

زهر احیدری 986203146

در این پروژه یک فایل از لیست بیماری های قلبی است که مواردی در آن مشخص شده

سن فرد

2- جنسیت (1 male و 2 female)

3- نوع درد در قفسه سینه

4- فشار خون

5- کلسترول

6- بیماری قند دارد یا نه

7- نتیجه دستگاه ضربه نگار قلب

8- حداکثر ضربان قلب

9- نتیجه تست ورزش

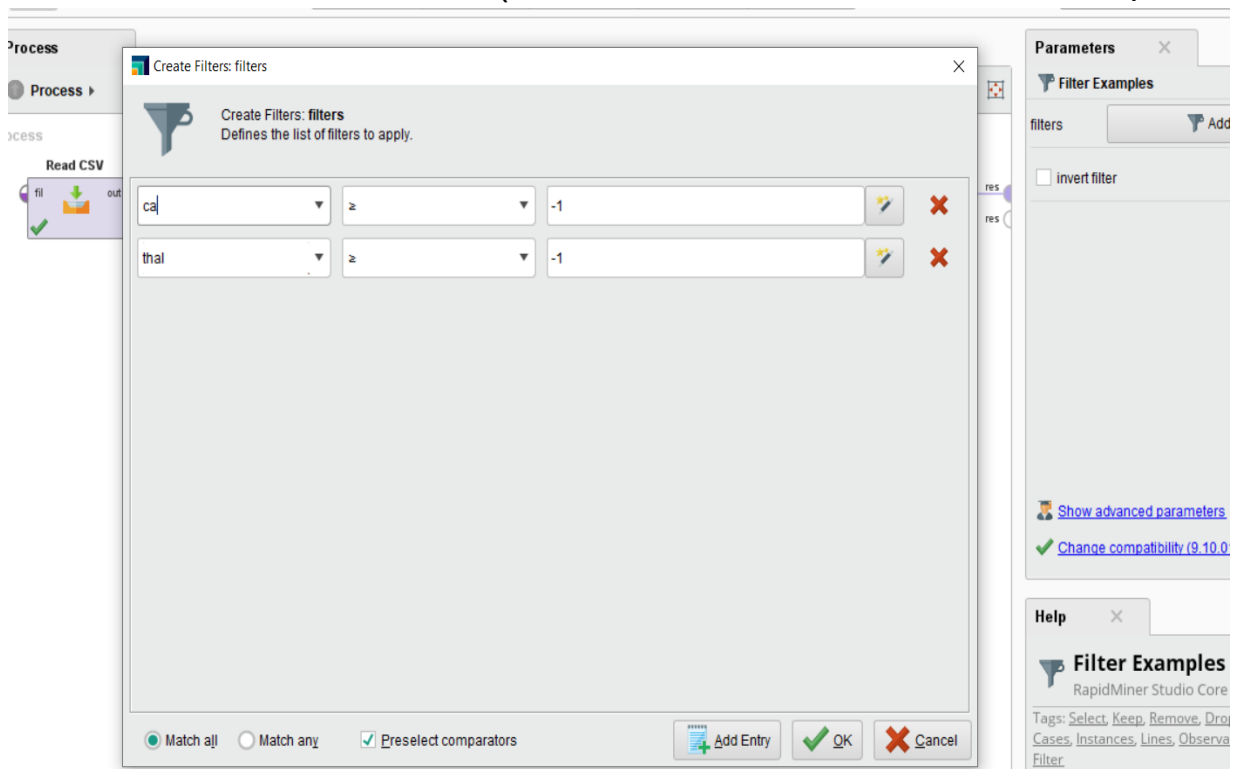
10 تا 13 سایر آزمایش ها

14- ستون برچسب کلاس که می تواند صفر به معنی بیماری قلبی و یک بیماری قلبی ندارد.

