به نام خدا

ارائه بهترین الگوریتم برای خوشه بندی دسته پرندگان در حال حرکت

استاد مربوطه:دکتر منصوره میرزایی

پریسا شفائی مهر

گزارش شماره یک :سمینار اول

در سمینار اول سعی شد به سوالات زیر پاسخ داده شود

کلان داده چیست؟

انواع دسته بندی های کلان داده هااز نظر ساختاری به چه شکل است؟

چرا کلان داده ها مهم هستند؟

منابع دسترسی به کلان داده ها چه چیزهایی هستند؟

کاربرد کلان داده چیست؟

آیا کلان داده خطرناک است؟

چالش هایی که درارتباط با کلان داده با آنها برخورد خواهیم داشت چه چیزهایی هستند؟

برای تحلیل کلان داده ها چه توانایی هایی لازم است؟

ابزار مورد استفاده در بحث کلان داده چه چیزهایی هستند؟

گزارش شماره دو :سمیناردوم

در این سمینار کارهای پیش پردازش داده و یادگیری مدل و ارزیابی و تفسیر به شرح زیر صورت گرفته است

توضیح کلی در مورد این پروژه:

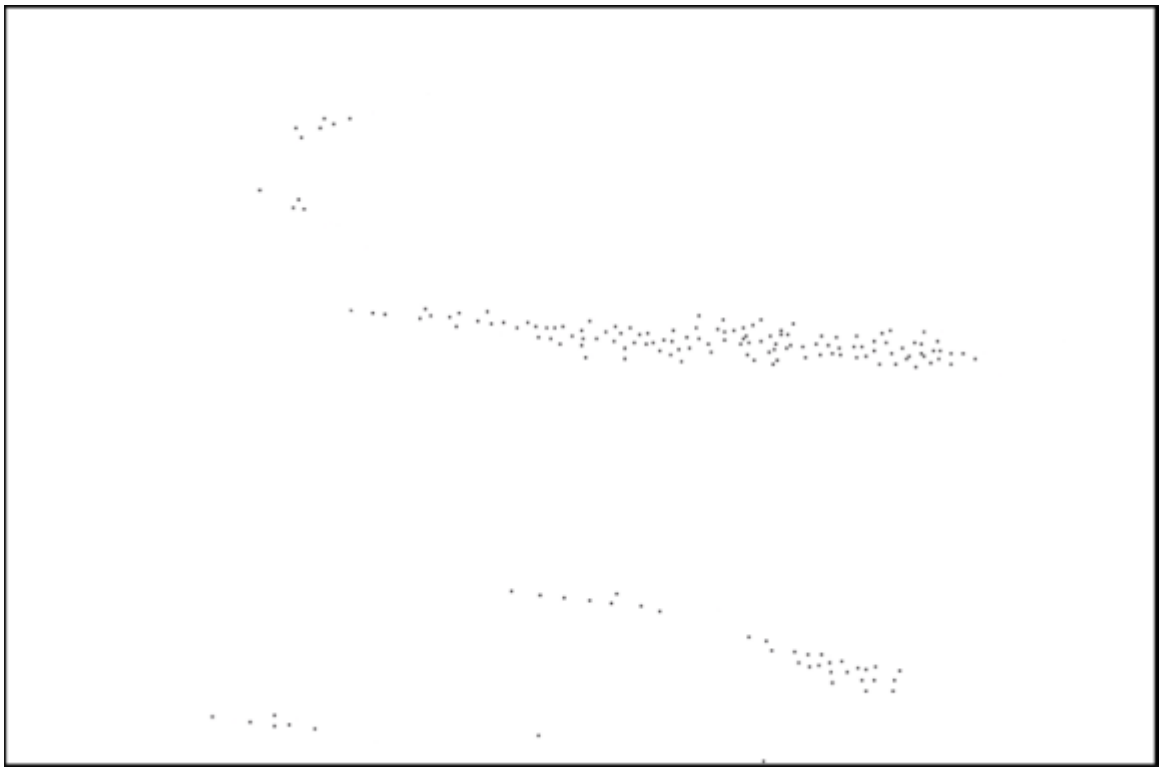
این مجموعه داده تشکیل شده از۲۴۰۰۰رکورد مربوط به دسته پرندگان در حال حرکت است که شامل ۲۰۰ دسته با ویژگی های شامل موقعیت و بردار سرعت و..... این پرندگان است که در مجموع شامل ۲۰۰۰ متغیرتوضیحی است .

انسان ها به راحتی دسته ای در حال حرکت و خوشه مربوط به آن در طبیعت را تشخیص می دهند، اگرچه گاهی اوقات توضیح دلیل آن برایشان سخت است. اما در این پروژه از دید یک دید یک ماشین به این مسئله پرداخته نگاهی کاوشگرانه به کلاس بندی حرکت این پرندگان خواهیم داشت.

هدف ما از این پروژه آموزش مدل برای کلاس های مربوطه و سپس آزمایش و ارزیابی مدل است .

میخواهیم بااستفاده از الگوریتم های متفاوت دقت مدل را در داده آموزشی بررسی کرده کارایی و دقت را افزایش داده و با دانش های کسب شده پیش بینی مناسبی برای ورودی های جدید انجام دهیم.

تصویر زیر شمایی کلی از حرکت منسجم پرندگان دردسته های مشخص را نشان می دهد



توضیحات داده:

ویژگی ها

xm، ym به عنوان موقعیت (X,Y) هر بخش نوع متغیر real ،

xVeln، yVeln به عنوان بردار سرعت نوع متغیر real،

xAm، yAm به عنوان بردار ترازنوع متغیر real،

xSm، ySm به عنوان بردار جدایی نوع متغیر real،

xCm، yCm به عنوان بردار پیوستگی نوع متغیر real،

nACm به عنوان تعداد boid در شعاع Alignment/Cohesion نوع متغیر integerو

nSm به عنوان تعداد Boidها در شعاع جداسازی نوع متغیر integer.

این ویژگی ها برای همه m boid ها، جایی که m=1,...,200 تکرار می شوند.

همچنین برچسب های کلاس باینری هستند که ۱ به grouped و ۰ به ، non grouped اشاره دارد.

Boid (دسته ای از زنبور ها پرندگان یا حشرات که به طور دسته جمعی حرکت میکنند)

پیش پردازش داده ها :

مرحله اول: داده هاکه تعدادی بالغ بر ۲۴۰۰۰ رکورد بود را به دو قسمت برای آموزش مدل(۱۶۰۰۰) تست مدل(۸۰۰۰) تقسیم کردم . ۷۰٪آموزش ۳۰٪تست

مرحله دوم: برای اجرای سریعتر مدل و محدود سازی مواجهه با خطا در مدل از داده ها نمونه گیری میکنیم در اینجا من ازعملگر sample(stratified)استفاده کردم

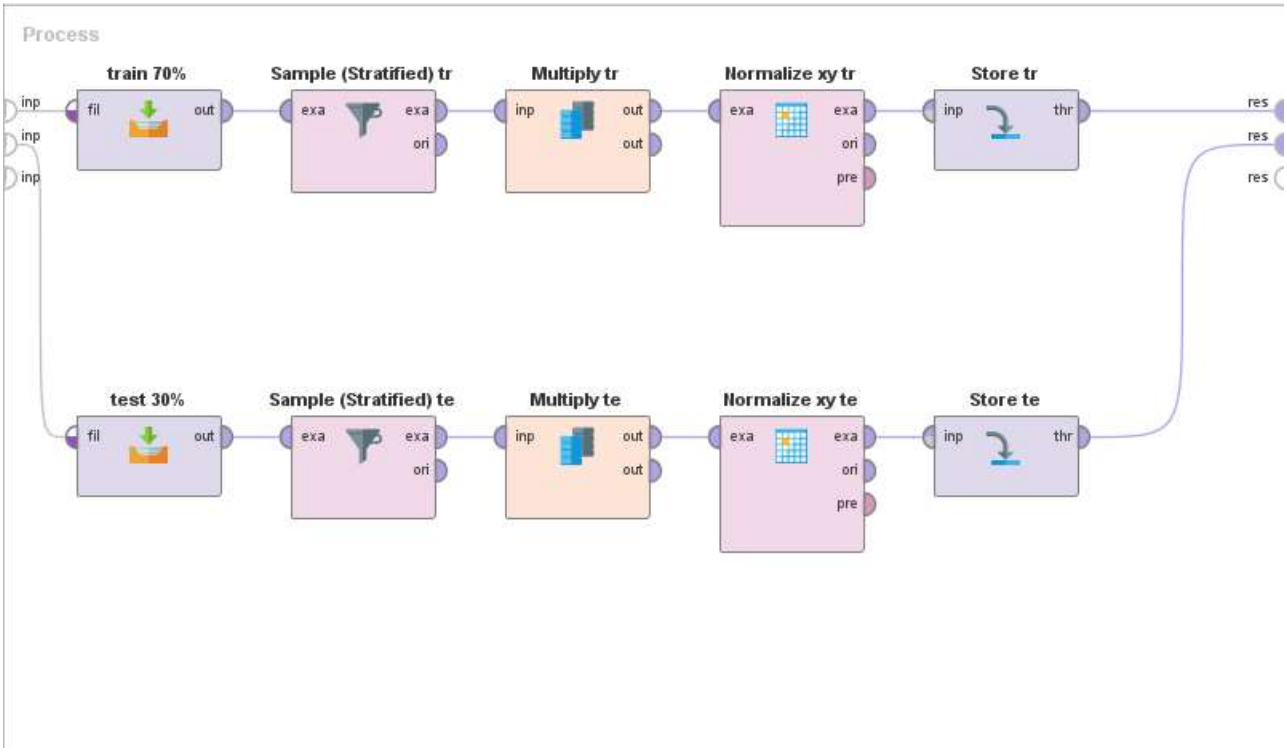
#ازrelative sample با نرخ ۰,۱استفاده شده

دلیل انتخاب عملگر sample(stratified): این روش نمونه گیری این را اجازه می دهد تا به سرعت جامعه نمونه ای را به دست آوریم که به بهترین نحو کل جمعیت مورد مطالعه را نشان می دهد.

مرحله سوم:از یک عملگر به نام multiply استفاده میکنیم تا بتوانیم دو دسته از متغیر ها را نرمال سازی کنیم. این کار را برای داده های آموزشو آزمایش و ارزیابی انجام میدهیم.

مرحله چهارم :داده های مربوط به متغیر های xm، ym را از طریق method range transformation به بازه ای بین -۱۰۰تا ۱۰۰نرمال سازی میکنیم تا دامنه تغییرات داده ها را به بازه ای مطلوب تر تبدیل کنیم.

