدی انتخاب شده است.

حال با داشتن تمام پارامترهای معادله اول، می توان زاویه هر سیاره در زمان دلخواه t را به دست آورد. در دیتاست داده شده، تاریخ و زمان وقوع هر زلزله مشخص است. در جاهایی که ساعت و یا دقیقه وقوع زلزله مشخص نبود، آن را با عدد ۱ فرض کردیم. سپس اختلاف زمانی لحظه وقوع هر زلزله با تاریخ مبدا (۱۵ ژانویه ۲۰۲۱، ساعت ۴ و ۴۵ دقیقه به وقت تهران) به دقیقه محاسبه شده است. سپس با کمک معادله اول زاویه هر سیاره در منظومه شمسی در لحظه وقوع هر زلزله به دست آمده است. از آنجا که به زمان های قبل از تاریخ مبدا بر می گردیم، سرعت زاویه منفی (ساعتگرد) در محاسبات لحاظ شده است. با داشتن موقعیت مکانی هر سیاره در مختصات دایروی، می توان موقعیت آن در مختصات کارتزین را محاسبه نمود:

$$\begin{cases} X_i = R_i \cos(\theta_i) \\ Y_i = R_i \sin(\theta_i) \end{cases}$$

حال می توان فاصله هر سیاره از کره زمین را به صورت زیر محاسبه نمود

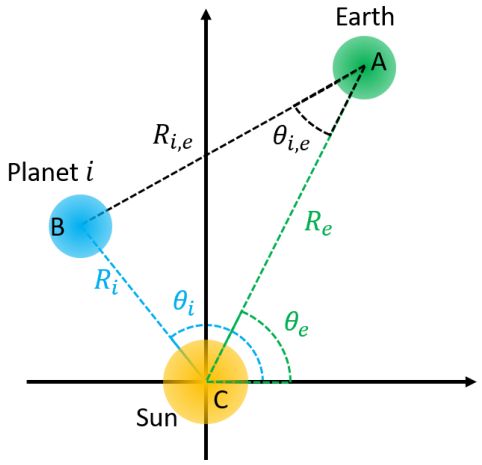
$$R_{i,e} = \sqrt{(X_i - X_{earth})^2 + (Y_i - Y_{earth})^2}$$

برای محاسبه زاویه هر سیاره از زمین ($\theta_{i,e}$) از قانون سینوس ها استفاده شده است. در مثلث ABC در شکل زیر می توان قانون مثلث ها را به صورت زیر نوشت:

$$\frac{R_i}{\sin\theta_{i,e}} = \frac{R_{i,e}}{\sin(\theta_i - \theta_e)}$$

$$\Rightarrow \theta_{i,e} = \sin^{-1}\left(\frac{R_i}{R_{i,e}}\sin(\theta_i - \theta_e)\right)$$

مختصات ماه نیز به صورت مشابه به دست آمده است با این تفاوت که برای ماه، زمین مبدا مختصات قرار گرفته و با داشتن تناوب مداری ماه به دور زمین، موقعیت ماه نسبت به زمین به دست آمده است. با داشتن موقعیت زمیین نسبت به خورشید، زاویه ماه از زمین به صورت مشابه از قانون سینوس ها به دست می آید.



اضافه کردن ستون برچسب

با توجه به ستون "Mag" که حاوی میزان زلزله ی آمده در تاریخ خاص است، برچسب موردنظر را به آن اختصاص می دهیم. به این صورت که اگر زلزله بالای ۴.۵ ریشتر اتفاق افتاده باشد برچسب یک و درغیر این صورت مقدار صفر را به خود اختصاص می دهد.

***** برای جلوگیری از افزایش حجم گزارش، کد قرار داده نشده است.**

دسته بندی با استفاده از روش های موجود و تکنیکهای خاص

در این قسمت با یکدیگر به بررسی دقت های حاصل، مشکلات موجود و راه های رفع آن ها خواهیم پرداخت.

مشکل اول، بالانس نبودن کلاس ها

یکی از مشکلات موجود در دنیای واقعی یا به طور خاص، دیتاست های حاصل از دنیایی است که در آن زندگی می کنیم. و آن مشکل بالانس نبودن دسته ها است. در سال های اخیر راه های زیادی برای رفع این مشکل مطرح و استفاده شده است که به آن ها خواهیم پرداخت. مشکل بالانس نبودن دسته ها بدین معنا است که فرض کنید در یک دیتاست ۲۰ نوع داده از کلاس اول و ۲۰۰۰ نوع داده از کلاس دوم داریم. بدیهی است هرچه قدر ویژگی های خوبی داشته باشیم بازهم به دلیل بالانس نبودن داده های درون کلاس ها دقت خوبی را نخواهیم گرفت. درنتیجه در تمامی روش هایی که بررسی خواهیم کرد با تکنیک هایی که در ادامه به ان خواهیم پرداخت سعی می کنیم مشکل گفته شده را رفع کنیم.