که شامل 303 سطر میباشد که با استفاده از نرم افزار rapidminer ما فایل اکسل آن را میخوانیم

Row No.	num	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exa
286	1	58	1	4	114	318	0	1	140	0
287	1	58	0	4	170	225	1	2	146	1
288	0	58	1	2	125	220	0	0	144	0
289	0	56	1	2	130	221	0	2	163	0
290	0	56	1	2	120	240	0	0	169	0
291	1	67	1	3	152	212	0	2	150	0
292	0	55	0	2	132	342	0	0	166	0
293	1	44	1	4	120	169	0	0	144	1
294	1	63	1	4	140	187	0	2	144	1
295	1	63	0	4	124	197	0	0	136	1
296	0	41	1	2	120	157	0	0	182	0
297	1	59	1	4	164	176	1	2	90	0
298	1	57	0	4	140	241	0	0	123	1
299	1	45	1	1	110	264	0	0	132	0
300	1	68	1	4	144	193	1	0	141	0
301	1	57	1	4	130	131	0	0	115	1
302	1	57	0	2	130	236	0	2	174	0
303	0	38	1	3	138	175	0	0	173	0

در ادامه با مشاهده کلی داده ها متوجه یه سری داده های اشتباه میشویم. که در ستون **ca**, **thal** دیده میشود که عدد منفی 1000 را نمایش میدهد برای همین ما کلا با ابزار فیلتر می آیم ان هایی که منفی هستن از این دو سطر به طور کلی حذف میکنیم (یا حتی میشود با **replace** مقدار میانگین را برابر این مقدار قرار داد)



که داده ها به 297 تا تغییر میکند یعنی 6 تا از داده ها فیلتر میشود.

Row No.	num	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exa
280	0	35	1	2	122	192	0	0	174	0
281	1	61	1	4	148	203	0	0	161	0
282	1	58	1	4	114	318	0	1	140	0
283	1	58	0	4	170	225	1	2	146	1
284	0	56	1	2	130	221	0	2	163	0
285	0	56	1	2	120	240	0	0	169	0
286	1	67	1	3	152	212	0	2	150	0
287	0	55	0	2	132	342	0	0	166	0
288	1	44	1	4	120	169	0	0	144	1
289	1	63	1	4	140	187	0	2	144	1
290	1	63	0	4	124	197	0	0	136	1
291	0	41	1	2	120	157	0	0	182	0
292	1	59	1	4	164	176	1	2	90	0
293	1	57	0	4	140	241	0	0	123	1
294	1	45	1	1	110	264	0	0	132	0
295	1	68	1	4	144	193	1	0	141	0
296	1	57	1	4	130	131	0	0	115	1
297	1	57	0	2	130	236	0	2	174	0

و سپس من یه تغییری در داده ها دادم اینک اون داده 14 که لیب میشود به اضافه 3 تا ستون دیگر را بجای 0 و 1 به ترتیب مقدار yes و no قرار دادم که برای آسانی خودم برای داده هاست.

The image displays a workflow in RapidMiner Studio. The workflow consists of the following steps:

- Read CSV**: The starting point of the process.
- Filter Examples**: A filter operator that filters the data based on specific criteria.
- Replace**: An operator that replaces values in the data.
- Replace (2)**: A second replacement operator.

The right-hand side of the image shows the configuration for the **Replace (2)** operator:

- value type**: polynomial
- replace what**: 0
- replace by**: yes

Below this, there are links for [Hide advanced parameters](#) and [Change compatibility \(9.10.011\)](#).

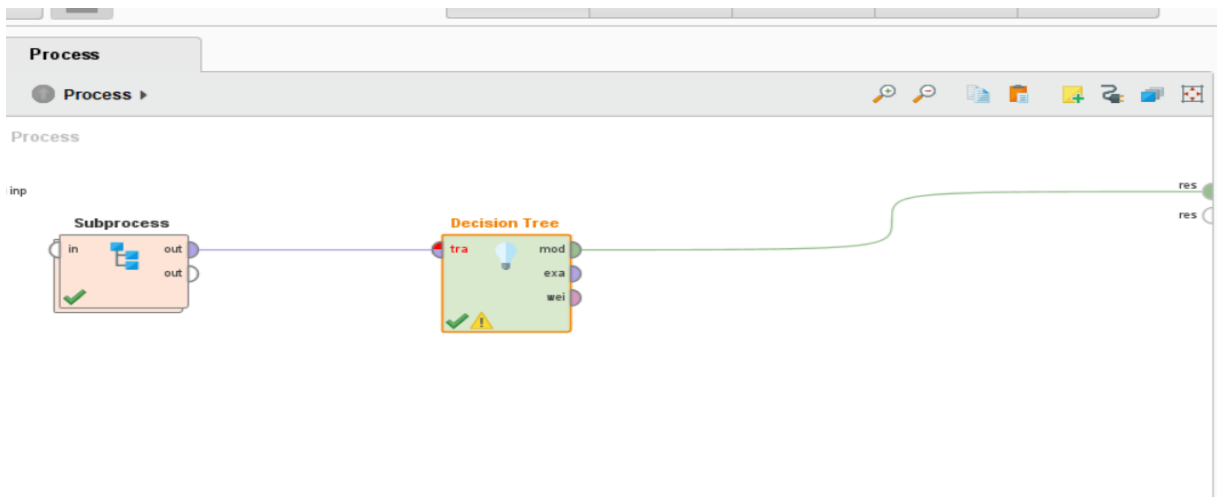
The bottom part of the image shows the **Process** view of the workflow, which is a subprocess. The workflow is shown in a design view, and the right-hand side shows the configuration for the **Replace (2)** operator, which is now set to:

- attribute filter type**: value_type
- value type**: polynomial
- replace what**: 1
- replace by**: no

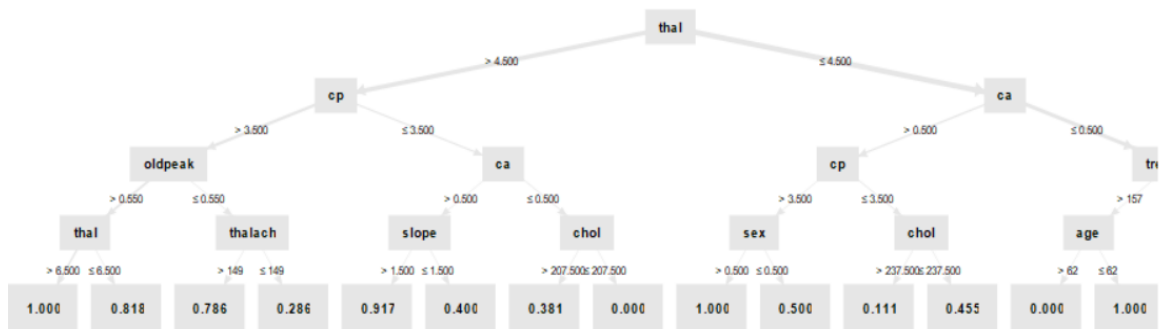
Below this, there are links for [Hide advanced parameters](#) and [Change compatibility \(9.10.011\)](#).

The bottom right corner shows the **Help** view for the **Replace** operator, which includes a synopsis and tags: [Map](#), [Change](#), [Regex](#), [Regular expressions](#), and [Values](#).

و همه این موارد را در subprocess قرار میدهم. که یکم دیدن شمای برنامه راحت تر شود.



ابزار درخت تصمیم را به آن اضافه میکنیم . و خروجی آن به صورت زیر است.



actions/DataMining* - RapidMiner Studio Educational 9.10.011 @ DESKTOP-JC7P0U8

w Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

ExampleSet (Apply Model)

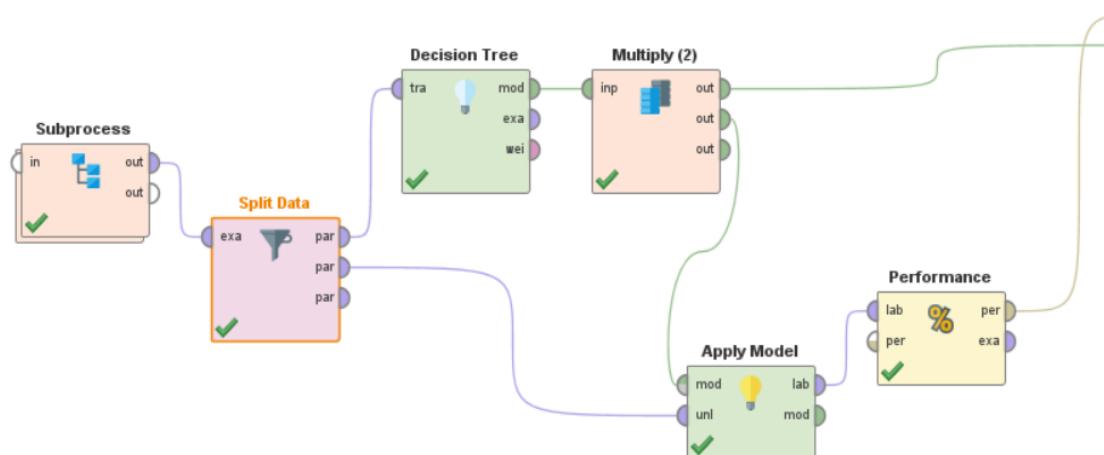
in Turbo Prep Auto Model Filter (297 / 297 examples): all