مرحله پنجم:از عملگرstore برای هر سه مجموعه داده استفاده میکنیم تا به عنوان مخزن در فرآیند های بعدی پردازش داده آن ها را فراخوانی کنیم.

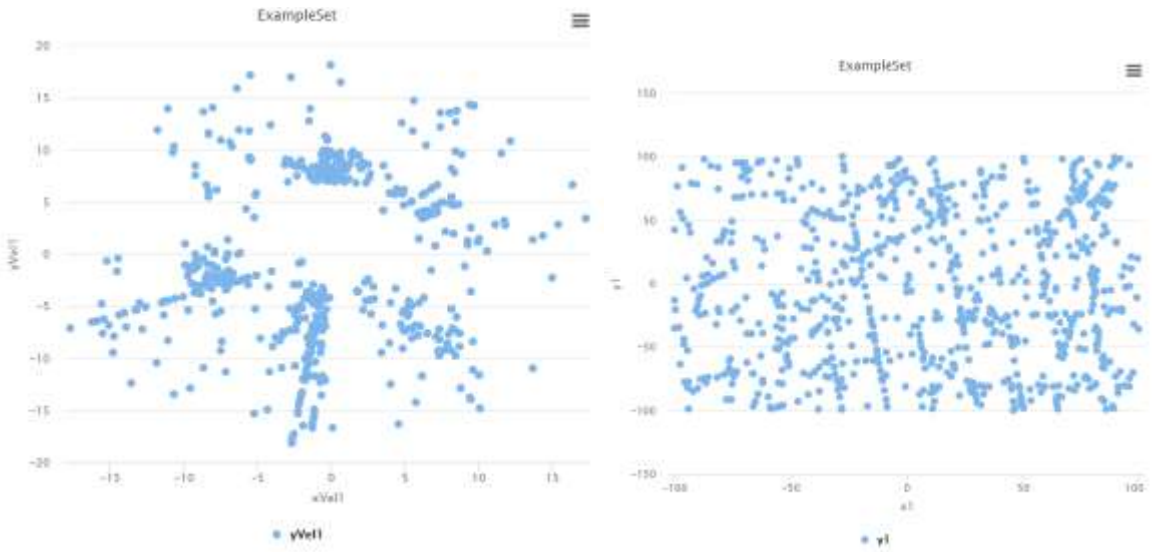


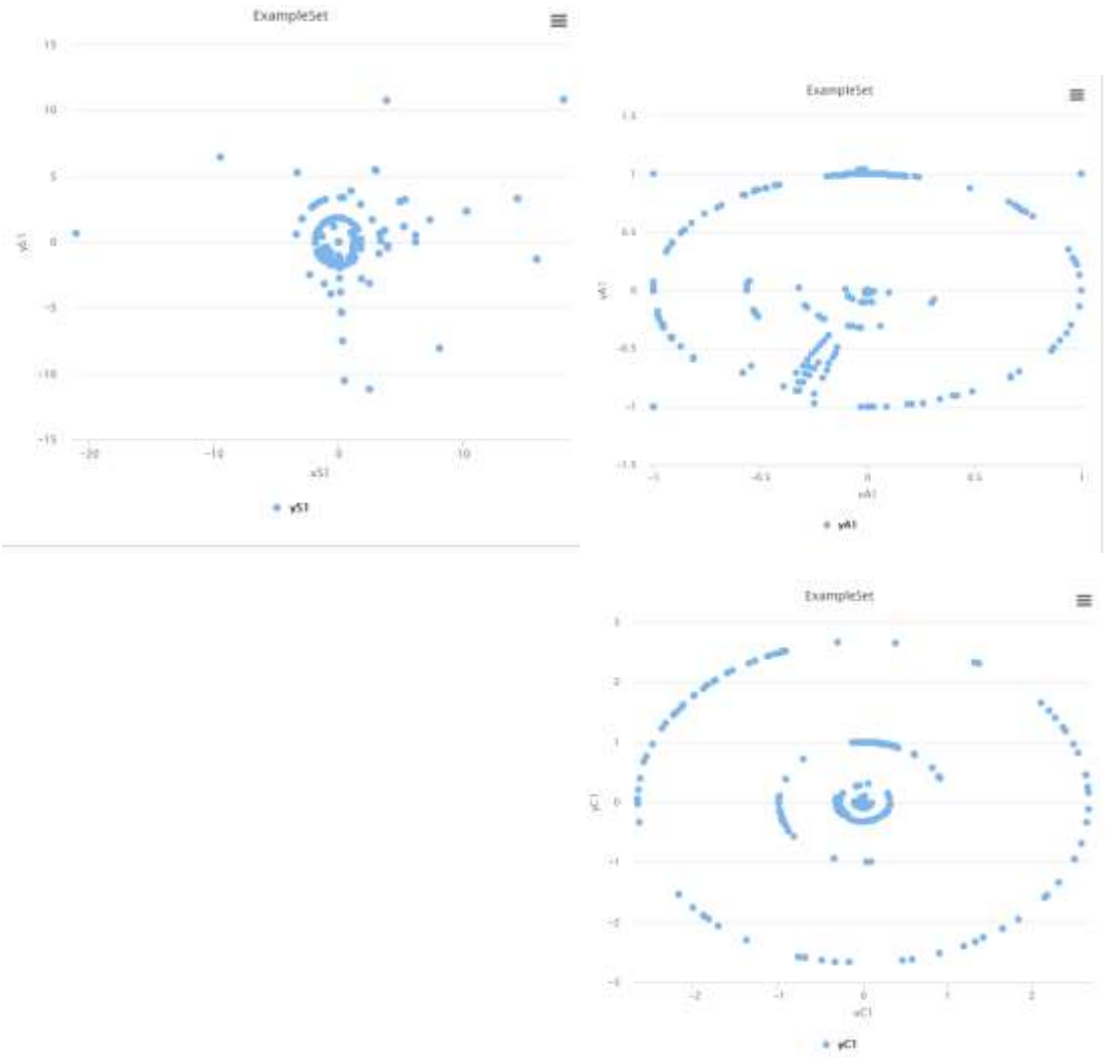
ExampleSet (Normalize xy te) | **ExampleSet (Normalize xy tr)**

Open in Turbo Prep | Auto Model | Filter (1,600 / 1,600 examples): all

Row No.	x1	y1	x2	y2	x3	y3	x4
1	-100	-52.672	-41.970	-11.045	59.443	80.977	-73.194
2	-99.313	-74.957	-92.669	-73.161	-39.814	27.946	-3.447
3	-98.580	93.880	-49.428	-51.465	-76.858	-47.212	-96.339
4	-98.442	-54.029	-68.628	45.448	50.617	57.867	-1.455
5	-98.056	94.098	-49.416	-49.525	-76.657	-45.548	-96.239
6	-98.052	-23.473	-93.455	-16.981	-40.216	-6.407	-44.720
7	-97.899	-60.934	-19.274	-51.545	58.000	79.517	92.849
8	-97.830	-85.489	22.799	-35.658	16.101	-20.422	-47.032
9	-97.726	-53.950	-24.886	-86.628	-6.258	-4.014	63.453
10	-97.684	72.942	-79.883	4.960	35.255	-56.468	-8.932
11	-97.654	-59.216	-17.765	-48.769	59.407	80.619	93.132
12	-97.490	-75.824	84.879	-22.269	-52.061	2.761	-4.422
13	-97.463	-57.883	-16.554	-46.540	60.557	81.520	93.281
14	-97.235	-68.213	-68.310	31.615	51.765	44.261	-2.685

Scatter plotمربوط به متغیر های مربوط به نمونه اول آورده شده است تا با تصویر سازی فهم بهتری از پراکندگی داده ها داشته باشیم.





مرحله ششم:از یک عملگر به نام retriveاستفاده میکنیم تا انباره ذخیره شده در عملگر storeبازیابی شود

مرحله هفتم : نوع ویژگی متغیرclass را که در مراحل بعدی پردازش میخواهیم به عنوان برچسب از آن برای انجام فرآیند استفاده کنیم از integer به bionaminalتغییر میدهیم.

مرحله هشتم : از عملگری به نام set roleاستفاده میکنیم تا به متغیر classبرچسبی تحت عنوان label بدهد تا مشخص کننده این امر باشد که داده ها بر چه اساس مدلسازی و پیش بینی خواهند شد

یادگیری مدل:

مرحله اول : در این مرحله از انواع الگوریتم های درخت تصمیم و روشهای خوشه بندی استفاده شده که بهترین مدل برای داده ها ساخته شود و سپس برای داده های جدید که وارد مدل میشوند پیش بینی با باالاترین دقت انجام گیرد

#انواع این الگوریتم ها بررسی دلایل انتخاب برای این داده ها و در آخر گفته شده کدام الگوریتم بالاترین دقت را برای این داده ها می دهد و بهترین گزینه برای انتخاب است.

مرحله دوم:از عملگر apply model برای اعمال مدل روی داده ها استفاده میکنیم این عملگر مدل ساخته شده را همراه با یک مجموعه داده دریافت میکند تا داده را روی مدل اعمال کند طی این فرآیند هر رکورد از داده ورودی مورد بررسی قرار میگیرد و یک برچسب به هر رکورد نسبت داده می شود.

۱) decision tree

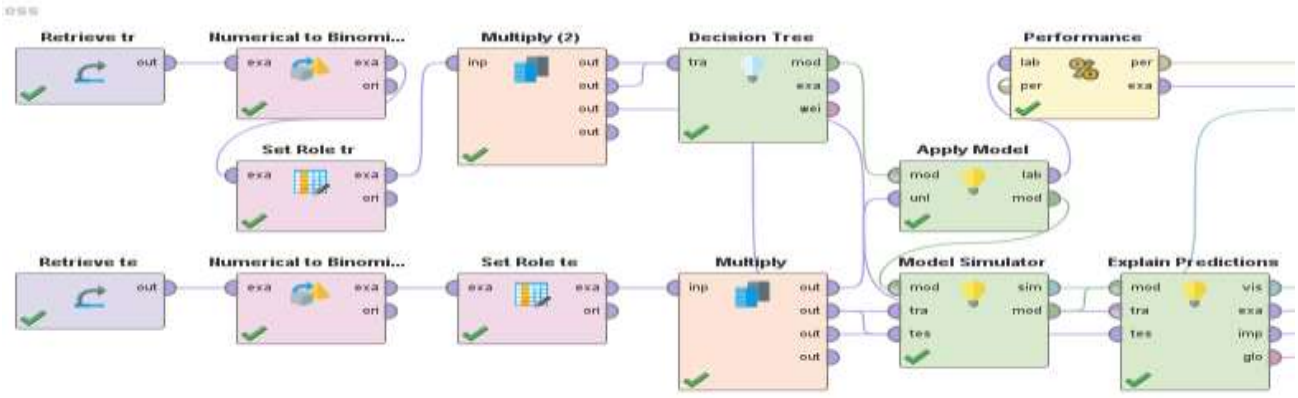
این درخت تصمیم مجموعه داده مارا بر اساس برچسب class مدلسازی و برای داده های جدید پیش بینی ارائه خواهد کرد.