✓ تکنیک های استفاده شده جهت رفع مشکل

برای رفع این مشکل از متد Oversampling و undersampling **همزمان** استفاده می کنیم. به طور کلی روش های متفاوتی برای اینکار وجود دارد که در این پروژه برای استفاده از Oversampling از روش Synthetic Minority Oversampling Technique (SMOTE) استفاده می کنیم که در آن برخلاف روش های قبلی که در برخی از آنها با تولید و جایگذاری از نمونه هایی که کلاس انها کمتر از بقیه است کار می کند و برخی روش های دیگر مانند (ADASYN) که نزدیک ترین سمپل را درفضای دو بعدی پیدا می کند و ان را کپی میکند ، در این روش ابتدا دو کلاس از هم تفکیک شده و داده هایی که تعداد کلاس انها از کلاس دیگر کمتر است را به صورت رندم کپی میکنیم تا تعداد آنها با تعداد سمپل های کلاس دیگر برابر شود . ولی اینکار باید قبل از Undersampling صورت پذیرد. البته این تکنیک کاملا منحصر به فرد بوده و اصلا ضرورتی برای اینکار وجود ندارد. اگر تعداد داده های با برچسب یک و صفر را محاسبه کنیم متوجه می شویم تعداد داده های برچسب صفر بسیار زیاد است درنتیجه در ابتدا سعی می کنیم با تکنیک undersampling تعدادی از داده های موجود را کنار گذاشته که اینکار نیز درصد خاص مربوط به خود را دارد که به اسم Smapling strategy می توانیم میزان درصد آن را مشخص کنیم. مزیت undersampling این است که باعث می شود که سرعت اجرای کد بر روی دیتاست افزایش یابد که در بحث های ارائه ی الگوریتم ها سرعت، یک پارامتر بسیار مهم و غیرقابل انکار است. البته همانطور که بیان شد می توانیم از این ایده استفاده نکنیم. در ادامه پس از

انجام این کار باید با اســتفاده از تکنیک Oversampling گفته شــده تعداد داده های کلاس کمتر را به کلاس بیشتر برسانیم. **در این پروژه از مرحله ی دوم به بعد این رویکرد استفاده شده است.**

***** برای اجرا دو متد نام برده شده از کتابخانه ی imblearn موجود در پایتون استفاده شده است.**

مشکل دوم، بررسی انتخاب ویژگی مناسب

یکی دیگر از مشکلات موجود در دیتاست ها، وجود ویژگی هایی است که تاثیر زیادی در دسته بندی ندارند یا به عبارتی دیگر در به روز رسانی مقادیر وزن ها جهت بدست آوردن خط جهت تفکیک کلاس ها تاثیری ندارند.که با استفاده از متدهای رایج می توانیم ویژگی هایی را انتخاب کنیم که قدرت تفکیک پذیری بالایی دارند.

✓ تکنیک های استفاده شده

تکنیک بررسی شده جهت این کار استفاده از Kbest selection است که k تا از بهترین ویژگی ها را انتخاب و نگهداری می کند. که انتخاب آن بر اساس تست های آماری صورت می گیرد. از کتابخانه ی Sklearn جهت انتخاب ویژگی استفاده شده است. پارامترهای مختلفی می توانیم به عنوان روش انتخاب به این متد بدهیم که برای کلاس بندی معمولا پارامتر "Chi2" استفاده می شود که در این پروژه ما نیز از این پارامتر استفاده کرده ایم. که البته تعداد مقدار k خود چالش بزرگی است که در این پروژه ۲ مقدار مختلف را تست خواهیم کرد.

مشکل سوم، بررسی استخراج ویژگی

گاهی اوقات زمانی که پیچیدگی توزیع داده ها از حدی بیشتر باشد نمی توان با استفاده از ویژگی های موجود دسته بندی را به درستی انجام داد. لذا باید ویژگی هایی را از بین ویژگی های موجود استخراج کنیم. تکنیک استفاده شده

تکنیک بررسی شده، Principle component analysis (PCA) می باشد که از بین ویژگی های موجود، تعدادی ویژگی را **استخراج** میکند و سپس داده ها را در فضایی جدید با توجه به ویژگی های بدست آمده ترسیم میکند. انتخاب تعداد مولفه های PCA خود چالشی است که برای ۲ مقدار پروژه را تست خواهیم کرد.