دقت پیش بینی مدل برای داده های تست بااین الگوریتم ۹۵٪است که نشان دهنده آن است که الگوریتم تقریبا مناسبی برای مجموعه داده است.

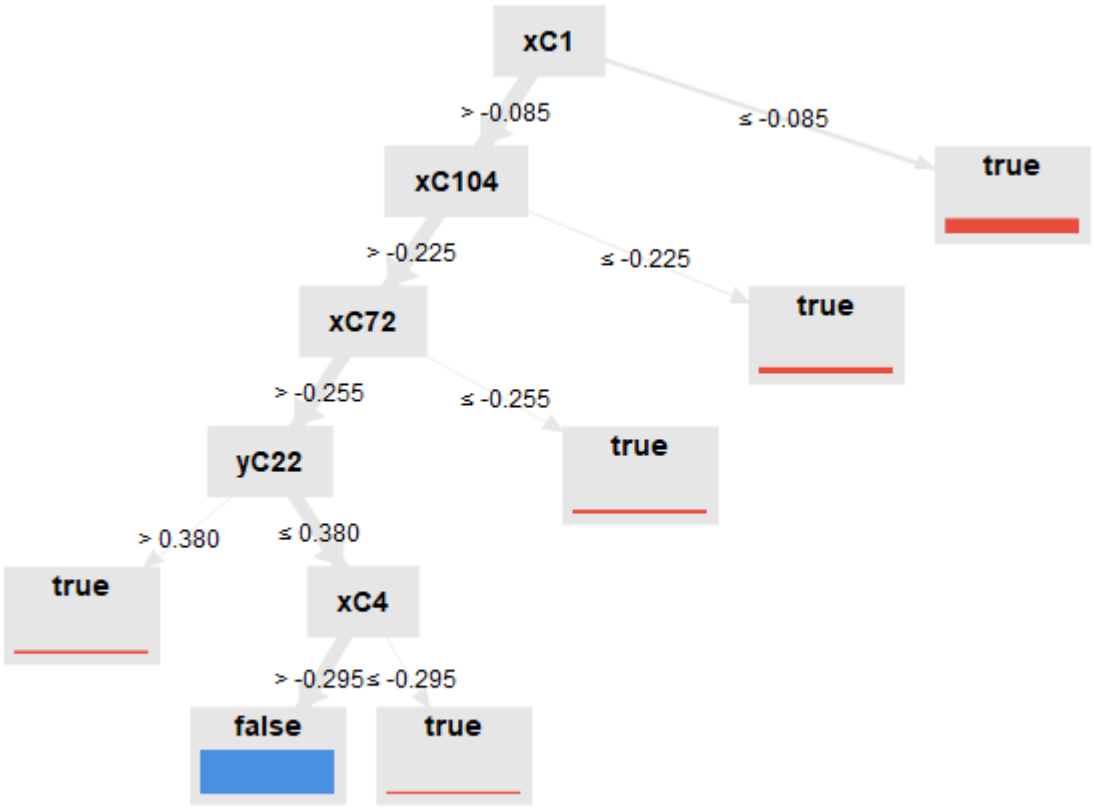
از عملگر perfomance classification برای ارزیابی دقت مدل استفاده کردیم

از یک عملگر به نام model simulatorبرای شبیه سازی مدل و یک عملگر به نام explain predictionبرای فهم بهتر مدل شبیه سازی شده استفاده میکنیم

بزرگترین support برای این تصمیم xC104 است ۹۵,۱۳٪ از تمام پیش بینی های انجام شده توسط این مدل درست است. وقتی مدل falseاست ، ۸۸,۸۲ درصد از این موارد را پوشش می دهد. و با ۹۹,۳۴٪ از تمام پیش بینی ها برای کلاس false درست است.

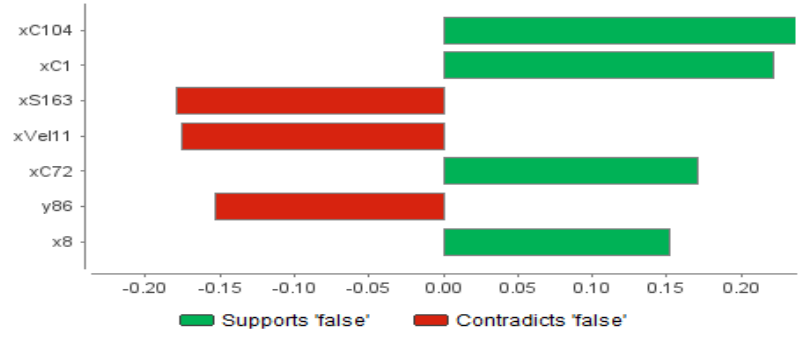


Ro...	Class	prediction(Class)	confidence(false)	confidence(tru...	x1	y1	x2	y2
1	false	false	0.999	0.001	9.720	45.385	-16.401	49.7
2	false	false	0.999	0.001	9.880	94.774	-83.558	-53.
3	false	false	0.999	0.001	10.121	54.628	-82.249	-60.
4	true	true	0	1	10.197	-86.601	74.284	-64.
5	true	false	0.999	0.001	10.406	73.393	17.985	6.12
6	true	true	0	1	10.631	34.063	72.530	69.7
7	false	false	0.999	0.001	10.661	70.537	66.990	1.01
8	false	false	0.999	0.001	10.674	-62.954	-90.367	-46.
9	true	true	0.003	0.997	11.040	-58.559	63.197	-45.
10	true	true	0	1	11.105	-98.444	94.404	8.80
11	false	false	0.999	0.001	11.132	14.311	46.845	71.0
12	false	false	0.999	0.001	11.205	-81.034	77.813	4.49
13	false	false	0.999	0.001	11.292	50.086	-3.973	94.7
14	false	false	0.999	0.001	11.330	71.264	67.520	1.62
15	true	true	0	1	11.452	98.385	94.545	5.55



Prediction: false

Important Factors for false



Tree

```
xC1 > -0.085
|   xC104 > -0.225
|   |   xC72 > -0.255
|   |   |   yC22 > 0.380: true {false=0, true=26}
|   |   |   yC22 ≤ 0.380
|   |   |   |   xC4 > -0.295: false {false=1079, true=1}
|   |   |   |   xC4 ≤ -0.295: true {false=0, true=7}
|   |   xC72 ≤ -0.255: true {false=0, true=46}
|   xC104 ≤ -0.225: true {false=0, true=109}
xC1 ≤ -0.085: true {false=1, true=331}
```

accuracy: 95.13%

	true false	true true	class precis
pred. false	151	1	99.34%
pred. true	19	240	92.66%
class recall	88.82%	99.59%	

Support Prediction	Contradict Prediction
xC1 = 0.130 (0.229); yVel118 = -5.060 (0.174); xC4 = 0 (0.152)	yC22 = 0.300 (-0.328); xS158 = 0 (-0.161); yVel199 = 2.290 (-0.1...
xC1 = 0 (0.314); xC104 = 0 (0.202); xC4 = 0 (0.175)	yC22 = 0 (-0.181); yVel199 = -8.770 (-0.148); nS181 = 0 (-0.131)
xC1 = 0 (0.314); xC104 = 0 (0.202); xC4 = 0 (0.175)	yC22 = 0 (-0.181); yVel199 = -9.120 (-0.148); nS181 = 0 (-0.131)
yC64 = -2.450 (0.000); nAC145 = 46 (0.000); yVel49 = -10.680 (0.000)	nS102 = 2 (-0.000); y17 = 66.794 (-0.000); yS47 = 0.030 (-0.000)
xC104 = 0.520 (0.530); xVel72 = 9.340 (0.140); xS37 = -3.290 (0.135)	x16 = -64.296 (-0.131); xS117 = -0.160 (-0.130); xC82 = -1.930 (-...
xC1 = 1.320 (0.147); yS75 = -0.780 (0.141); xVel76 = 17.190 (0.140)	x142 = 92.309 (-0.149); yC100 = -1.800 (-0.144); xC133 = 2.480 ...
xC1 = 0 (0.314); xC104 = 0 (0.202); xC4 = 0 (0.175)	yC22 = 0 (-0.181); yVel199 = -6.210 (-0.148); nS181 = 0 (-0.131)
xC1 = 0.080 (0.370); xC104 = 0.040 (0.264); nS9 = 0 (0.168)	yC22 = -0.030 (-0.187); xA155 = 0 (-0.146); yC73 = 0.050 (-0.146)
yC64 = 0.860 (0.000); nAC145 = 84 (0.000); yVel49 = 8.550 (0.000)	nS102 = 0 (-0.000); y17 = -13.220 (-0.000); yS47 = -1.410 (-0.000)
yC22 = 1 (0.266); xC53 = 0.010 (0.161); xA104 = -0.020 (0.144)	xA109 = -0.020 (-0.133); yC198 = -1 (-0.133); xVel46 = 0.030 (-0....
xC1 = 0.090 (0.249); xC104 = 0.010 (0.192); x182 = 0.420 (0.145)	yC22 = 0.310 (-0.303); nAC52 = 1 (-0.137); yVel199 = 2.930 (-0.1...
xC1 = -0.020 (0.294); xC104 = 0.060 (0.180); xC4 = -0.090 (0.175)	yC22 = 0.070 (-0.191); y18 = -35.133 (-0.136); yVel199 = -6.560 (...

2)random forest

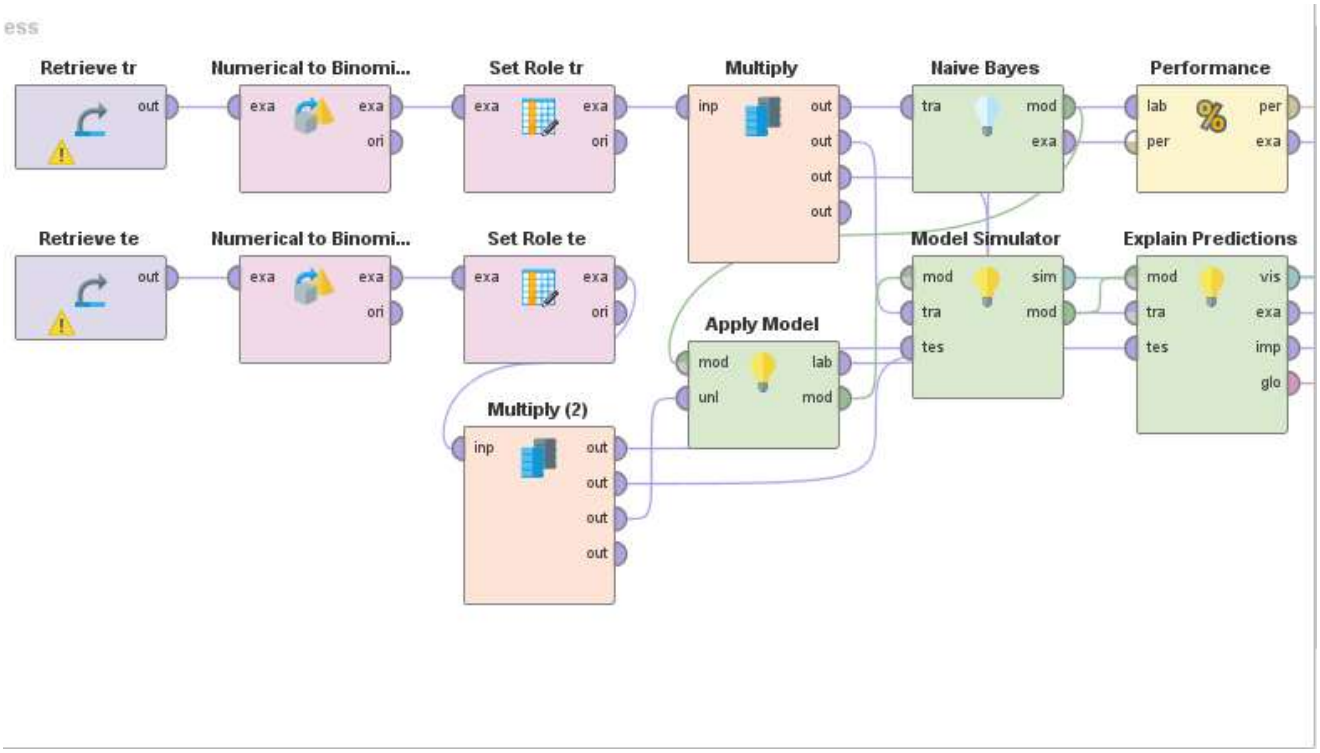
این درخت تصمیم مجموعه داده مارا بر اساس برچسب class مدلسازی و برای داده های جدید پیش بینی ارائه خواهد کرد..

دقت پیش بینی مدل برای داده های تست بااین الگوریتم 100% است که نشان دهنده آن است که الگوریتم خیلی مناسبی برای مجموعه داده است.

از عملگر perfomance classification برای ارزیابی دقت مدل استفاده کردیم

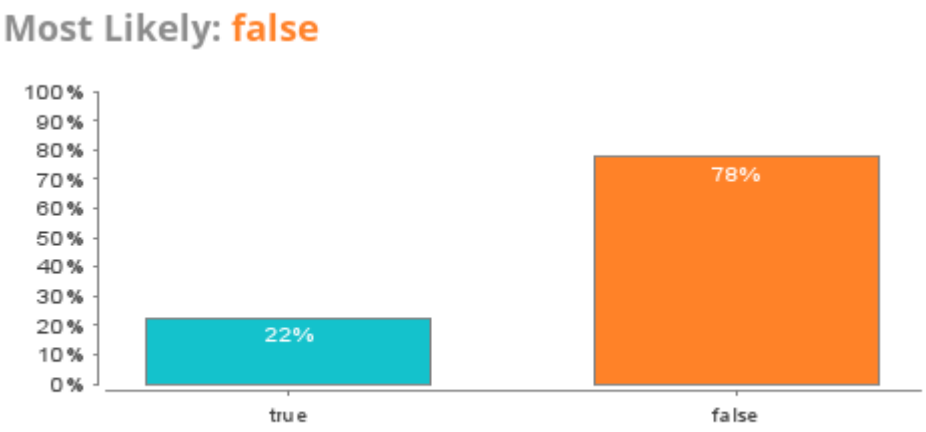
از یک عملگر به نام model simulator برای شبیه سازی مدل و یک عملگر به نام explain prediction برای فهم بهتر مدل شبیه سازی شده استفاده میکنیم

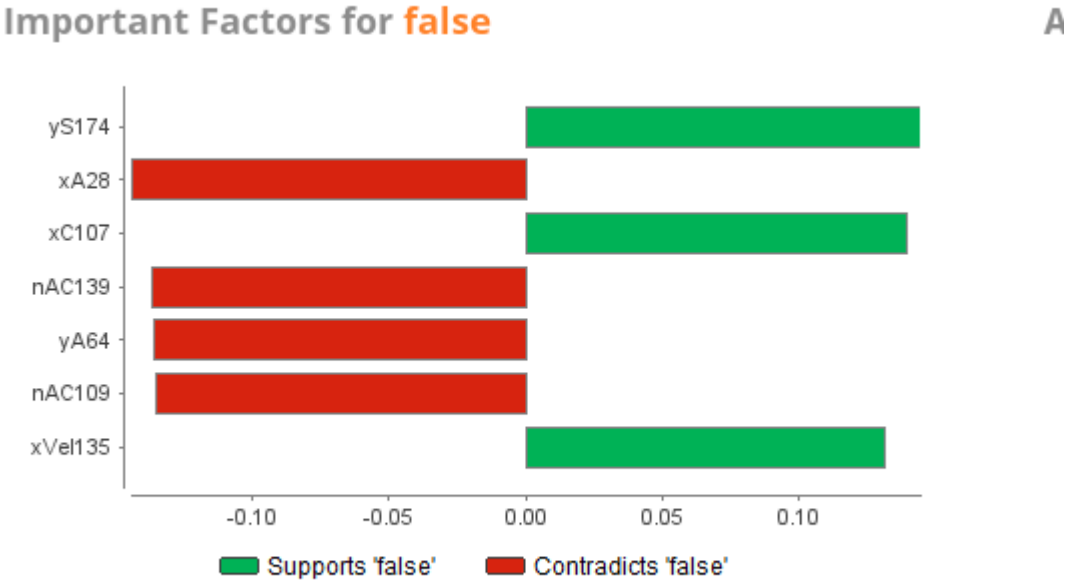
بزرگترین support برای این تصمیم از yS174 است. ۱۰۰,۰۰٪ از تمام پیش بینی های انجام شده توسط این مدل درست است. وقتی مدل falseاست، ۱۰۰,۰۰٪ موارد را پوشش می‌دهد. و با ۱۰۰,۰۰٪ از همه پیش بینی ها برای کلاس false درست است.

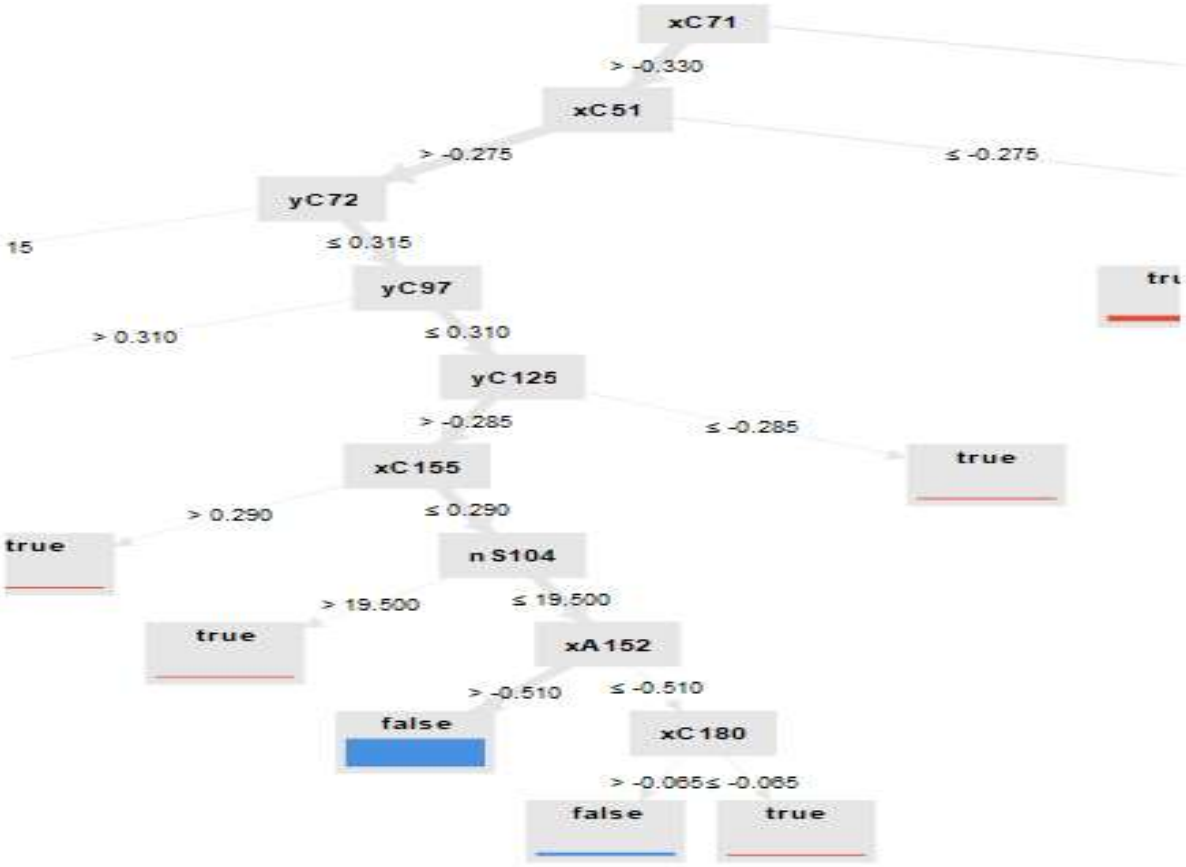


accuracy: 100.00%

	true false	true true	class preci
pred. false	170	0	100.00%
pred. true	0	241	100.00%
class recall	100.00%	100.00%	