مشکل چهارم، بررسی ترتیب قرار گیری داده ها در دیتاست بر اساس برچسب

در بعضی مواقع امکان این وجود دارد که داده هایی که برچسب یک دارند در ابتدا پشت سر هم قرار گیرند و سپس داده هایی با برچسب صفر قرار بگیرند. در این صورت زمانی که مدل را بر روی داده ها اجرا کنیم، به دلیل اینکه داده ها به درستی پخش نشده اند مقدار precision، recall و f measure درستی نخواهند داشت. به عنوان مثال فرض کنید ۲۰ داده ی اول دارای برچسب یک و ۲۰ داده ی دوم دارای برچسب صفر باشند، زمانی که از تکنیک K-fold cross validation استفاده کنیم به دقت مناسبی نخواهیم رسید

✓ تکنیک استفاده شده

در این پروژه از مرحله ی دوم به بعد از ماژول shuffle موجود در کتابخانه ی sklearn استفاده شده است که باعث می شود ترتیب داده ها به هم ریخته شده و در هر فولد از دو کلاس به تعداد نسبتا مساوی داده داشته باشیم.

نتایج بدست آمده

بررسی دقت های حاصل بر روی الگوریتم های دسته بندی SVM، Decision Tree، Gradient boosting و Random Forest با استفاده از تکنیک K-Fold Cross validation و رسم ماتریس درهم ریختگی در هر مرحله صورت گرفته است.

***** تمام تکنیک های مطرح شده با استفاده از کتابخانه ی Sklearn در پایتون زده شده است.**

مراحل بررسی شده به شرح زیر است

۱. بدست آوردن دقت بر روی دیتاست تولید شده بدون استفاده از راه حل های معرفی شده (جهت مقایسه با راه حل های بیان شده و مشاهده ی اینکه چقدر راه حل های بیان شده در بهبود دقت موثر اند)
۲. استفاده از روش انتخاب ویژگی Kbest selection
۳. استفاده از روش استخراج ویژگی PCA
۴. استفاده از روش های دوم و سوم به صورت همزمان
۵. انتخاب بهترین کلاس بند و روش از بین روش های معرفی شده
۶. نتیجه گیری نهایی

مرحله ی اول (بدست آوردن دقت بر روی دیتاست تولید شده بدون استفاده از راه حل های معرفی شده)

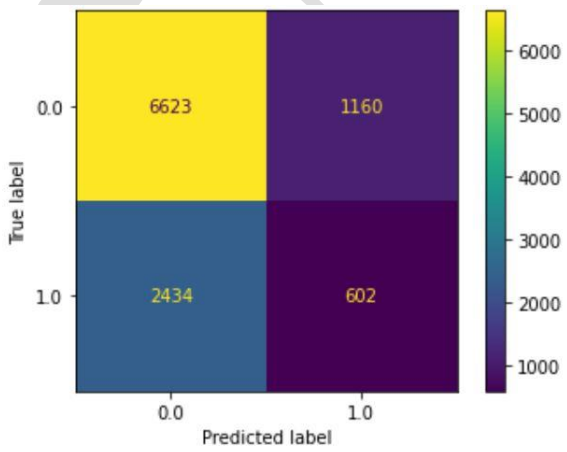
تنها کاری که در این مرحله می کنیم این اســت که می خواهیم دیتاســت را بخوانیم و مدل های نام برده شــده بدون هیچ گونه اعمال رویکردهای نام برده شده خروجی بگیریم. از خروجی این مرحله می توانیم برای مقایسه با دقت های حاصل از مراحل بعدی مقایسه کنیم.

***** برای محاســبه ی ســه دقت دیگر از معیار Macro اســتفاده شــده اســت. زیرا در معیار Micro تفاوتی بین دقت ها نیست و همگی مقداری مشابه دارند**

Decision Tree Classifier

Fold / Measure	Accuray	Precision	Recall	F-measure
Fold 1	66	53.64	52.46	51.87
Fold 2	56.34	49.65	42.79	36.74
Fold 3	56.38	50.31	53.68	38.24
Fold 4	51.13	50.02	50.42	35.12
Fold 5	65.52	49.95	49.08	40.59
Overall	59.07	50.71	49.68	40.51

برای فولد اول Confusion matrix به صــورت زیر اســت، توجه داشــته باشــید جهت کاهش حجم گزارش از گذاشتن تمامی ماتریس ها خودداری نموده ام.