Tree

```
x158 > -98.807
|   yC141 > 0.085
|   |   yVel100 > -12.055: true (false=0, true=252)
|   |   yVel100 ≤ -12.055
|   |   |   x48 > 11.164: true (false=0, true=3)
|   |   |   x48 ≤ 11.164: false (false=5, true=0)
|   yC141 ≤ 0.085
|   |   xC164 > -0.340
|   |   |   xC32 > -0.225
|   |   |   |   xC67 > -0.295
|   |   |   |   |   xC94 > 0.405: true (false=0, true=7)
|   |   |   |   |   xC94 ≤ 0.405
|   |   |   |   |   |   xC34 > -0.300
|   |   |   |   |   |   |   yC91 > 0.590: true (false=0, true=5)
|   |   |   |   |   |   |   yC91 ≤ 0.590
|   |   |   |   |   |   |   |   nS1 > 4: true (false=0, true=1)
|   |   |   |   |   |   |   |   nS1 ≤ 4: false (false=1097, true=4)
|   |   |   |   |   |   xC34 ≤ -0.300: true (false=0, true=2)
|   |   |   |   xC67 ≤ -0.295: true (false=0, true=33)
|   |   |   xC32 ≤ -0.225
|   |   |   |   yVel85 > -10.760: true (false=0, true=79)
|   |   |   |   yVel85 ≤ -10.760
|   |   |   |   |   nS12 > 1: true (false=0, true=1)
|   |   |   |   |   nS12 ≤ 1: false (false=2, true=0)
|   |   xC164 ≤ -0.340: true (false=0, true=105)
x158 ≤ -98.807: true (false=0, true=4)
```

Row No.	Class	prediction(C...	confidence(f...	confidence(true)	Support Prediction	Contradict Predicti...
1	false	false	0.965	0.035	xC115 = 0.200 (0.18...	x129 = -78.423 (-0.1...
2	false	false	0.961	0.039	y164 = 24.501 (0.133...	y31 = 14.612 (-0.15...
3	false	false	0.971	0.029	xC15 = 0 (0.135); yA1...	nS78 = 0 (-0.155); y...
4	true	true	0.090	0.910	x33 = -92.450 (0.179...	yC115 = 0.020 (-0.3...
5	true	true	0.040	0.960	y91 = -13.101 (0.198...	xS152 = -64.760 (-0....
6	true	true	0.050	0.950	nS133 = 3 (0.197); y...	x2 = 72.530 (-0.248)...
7	false	false	0.996	0.004	yC15 = 0 (0.138); xVe...	yA27 = 0 (-0.148); y...
8	false	false	0.994	0.006	xC115 = 0.280 (0.16...	yC92 = 0.310 (-0.15...
9	true	true	0.080	0.920	xC90 = -0.540 (0.257...	x107 = 67.197 (-0.2...
10	true	true	0.020	0.980	xC30 = -0.020 (0.222...	nAC122 = 8 (-0.187)...
11	false	false	0.974	0.026	nS38 = 0 (0.138); xC...	nS23 = 0 (-0.141); x...
12	false	false	0.965	0.035	xC115 = 0 (0.205); n...	yVel29 = -7.440 (-0....

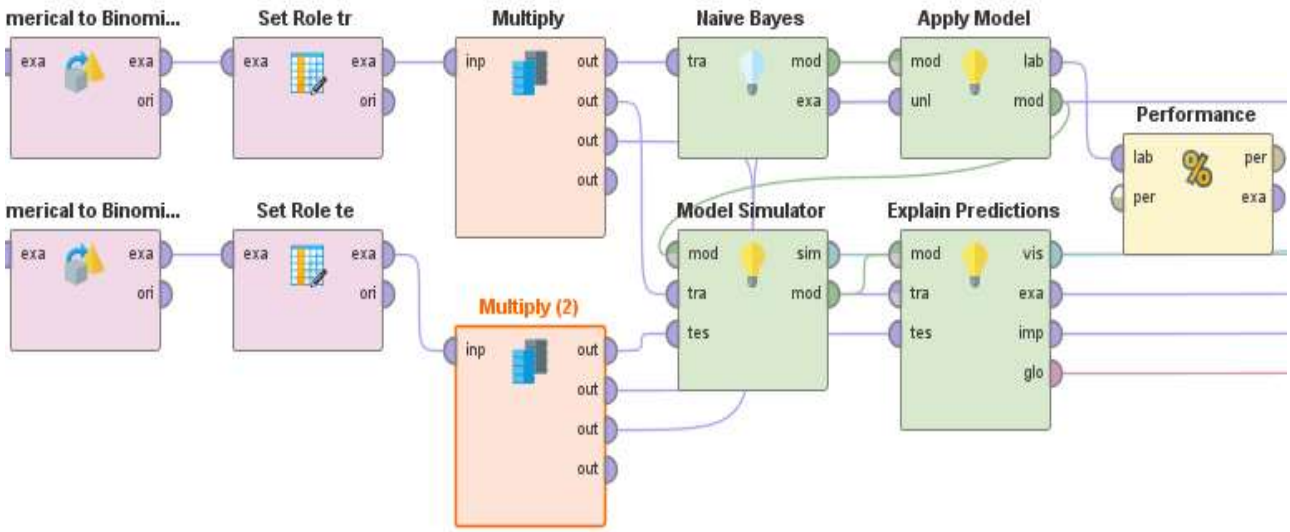
Row No.	Class	prediction(Class)	confidence(false)	confidence(true)	x1	y1	x2
1	false	false	0.965	0.035	9.720	45.385	-16.401
2	false	false	0.961	0.039	9.880	94.774	-83.558
3	false	false	0.971	0.029	10.121	54.628	-82.249
4	true	true	0.090	0.910	10.197	-86.601	74.284
5	true	true	0.040	0.960	10.406	73.393	17.985
6	true	true	0.050	0.950	10.631	34.063	72.530
7	false	false	0.996	0.004	10.661	70.537	66.990
8	false	false	0.994	0.006	10.674	-62.954	-90.367
9	true	true	0.080	0.920	11.040	-58.559	63.197
10	true	true	0.020	0.980	11.105	-98.444	94.404
11	false	false	0.974	0.026	11.132	14.311	46.845
12	false	false	0.965	0.035	11.205	-81.034	77.813
13	false	false	0.816	0.184	11.292	50.086	-3.973
14	false	false	0.996	0.004	11.330	71.264	67.520
15	true	true	0	1	11.452	98.385	94.545

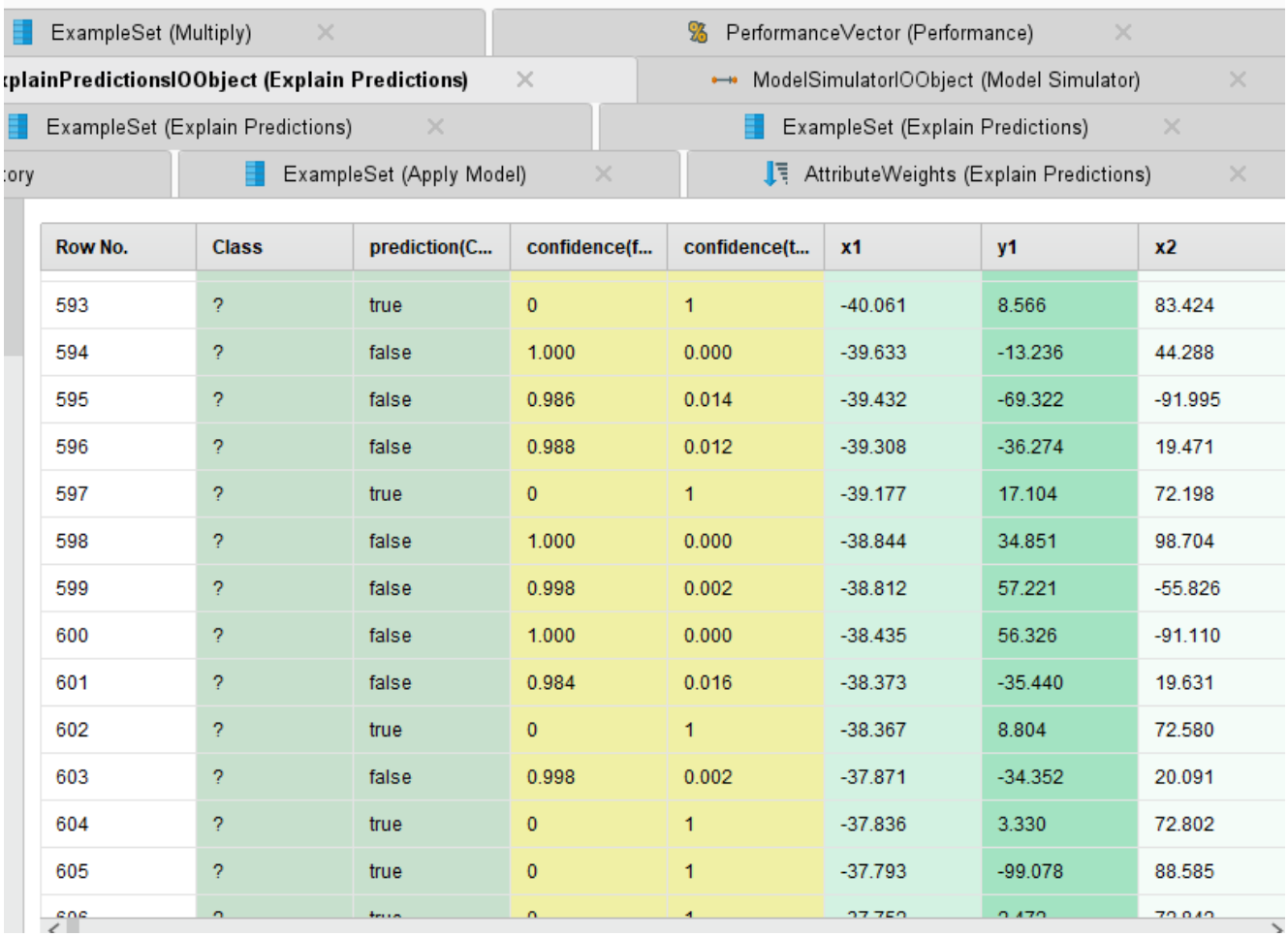
naïve bayes(3

این الگوریتم براساس نظریه بیز مدلی برای پیش بینی آینده در اختیار ما قرار میدهد دقت پیش بینی مدل برای دیتا ست ما ۹۹٪ درصد است که دقت بسیار خوبی است

از عملگر perfomance classification برای ارزیابی دقت مدل استفاده کردیم

مدل فوق العاده مطمئن است که پیش بینی صحیح false است. اطمینان برای این تصمیم با ۱۰۰,۰۰٪ بالا است. بزرگترین پشتیبانی برای این تصمیم از xA187 است. لطفاً به خاطر داشته باشید که ۹۹,۰۳٪ از تمام پیش بینی های انجام شده توسط این مدل درست است. وقتی مدل falseاست ، ۱۰۰,۰۰٪ موارد را پوشش می‌دهد. و با ۹۷,۷۰٪ از تمام پیش بینی ها برای کلاس false صحیح است.





Row No.	Class	prediction(C...	confidence(f...	confidence(t...	x1	y1	x2
593	?	true	0	1	-40.061	8.566	83.424
594	?	false	1.000	0.000	-39.633	-13.236	44.288
595	?	false	0.986	0.014	-39.432	-69.322	-91.995
596	?	false	0.988	0.012	-39.308	-36.274	19.471
597	?	true	0	1	-39.177	17.104	72.198
598	?	false	1.000	0.000	-38.844	34.851	98.704
599	?	false	0.998	0.002	-38.812	57.221	-55.826
600	?	false	1.000	0.000	-38.435	56.326	-91.110
601	?	false	0.984	0.016	-38.373	-35.440	19.631
602	?	true	0	1	-38.367	8.804	72.580
603	?	false	0.998	0.002	-37.871	-34.352	20.091
604	?	true	0	1	-37.836	3.330	72.802
605	?	true	0	1	-37.793	-99.078	88.585

Row No.	Class	prediction(Class)	confidence(false)	confidence(true)	x1	y1	x2
1	false	false	0.995	0.005	9.720	45.385	-16.401
2	false	false	1	0	9.880	94.774	-83.558
3	false	false	1	0	10.121	54.628	-82.249
4	true	true	0	1	10.197	-86.601	74.284
5	true	true	0	1	10.406	73.393	17.985
6	true	true	0	1	10.631	34.063	72.530
7	false	false	1	0	10.661	70.537	66.990
8	false	false	1	0	10.674	-62.954	-90.367
9	true	true	0	1	11.040	-58.559	63.197
10	true	true	0	1	11.105	-98.444	94.404
11	false	false	1	0	11.132	14.311	46.845
12	false	false	1	0	11.205	-81.034	77.813
13	false	false	1	0	11.292	50.086	-3.973
14	false	false	1	0	11.330	71.264	67.520
15	true	true	0	1	11.452	98.385	94.545

Open in Turbo Prep Auto Model Filter (1,600 / 1,600 examples): all

Row No.	Class	x1	y1	x2	y2	x3	y3
1	false	-100	-52.672	-41.970	-11.045	59.443	80.977
2	true	-99.313	-74.957	-92.669	-73.161	-39.814	27.946
3	false	-98.580	93.880	-49.428	-51.465	-76.858	-47.212
4	true	-98.442	-54.029	-68.628	45.448	50.617	57.867
5	false	-98.056	94.098	-49.416	-49.525	-76.657	-45.548
6	true	-98.052	-23.473	-93.455	-16.981	-40.216	-6.407
7	false	-97.899	-60.934	-19.274	-51.545	58.000	79.517
8	false	-97.830	-85.489	22.799	-35.658	16.101	-20.422
9	false	-97.726	-53.950	-24.886	-86.628	-6.258	-4.014
10	false	-97.684	72.942	-79.883	4.960	35.255	-56.468
11	false	-97.654	-59.216	-17.765	-48.769	59.407	80.619
12	false	-97.490	-75.824	84.879	-22.269	-52.061	2.761
13	false	-97.463	-57.883	-16.554	-46.540	60.557	81.520
14	true	-97.235	-68.213	-68.310	31.615	51.765	44.261

accuracy: 99.03%

	true false	true true	class precis
pred. false	170	4	97.70%
pred. true	0	237	100.00%
class recall	100.00%	98.34%	

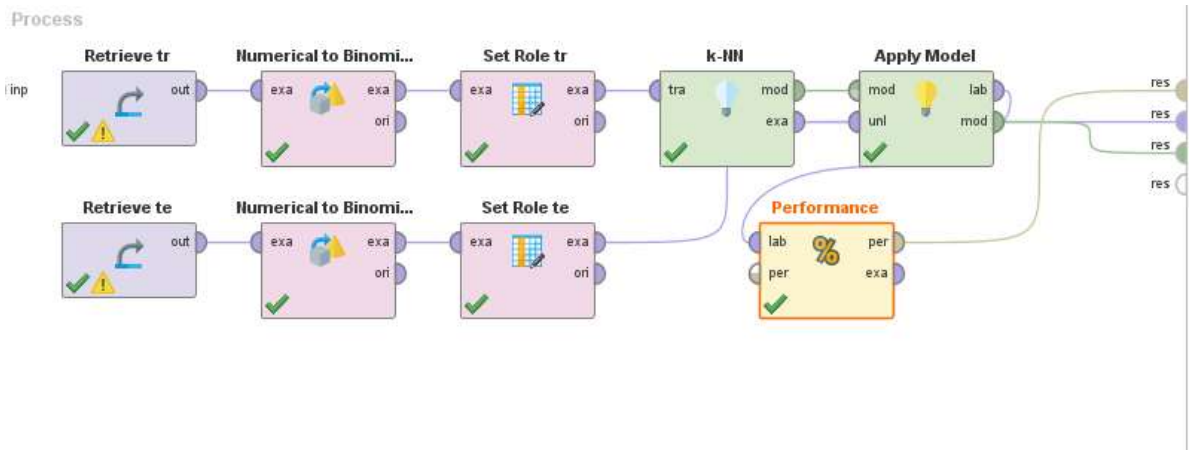
Attribute	Parameter	false	true
x1	mean	5.586	-14.987
x1	standard deviation	58.626	55.594
y1	mean	-0.114	-24.697
y1	standard deviation	58.009	50.145
x2	mean	-6.470	-36.716
x2	standard deviation	52.164	40.628
y2	mean	0.562	5.276
y2	standard deviation	55.955	46.993
x3	mean	-0.714	-17.904
x3	standard deviation	56.285	60.786
y3	mean	-3.969	13.812
y3	standard deviation	57.312	47.980
x4	mean	-11.203	-14.802
x4	standard deviation	54.249	40.650

4)knn classification

این الگوریتم k تا از نزدیک ترین همسایه ها را در یک دسته قرار می دهد

دقت این مدل برای داده های ما بسیار پایین تر از سایر الگوریتم های بررسی شده در این گزارش و پروژه است(دقت ۶۶ %)

از عملگر perfomance classification برای ارزیابی دقت مدل استفاده کردیم



Open in Turbo Prep Auto Model Filter (1,600 / 1,600 examples): all

Row No.	Class	x1	y1	x2	y2	x3	y3
1	false	-100	-52.672	-41.970	-11.045	59.443	80.977
2	true	-99.313	-74.957	-92.669	-73.161	-39.814	27.946
3	false	-98.580	93.880	-49.428	-51.465	-76.858	-47.212
4	true	-98.442	-54.029	-68.628	45.448	50.617	57.867
5	false	-98.056	94.098	-49.416	-49.525	-76.657	-45.548
6	true	-98.052	-23.473	-93.455	-16.981	-40.216	-6.407
7	false	-97.899	-60.934	-19.274	-51.545	58.000	79.517
8	false	-97.830	-85.489	22.799	-35.658	16.101	-20.422
9	false	-97.726	-53.950	-24.886	-86.628	-6.258	-4.014
10	false	-97.684	72.942	-79.883	4.960	35.255	-56.468
11	false	-97.654	-59.216	-17.765	-48.769	59.407	80.619
12	false	-97.490	-75.824	84.879	-22.269	-52.061	2.761
13	false	-97.463	-57.883	-16.554	-46.540	60.557	81.520

KNNClassification

```
Weighted 10-Nearest Neighbour model for classification.
The model contains 1600 examples with 2400 dimensions of the following classes:
  false
  true
```

accuracy: 66.18%

	true false	true true	class precisi
pred. false	80	49	62.02%
pred. true	90	192	68.09%
class recall	47.06%	79.67%	

5)gradiant boosted tree

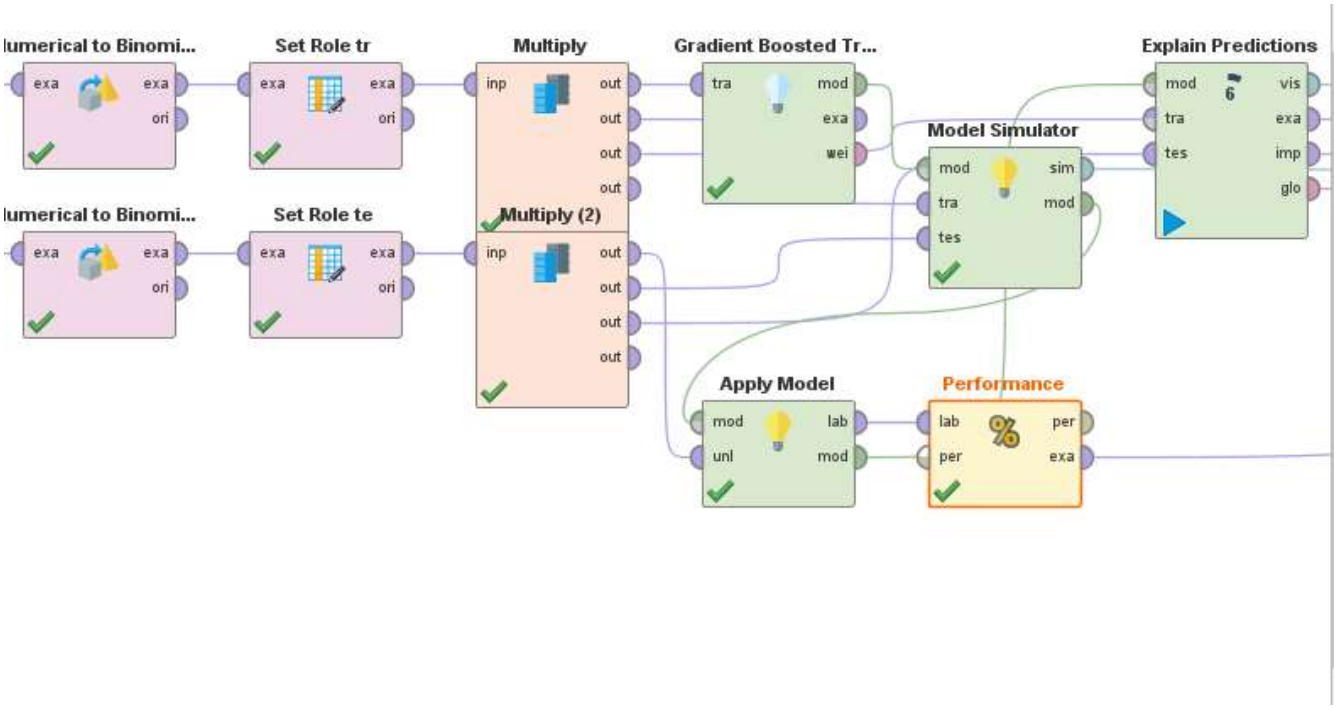
در مورد الگوریتم درختان تصمیم تقویت شده با گرادیان، هر درخت تلاش می کند تا خطاهای درخت قبلی را به حداقل برساند. از آنجایی که درخت ها به صورت متوالی اضافه می شوند، الگوریتم های تقویت به آرامی یاد می گیرند و مدل گام به گام خود را بهبود می بخشد