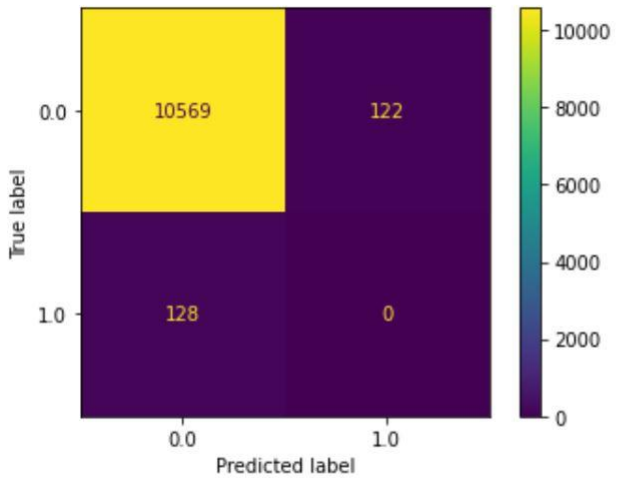

```
0.6678066364728718
(0.5364573992661185, 0.5246222172340412, 0.518758731617192, None)
```

مشکل اول مطرح شده در قسمت های قبل به وضوح در ماتریس Confusion قابل مشاهده است.

Random Forest Classifier

Fold / Measure	Accuray	Precision	Recall	F-measure
Fold 1	71.93	35.96	50	41.83
Fold 2	97.68	49.40	49.42	49.41
Fold 3	97.85	48.92	50	49.45
Fold 4	96.29	49.86	49.79	49.79
Fold 5	97.65	49.41	49.39	49.40
Overall	92.28	49.71	49.72	47.97

برای فولد دوم داریم




```
0.9768925039282743
(0.4940170141161073, 0.4942942662052193, 0.49415560127174113, None)
```

مشکل اول در شکل فوق نیز قابل مشهود است. درضمن ذکر این نکته ضروری است که ما به دنبال دقت بالای برای تمامی معیارها هستیم و صرف داشتن Accuracy بالا لزوما خوب نیست.

Support Vector machine (SVM)

Fold / Measure	Accuray	Precision	Recall	F-measure
Fold 1	71.93	35.96	50	41.83
Fold 2	98.81	49.40	50	49.70
Fold 3	97.85	48.92	50	49.45
Fold 4	98.53	49.26	50	49.62
Fold 5	98.85	49.42	50	49.71
Overall	93.19	46.59	50	48.06

دقیقا مشکل اول که در روش های قبل داشتیم در اینجا نیز داریم.

***** تنها برای بعضی قسمت ها ماتریس درهم ریختگی ترسیم شده است که بتوانیم تحلیل خوبی از آن داشــته باشــیم. ولی اگر فایل کد را اجرا نمایید خروجی آن برای تمامی قســمت ها دارای ماتریس درهم ریختگی است**

Gradient Boosting Classifier

Fold / Measure	Accuray	Precision	Recall	F-measure
Fold 1	71.29	56.67	51.13	46.09
Fold 2	98.79	49.40	49.99	49.69
Fold 3	97.84	48.92	49.99	49.45
Fold 4	97.47	50.11	50.08	50.08
Fold 5	95.24	49.79	49.37	49.35
Overall	92.12	50.97	50.11	48.93

تحلیل مانند تحلیل دسته بند های قبل است.

جمع بندی کلاس بندهای خروجی گرفته شده در جدول زیر آمده است.

Model / Measure	Accuracy	Precision	Recall	F-measure
SVM	93.19	46.59	50	48.06
Decision Tree	59.07	50.71	49.68	40.51
Gradient B	92.12	50.97	50.11	48.93
Random F	92.28	49.71	49.72	47.97

مشکلات بیان شده به وضوح قابل مشاهده است. همچنین این نکته را باید ذکر کرد که در انتخاب الگوریتم باید الگوریتمی را انتخاب کنیم که ۴ دقت حاصل از معیارهای فوق دارای مقدار بالا و در یک سطح باشد. زمانی که دقت ها باهم فاصله ی زیادی داشته باشند و یا نوسان دقت بشدت بالا باشد مشخص است که باید تغییری در روند حل مسئله ایجاد کنیم. در ادامه می خواهیم از راهکارهایی استفاده کنیم تا دقت های فوق را افزایش دهیم تا در ادامه بر اساس نتایج بدست آمده بهترین دسته بند را انتخاب نماییم.