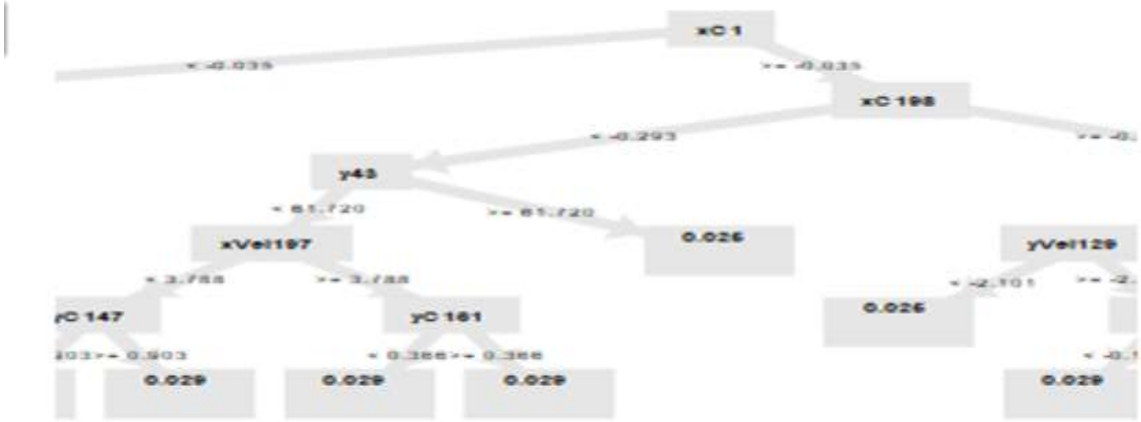
دقت پیش بینی داده ها در این مدل ۹۰٪است نشان دهنده این است که این الگوریتم دقت مناسبی برای داده های ما دارد

از عملگر perfomance classificationبرای ارزیابی دقت مدل استفاده کردیم

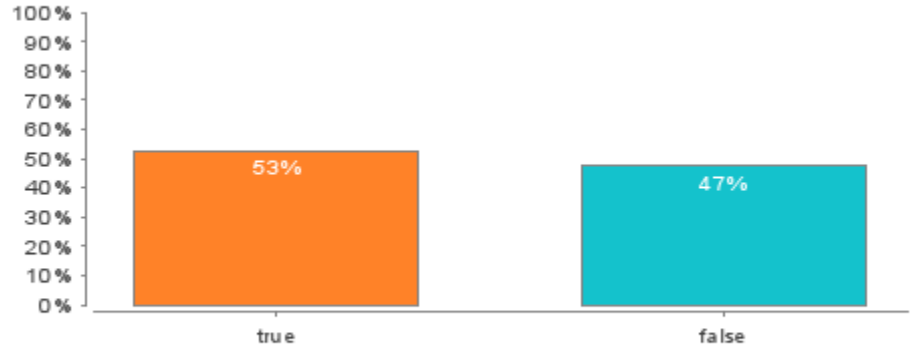
اطمینان برای این تصمیم تنها ۵۲,۵۳ درصد است. مقدار xC1 این تصمیم را پشتیبانی نمی کند. لطفاً به خاطر داشته باشید که ۹۰,۷۵٪ از تمام پیش بینی های انجام شده توسط این مدل درست است. وقتی مدل trueاست ، ۹۶,۲۷ درصد از این موارد را ساپورت می کند . و با ۸۸,۸۹٪ از تمام پیش بینی ها برای کلاس true است.

Row No.	Class	prediction(C...	confidence(f...	confidence(t...	x1	y1	x2
1	false	false	0.803	0.197	9.720	45.385	-16.401
2	false	false	0.803	0.197	9.880	94.774	-83.558
3	false	false	0.803	0.197	10.121	54.628	-82.249
4	true	true	0.458	0.542	10.197	-86.601	74.284
5	true	false	0.577	0.423	10.406	73.393	17.985
6	true	true	0.425	0.575	10.631	34.063	72.530
7	false	false	0.803	0.197	10.661	70.537	66.990
8	false	false	0.803	0.197	10.674	-62.954	-90.367
9	true	true	0.416	0.584	11.040	-58.559	63.197
10	true	true	0.422	0.578	11.105	-98.444	94.404
11	false	false	0.803	0.197	11.132	14.311	46.845
12	false	false	0.708	0.292	11.205	-81.034	77.813
13	false	true	0.509	0.491	11.292	50.086	-3.973
14	false	false	0.803	0.197	11.330	71.264	67.520
15	true	true	0.417	0.583	11.452	98.385	94.545

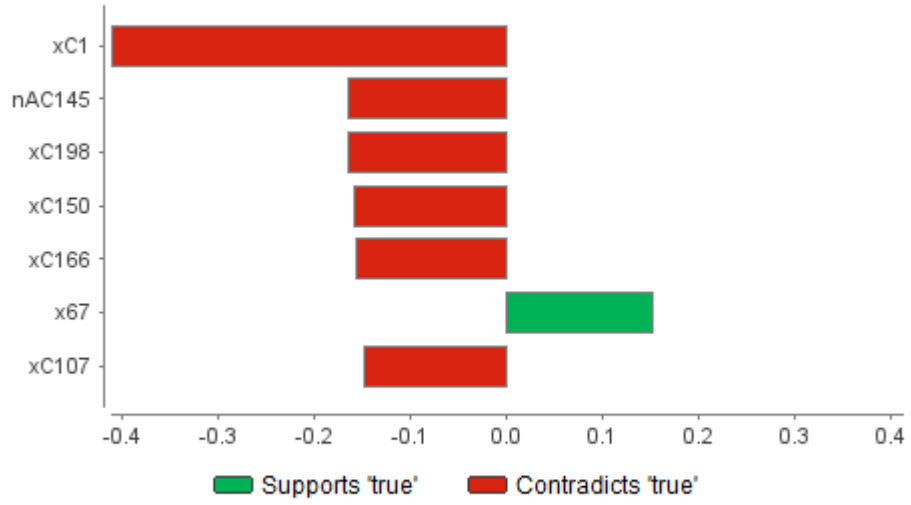




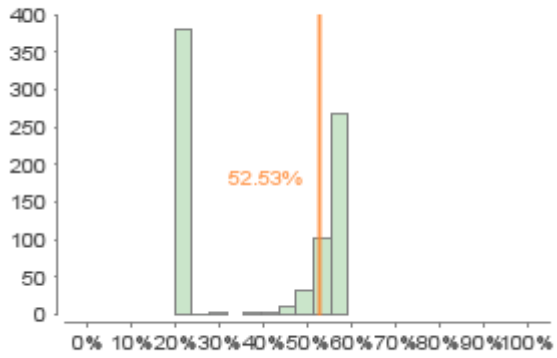
Most Likely: true



Prediction: true



Confidence Distribution for true



Tree

```
xC1 < -0.035
|   yVel154 < -8.557: 0.012 ()
|   yVel154 >= -8.557: 0.031 ()
xC1 >= -0.035
|   xC5 < -0.262
|   |   yA88 < -0.564: 0.026 ()
|   |   yA88 >= -0.564: 0.031 ()
|   xC5 >= -0.262
|   |   yC141 < 0.099
|   |   |   xC161 < -0.288: 0.031 ()
|   |   |   xC161 >= -0.288
|   |   |   |   nS66 < 2.500: -0.015 ()
|   |   |   |   nS66 >= 2.500: 0.013 ()
|   |   yC141 >= 0.099
|   |   |   yVe16 < -2.180: 0.022 ()
|   |   |   yVe16 >= -2.180: 0.031 ()
```

Open in Turbo Prep Auto Model Filter (802 / 802 examples): all

Row No.	Class	prediction(C...	confidence(false)	confidence(true)	x1	y1
1	false	false	0.803	0.197	9.720	45.385
2	false	false	0.803	0.197	9.880	94.774
3	false	false	0.803	0.197	10.121	54.628
4	true	true	0.458	0.542	10.197	-86.601
5	true	false	0.577	0.423	10.406	73.393
6	true	true	0.425	0.575	10.631	34.063
7	false	false	0.803	0.197	10.661	70.537
8	false	false	0.803	0.197	10.674	-62.954
9	true	true	0.416	0.584	11.040	-58.559
10	true	true	0.422	0.578	11.105	-98.444
11	false	false	0.803	0.197	11.132	14.311
12	false	false	0.708	0.292	11.205	-81.034
13	false	true	0.509	0.491	11.292	50.086

accuracy: 90.75%

	true false	true true	class precis
pred. false	141	9	94.00%
pred. true	29	232	88.89%
class recall	82.94%	96.27%	

6)k-mean clustering

این الگوریتم بدون نظارت بااستفاده از روش خوشه بندی افرازی مجموعه داده را به تعداد خوشه هایی که تعیین کردیم(در این مثال ۱۰) تقسیم میکند

همانطورکه در شکل پیداست بیشترین تعداد رکورد ها در خوشه صفر قرار دارد و بقیه داده ها بین خوشه های دیگر تقسیم شده اند که نشان دهنده این است که رکورد ها بسیار به هم نزدیک اند و دسته پرندگان که در یک خوشه در حال حرکت هستند به خوبی قابل تعیین است

بقیه داده ها را میتوان به عنوان داده پرت در نظر گرفت.نشان دهنده دسته پرندگانی است که از خوشه اصلی جداافتاده اند

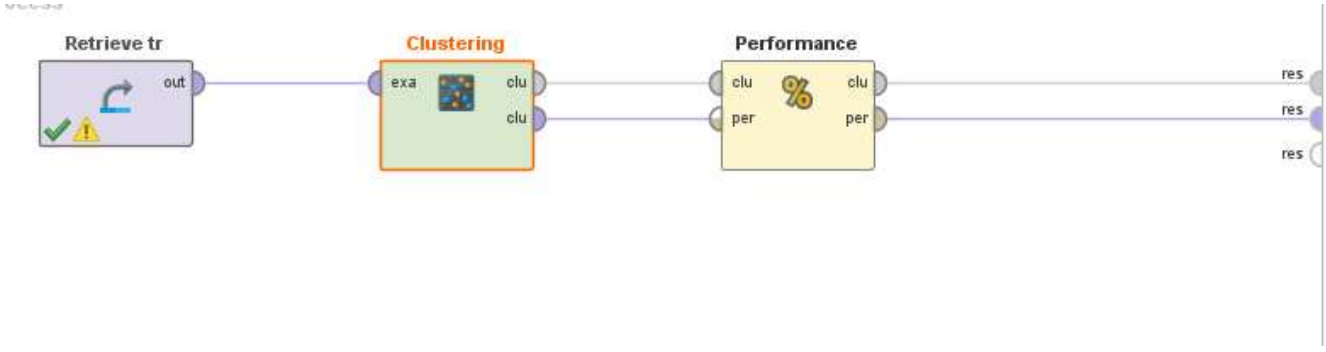
با عملگر cluster count perfomanceکارایی مدل را بررسی کرده

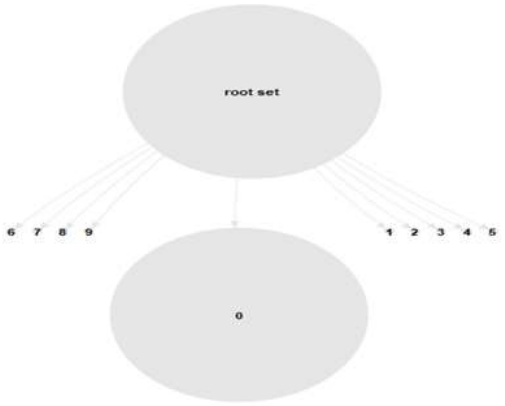
Open in Turbo Prep Auto Model Filter (1,600 / 1,600 examples): all

Row No.	id	cluster	x1	y1	x2	y2	x3
1142	1142	cluster_9	41.683	44.069	10.277	-88.903	-27.177
518	518	cluster_8	-32.188	-10.532	54.302	3.698	-80.396
552	552	cluster_8	-27.296	-7.252	52.123	-0.968	-77.712
792	792	cluster_7	-6.544	-62.210	42.451	8.905	22.598
806	806	cluster_7	-5.083	-61.301	42.602	9.694	21.748
437	437	cluster_6	-46.390	84.975	-24.641	-37.012	73.946
495	495	cluster_6	-35.226	78.050	-15.147	-36.532	67.174
22	22	cluster_5	-96.461	20.542	-30.361	-34.018	-32.824
31	31	cluster_5	-95.863	19.138	-29.573	-33.224	-33.199
291	291	cluster_4	-67.433	37.297	-11.735	-11.065	-66.503
591	591	cluster_3	-19.696	-70.391	41.485	2.279	28.265
257	257	cluster_2	-71.830	31.596	-18.447	-20.662	-52.475
270	270	cluster_2	-70.365	33.496	-16.559	-18.067	-56.506

Open in Turbo Prep Auto Model Filter (1,600 / 1,600 examples): all

Row No.	id	cluster	x1	y1	x2	y2	x3
1	1	cluster_0	-100	-52.672	-41.970	-11.045	59.443
2	2	cluster_0	-99.313	-74.957	-92.669	-73.161	-39.814
3	3	cluster_0	-98.580	93.880	-49.428	-51.465	-76.858
4	4	cluster_0	-98.442	-54.029	-68.628	45.448	50.617
5	5	cluster_0	-98.056	94.098	-49.416	-49.525	-76.657
6	6	cluster_0	-98.052	-23.473	-93.455	-16.981	-40.216
7	7	cluster_0	-97.899	-60.934	-19.274	-51.545	58.000
8	8	cluster_0	-97.830	-85.489	22.799	-35.658	16.101
9	9	cluster_0	-97.726	-53.950	-24.886	-86.628	-6.258
10	10	cluster_0	-97.684	72.942	-79.883	4.960	35.255
11	11	cluster_0	-97.654	-59.216	-17.765	-48.769	59.407
12	12	cluster_0	-97.490	-75.824	84.879	-22.269	-52.061
13	13	cluster_0	-97.463	-57.883	-16.554	-46.540	60.557
14	14	cluster_0	-97.235	-68.213	-68.310	31.615	51.765





Cluster Model

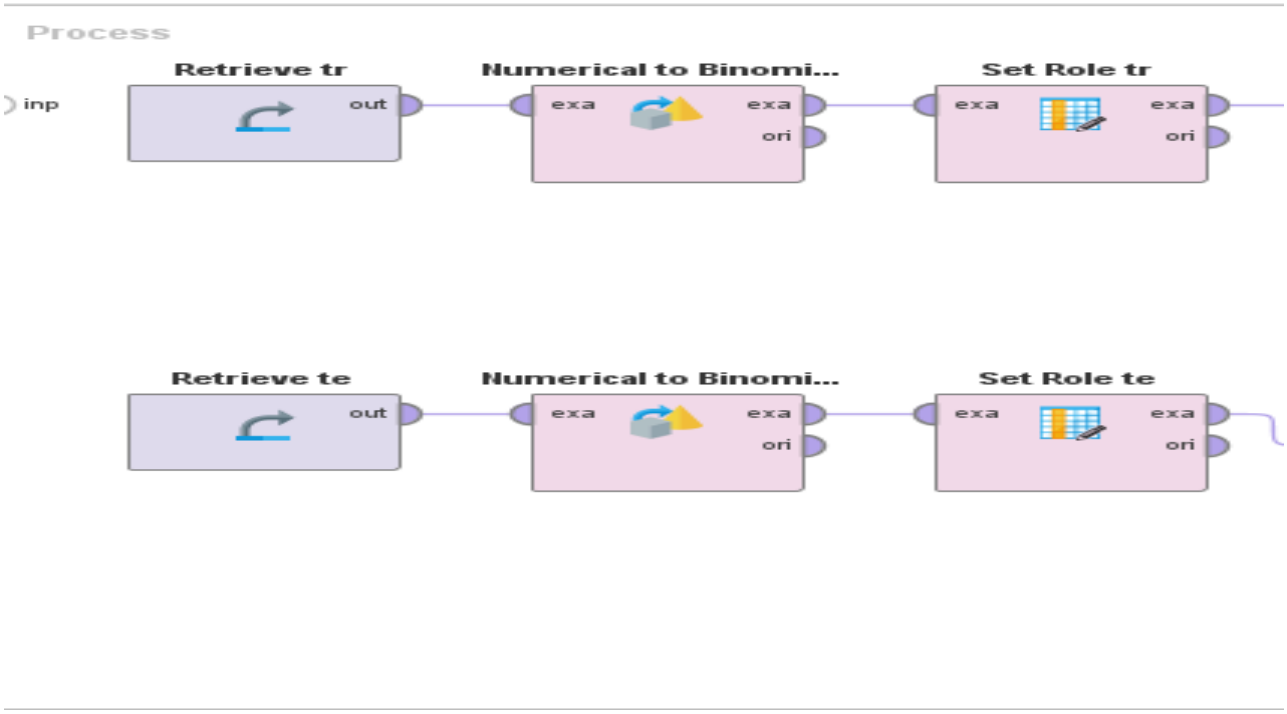
```
Cluster 0: 1585 items
Cluster 1: 2 items
Cluster 2: 2 items
Cluster 3: 1 items
Cluster 4: 1 items
Cluster 5: 2 items
Cluster 6: 2 items
Cluster 7: 2 items
Cluster 8: 2 items
Cluster 9: 1 items
Total number of items: 1600
```

Attribute	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4	cluster_5	cluster_6	cluster_7	cluster_8	cluste
x1	-0.712	-49.547	-71.097	-19.696	-67.433	-96.162	-40.808	-5.814	-29.742	41.683
y1	-8.374	85.258	32.546	-70.391	37.297	19.840	81.512	-61.755	-8.892	44.069
x2	-16.481	-27.351	-17.503	41.485	-11.735	-29.967	-19.894	42.527	53.213	10.277
y2	2.322	-37.289	-19.365	2.279	-11.065	-33.621	-36.772	9.300	1.365	-88.90
x3	-6.322	76.145	-54.490	28.265	-66.503	-33.012	70.560	22.173	-79.054	-27.17
y3	1.813	-89.187	39.269	12.133	38.355	40.970	-87.339	7.828	83.465	-17.85
x4	-12.542	-62.688	-1.520	43.530	11.730	-24.973	-51.881	49.290	93.546	23.466
y4	-5.713	28.509	14.116	-90.316	4.152	30.029	29.932	-76.385	98.637	82.825
x5	3.152	39.893	21.105	-15.134	27.520	0.093	35.837	-5.031	-95.033	-33.83
y5	-1.109	74.088	93.312	77.864	-92.220	51.522	86.204	68.322	-10.414	-62.99
x6	-1.797	94.235	-41.140	47.089	-31.690	-65.231	89.873	51.584	61.763	75.818
y6	-4.542	70.853	-96.932	14.571	-82.995	70.179	73.062	12.868	-22.307	89.117
x7	-0.856	-38.050	14.366	49.314	5.682	32.340	-35.185	58.801	28.270	14.366
y7	0.294	-59.524	-9.983	54.896	-4.680	-31.911	-70.764	50.940	84.902	-16.19
x8	-10.151	-13.168	-25.549	72.956	-38.380	-8.770	-13.193	75.187	76.194	43.338

ارزیابی و تفسیر مدل :

در این مرحله از فرآیند داده کاوی یک عملگر به نام perfomance تعریف می شود که انواع مختلف این عملگر یک لیست از مقادیر کارایی را به صورت خودکار ایجاد و بسته به میزان تناسب با فرآیند یادگیری آنها را به خرجی انتقال می دهد این عملگر میتواند برای تمامی روش های مدل سازی به کار رود

اما برای بالا بردن دقت ارزیابی مدل از عملگرهای گفته شده در بالا (در توضیحات مربوط به هر مدل)استفاده می شود .



نتیجه گیری:با توجه به الگوریتم های بررسی شده در این پروژه بهترین کارایی و دقت برای این مجموعه داده را میتوان باالگوریتم random forestبدست آورد این الگوریتم بهترین مدل را برای این مجموعه داده ساخته و برای داده های جدیدی که وارد مدل می شوند پیش بینی بسیار خوبی با دقت بالا ارائه می دهد