از مرحله ی بعد راه کارهای مقابله با بالانس نبودن داده ها و شافل کردن آن ها را بکار خواهیم گرفت.

مرحله ی دوم، استفاده از روش انتخاب ویژگی Kbest selection

۱. K = 5

چون می خواهیم به ازای ۲ مقدار برای k خروجی ها را مقایسه کنیم، درنتیجه در این قسمت از تفکیک کردن نتایج جهت کاهش گزارش خودداری می کنیم.

کد استفاده شده در زیر آمده است

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
X = SelectKBest(chi2, k=5).fit_transform(x, y)
```

در ادامه جدول مقایسه بر اساس این تعداد ویژگی انتخاب شده را مشاهده می کنید

Model / Measure	Accuracy	Precision	Recall	F-measure
SVM	59.69	59.71	59.70	59.68
Decision Tree	80.55	80.55	80.54	80.54
Gradient B	68.81	69.47	68.80	68.54
Random F	79.54	80.67	79.54	79.35

به تفاوت دقت ها مشـــاهده کنید!!!!. دلیل یک رنج شـــدن دقت ها بخاطر بکاربردن تکنیک های Undersampling ، Oversampling و Shuffle کردن دیتاست است. همچنین در این قسمت از ۵ ویژگی با بالاترین درصد تفکیک کنندگی استفاده کرده ایم. درصد بسیار خوبی به ازای انتخاب ۵ ویژگی بر روی درخت تصمیم بدست آمده است. ولی باید روش ها و کاربردهای دیگری را نیز بررسی کنیم. **در ادامه میخواهیم بررسی کنیم اثر افزایش تعداد ویژگی ها چه تاثیری بر روی دقت خواهد داشت.**

***** تعداد ویژگی ها برابر با ۱۶ می باشد.**

۲. K = 10

Model / Measure	Accuracy	Precision	Recall	F-measure
SVM	53	53	53	52.98
Decision Tree	80.55	80.56	80.55	80.55
Gradient B	71.59	71.89	71.59	71.43
Random F	83.56	84.04	83.56	83.50

افزایش تعداد ویژگی های منتخب بر روی SVM تاثیر منفی داشـــته، بر روی Decision Tree تقریبا تاثیری نداشـــته ولی بر روی دو الگوریتم دیگر تاثیر مثبت داشـــته اســـت. ولی دقت نمایید، دو کلاس بندی که به ازای مقادیر K های ویژگی به صـورت منتخب انتخاب شـده اند (**با رنگ قرمز مشـخص شـده اند**) ، کلاسـیفایر Random Forest با تعداد ویژگی ۱۰، درصد بیشتری نسبت به دسته بند Decision Tree با تعداد ویژگی ۵ داشـته اسـت. درنتیجه به عنوان خروجی در این مرحله، دسـته بند Random Forest با تعداد ویژگی ۱۰ بهترین عملکرد را داشته است.

البته توجه داشته باشید که پیدا کردن تعداد مورد مناسب K کار ساده ای نیست که در این پروژه به اختصار دو مقدار که نشان دهنده ی تعداد ویژگی نسبتا بالا (۱۰) و تعداد ویژگی نسبتا کم (۵) بررسی شده است. (کل تعداد ویژگی ها برابر با ۱۶ است)

مرحله ی سوم، استفاده از روش استخراج ویژگی PCA با تعداد مولفه ی ۵ و ۱۰

۱. Component = 5

حال درنظر داریم تا بر اساس ۱۶ ویژگی موجود، ۵ ویژگی از بین آن ها **اسـتخراج** کنیم و داده ها را در فضـا ی بعدی درنظر بگیریم و Mapping را انجام دهیم. بر این موضـوع معتقدیم که PCA و یا هر الگوریتم اسـتخراج ویژگی می تواند ویژگی هایی را استخراج نماید که قابلیت بدست آوردن ساختارهای پیچیده را نیز داشته باشد. در این قسمت نیز برای اجرای PCA از دستور موجود در کتابخانه ی Sklearn استفاده شده است.

Model / Measure	Accuracy	Precision	Recall	F-measure
SVM	72.27	72.70	72.28	72.15
Decision Tree	81.90	81.90	81.90	81.90
Gradient B	72.67	72.69	72.67	72.66
Random F	81.99	82.05	81.99	81.98

همانطور که از نتایج فوق قابل برداشت است در این قسمت نیز دسته بند Ranfom forest با دقت نزدیک به ۸۲ و ماکزیمم عمق ۱۰ بالاترین دقت را داشـته اسـت. **در قسـمت بعد میخواهیم ببینیم آیا با افزایش تعداد**

مولفه ها دقت بهبود می یابد یا خیر. در پایان این مرحله نتیجه گیری از مراحل دوم و سوم خواهیم داشت و در نهایت با استفاده از نتایج حاصل به سراغ قسمت بعد خواهیم رفت.

۲. Component = 10

Model / Measure	Accuracy	Precision	Recall	F-measure
SVM	79.74	81.24	79.74	79.49
Decision Tree	82.56	82.56	82.56	82.55
Gradient B	76.96	78.90	76.96	76.57
Random F	80.95	82.83	80.95	80.67

همانطور که مشاهده می کنید با افزایش تعداد مولفه های PCA به جز کلسیفایر Random Forest، در سه دسته بند شاهد افزایش دقت بوده ایم.

نتیجه گیری مراحل دوم و سوم

- ✓ افزایش تعداد k در انتخاب ویژگی و همچنین تعداد اجزا در روش PCA، باعث افزایش دقت در بسیاری از روش ها خواهد شد
- ✓ دسته بند های Decision Tree و Random Forst بالاترین دقت را در مراحل دوم و سوم داشته اند.

مرحله ی چهارم، استفاده از روش های دوم و سوم به صورت همزمان

در این مرحله قصد داریم با استفاده از دانش کسب شده از مراحل قبل، بررسی کنیم که آیا اگر در ابتدا ویژگی ها را دریافت و سپس از بین آن ها ویژگی هایی را استخراج کنیم دقت بهبود می یابد یا خیر و نتایج را با یکدیگر مقایسه کنیم. بررسی تمامی حالات ممکن امری ناممکن و دور از عقل می باشد. لذا در این مرحله سعی می کنیم با استفاده از دانش بدست آمده از مراحل اول تا سوم، یک رویکرد را مورد بررسی قرار دهیم. در مراحل دوم و سوم دیدیم هرچه مقدار ویژگی ها را افزایش دهیم خروجی بهتر و قابل قبول تری را بدست می آوریم لذا در این مرحله سعی می کنیم در ابتدا ۱۲ ویژگی را انتخاب و سپس از بین ۱۲ ویژگی انتخاب شده، ۱۰ ویژگی را استخراج نماییم.

دلیل اینکه در گام دوم تعداد ویژگی های منتخب را ۱۲ قرار داده ایم این است که در این پروژه فرض کرده ایم که ۱۶ ویژگی داریم و درنتیجه انتخاب ۱۵ ویژگی از ۱۶ تا خیلی قابل قبول و موجه نیست و کاری اضافه می باشد. علت این موضوع که در ابتدا ویژگی استخراج نمیکنیم و سپس از روی آن ویژگی هایی را انتخاب نمی کنیم به خاطر این است که ورودی تابع موجود در Sklearn برای انتخاب ویزگی نمی تواند اعداد منفی باشد درنتیجه بررسی این گام به طور کلی منتفی است

۱. **K = 12 و Component = 10**

Model / Measure	Accuracy	Precision	Recall	F-measure
SVM	79.03	80.81	79.03	78.73
Decision Tree	82.73	82.74	82.72	82.72
Gradient B	79.51	81.35	79.51	79.20
Random F	80.99	82.90	80.99	80.71

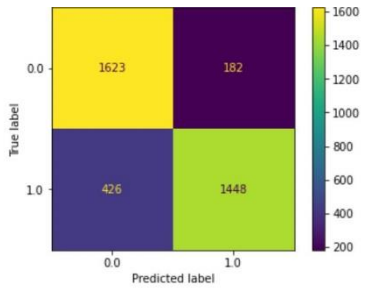
نتیجه گیری

- ✓ **باتوجه به نتایج بدست آمده از مراحل قبل و جداول بررسی شده، اگر از روش Random forest استفاده کنیم و بر روی آن تکنیک انتخاب ویژگی‌با تعداد ۱۰تا ویژگی منتخب را بررسی کنیم دقتی معادل نزدیک ۸۴ درصد را بدست خواهیم آورد.**
- ✓ **مدل انتخاب شده مدل Random Forest است**
- ✓ **روش مورد استفاده ، روش انتخاب ویژگی با تعداد ویژگی ۱۰ عدد است.**

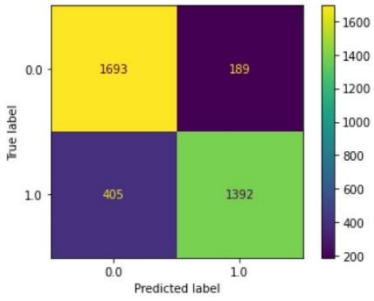
در مراحل بعد سعی می کنیم با تکنیک هایی دیگر دقت بدست آمده را بهبود دهیم.

مرحله پنجم، انتخاب بهترین کلاس بند و روش از بین روش های معرفی شده

از دسته بند انتخاب شده در مرحله ی قبل استفاده می کنیم و سعی می کنیم دقت بدست آمده که نزدیک به ۸۴ درصد می باشد را بهبود دهیم. دقت های بدست آمده از ماتریس Confusion است که در زیر برای فولد اول و دوم نتایج را مشاهده می کنید.




```
0.8347377004620821
(0.8402186312640911, 0.8359238685378277, 0.834362922469213, None)
```




```
0.8385430823593367
(0.8437072082746195, 0.8370996471270751, 0.837455037175219, None)
```

✓ روش اول مورد بررسی جهت افزایش دقت روش انتخاب شده

در این روش می خواهیم با اســـتفاده از تکنیک Normalization، مقادیر هر ویژگی برای داده ها را به بازه ای خاص منتقل کنیم. و خروجی را به ازای هر فولد جداگانه بدســـت آوریم. همانطور که از نتایج مشـــخص اســـت تغییری در نتایج بدست نیامده است بلکه دقت کمی کاهش هم یافته است!.

Fold / Measure	Accuray	Precision	Recall	F-measure
Fold 1	84.15	84.60	84.21	84.11
Fold 2	83.06	83.58	83.01	82.98
Fold 3	83.22	83.93	83.26	83.15
Fold 4	83.93	84.48	83.82	83.83
Fold 5	81.10	81.77	81.17	81.02
Overall	83.09	83.67	83.09	83.01

✓ روش دوم، تغییر مقدار داده های خروجی از تکنیک Undersampling

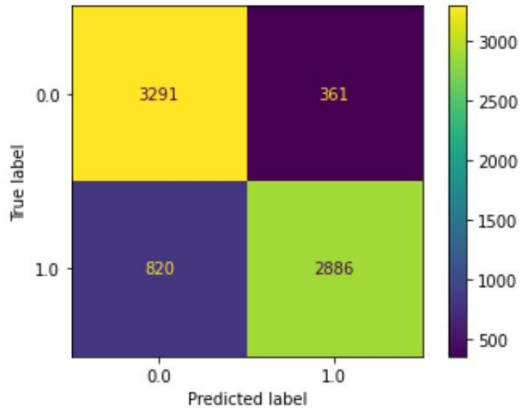
در این تکنیک با استفاده از پارامتر اولیه ی Sampling Strategy، تعداد داده های مورد استفاده برابر ۱۸۰۰۰ به صورت حدودی می باشد. در صورتی که اگر مقدار پارامتر را کاهش دهیم در گام بعدی زمانی که تکنیک Oversampling را اعمال کنیم، تعداد داده ها برای آموزش و تست افزایش می یابد. ازآنجا که افزایش داده ها سبب آموزش بیشتر و یادگیری بیشتر بر روی دیتاست های پیچیده می شود، درنتیجه ممکن است باعث بهبود دقت شود که در زیر به آن می پردازیم.

در جدول مورد بررسی در مرحله ی قبل، Smapling Strategy بر روی عدد ۰.۴ بود که منجر به این می شد درنهایت چیزی نزدیک به ۱۸۰۰۰ داده برای آموزش داشـــته باشـــیم. حال این مقدار را از ۰.۴ به ۰.۲ کاهش می دهیم تا نتایج زیر بدست آیند

لازم به ذکر است، از تکنیک Normalization در این روش نیز استفاده شده است.

Fold / Measure	Accuray	Precision	Recall	F-measure
Fold 1	83.94	84.46	83.99	83.90
Fold 2	84.35	84.94	84.39	84.30
Fold 3	84.03	84.59	84.09	83.98
Fold 4	83.16	83.68	83.07	83.06
Fold 5	83.93	84.24	83.88	83.88
Overall	83.88	84.38	83.88	83.82

که برای فولد اول دقت های نوشته در زیر آمده است




```
0.8394944278336505
(0.8446777996222323, 0.8399436188555429, 0.8390066766353925, None)
```

با کمی دقت متوجه می شویم که این تکنیک حتی باعث کاهش دقت بر روی بعضی از معیارها شده است.

که ظاهرا بازهم بهبودی حاصل نشده است.

در نهایت تنها کار باقی مانده افزایش ویژگی ها از ۱۶ به ۱۹ اســت که ویژگی های اضــافه شــده عبارت اند از Latitude ، Long و Mag (میزان ریشتر) است که موقعیت را در تاریخ مشخص شده نشان می دهد. بنظر ویژگی های اضـافه شـده می توانند مورد اسـتفاده قرار گیرند زیرا دانسـتن موقعیت زمین در زمان زلزله می تواند ویژگی خوبی برای پیش بینی وقوع زلزله باشد.

✓ روش سوم، افزایش ویژگی

در صورت اعمال این کار، دقت به حدود ۱۰۰ درصد همگرا خواهد شد. لذا دقت بدست آمده از طرفی خوشحال کننده و از طرفی دیگر نگران کننده اسـت. زیرا از طرفی مشـکل زلزله حل شـده اسـت و از طرفی دیگر ممکن ویژگی های انتخاب شده قدرت تفکیک پذیری زیاد داشته باشند و باکمی تغییر مشکلی در پیش بینی مدل در آینده به وجود می آورند. **لذا به دقت بدســت آمده اســتناد نکرده و خروجی نهایی را مانند جدول اولیه درنظر می گیریم.**

که البته وجود دقت نزدیک به ۸۴ درصد نیز خود دسـتیابی بزرگی در این زمینه اسـت که امید اسـت روش های بیان شده به اندازه ی کافی قانع کننده باشند.

مرحله ششم، نتیجه گیری نهایی

توانستیم با راهکارهای بیان شده و بررسی چندین مدل، بهترین مدل و روش را به طور تقریبی بدست آوریم که دقتی حدود ۸۴ درصد برای معیارهای خواسته شده برمیگرداند.

درنهایت دسته بند Random Forest با ماکزیمم عمق ۱۰، و استفاده از راهکارهای بیان شده جهت مقابله با مشکل بالانس نبودن داده ها و همچنین مشکلات دیگر، زلزله بالای ۴.۵ ریشتر را تاحدودی پیش بینی کنیم. همچنین توانستیم دقت نزدیک به ۱۰۰ درصد را درصورت استفاده از ویژگی های بیشتر بدست آوریم ولی به دلیل اینکه اثبات آن به لحاظ تئوری کار سختی است، به عنوان اهداف پژوهشی آینده به آن خواهیم پرداخت.

تمامی مراحل نوشته در فایلی با پسوند پایتون قرار گرفته شده اند.

پیشگویی بر روی داده ها

در قسمت پایانی قصد داریم، به عنوان نمونه ۴ داده را درون دیتاست قرار داده و پیشگویی کند که آیا زلزله بالای ۴.۵ ریشتر اتفاق افتاده یا خیر که در زیر خروجی آن را مشاهده می کنید

```
   Year  Month  Day  Hour  Minute    Lat   Long  Mag   R_mercury  \
0  1903      1    2     7      15  36.50  54.90  5.0  146.184579
1  1903      2    9     5      18  36.58  47.65  5.6  113.694438
2  1903      3   22    14      35  33.16  59.71  6.2  206.968385
3  1903      4   19     0       0  39.10  42.40  6.8  173.057907

   theta_mercury  ...   R_jupiter  theta_jupiter     R_saturn  theta_saturn  \
0      22.538839  ...  893.715168      36.522322  1576.060643     16.802514
1      20.057547  ...  926.699193       7.440769  1576.766205     15.930838
2       2.549143  ...  912.881976      24.002519  1520.210194     52.244931
3      18.935818  ...  876.425877      45.350483  1459.155612     77.217561

      R_uranus  theta_uranus     R_neptun  theta_neptun  R_moon  theta_moon
0  3003.936203     27.838936  4348.036574    169.254750   0.384   48.857174
1  2936.156174     63.488839  4395.485059    131.026133   0.384  113.750643
2  2833.733311    103.562852  4492.571119     90.014859   0.384  142.495056
3  2772.353765    130.884547  4560.133557     63.383312   0.384  164.446374

[4 rows x 24 columns]

array([1., 1., 1., 0.])
```

بر اساس ستون "Mag" که میزان ریشتر را دربردارد، باید هر ۴ تا خروجی برچسب یک بگیرند ولی دریکی از آن ها برچسب صفر داده ایم. که نسبتا دقت خوبی است. البته همیشه ممکن است خروجی درستی حاصل نشود که جای تامل دارد

از زحمات سرکارخانوم آقایی و آقای دکتر آبین در طول گذشته تشکر می نمایم.

باتشکر، حسین سیم چی، ۹۸۴۴۳۱۱۹

۱۹ بهمن ۱۳۹۹