No.	num	prediction(num)	age	sex	cp	trestbps	chol	fb	restecg
	0	0.296	63	1	1	145	233	1	2
	1	0.850	67	1	4	160	286	0	2
	1	0.971	67	1	4	120	229	0	2
	0	0.083	37	1	3	130	250	0	0
	0	0.083	41	0	2	130	204	0	2
	0	0.083	56	1	2	120	236	0	0
	1	0.850	62	0	4	140	268	0	2
	0	0.083	57	0	4	120	354	0	0
	1	0.971	63	1	4	130	254	0	2
	1	0.971	53	1	4	140	203	1	2
	0	0.619	57	1	4	140	192	0	0
	0	0.083	56	0	2	140	294	0	2
	1	0.765	56	1	3	130	256	1	2
	0	0.296	44	1	2	120	263	0	0
	0	0.296	52	1	3	172	199	1	0
	0	0.083	57	1	3	150	168	0	0
	1	0.296	48	1	2	110	229	0	0
	0	0.083	54	1	4	140	239	0	0

حال می‌ایم با استفاده از **apply model** درخت تصمیم را برحسب مدلی که بهش می‌دهیم که اینجا برحسب ابزار **split data** داده‌ها رو به دو بخش تقسیم می‌کنم که یک بخش رو به مدل می‌دهیم که بر اساس آن تصمیم بگیرد که داده‌های اصلی رو چگونه تقسیم کند و به آن درصد می‌دهیم در ابزار **split data** داده‌ها رو **split** می‌کنیم به دو بخش زیر:

ratio
0.8
0.2

و به صورت زیر بهم وصلشان می‌کنیم.

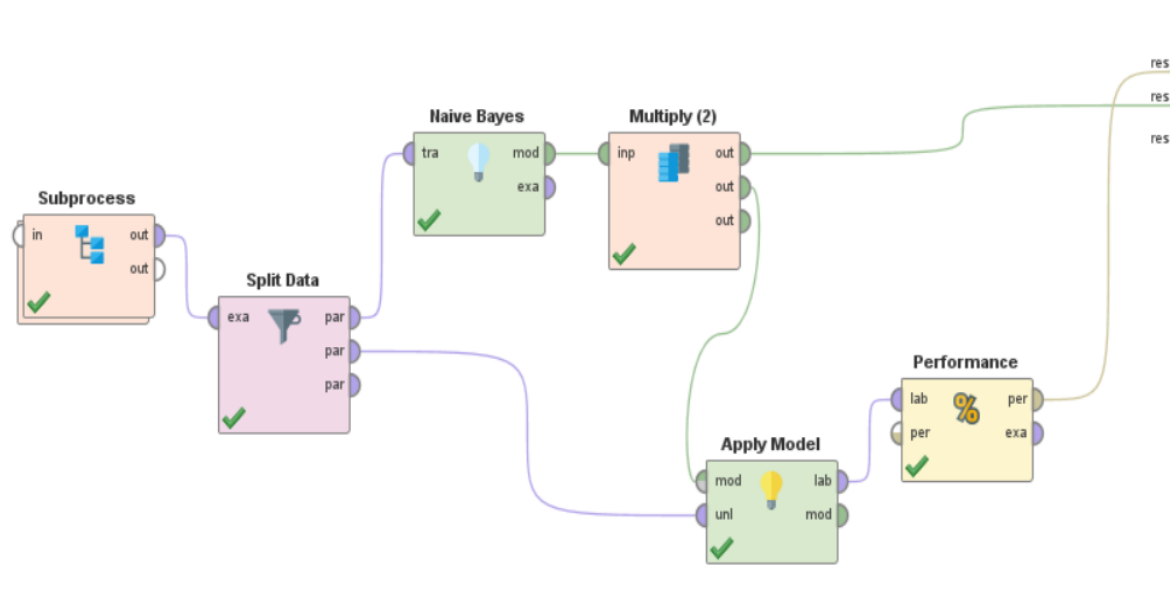


ابزار performance در آن ما می توانیم accuracy و recall مثلا این روش درخت تصمیم و روش بیزین رو ببینیم و باهم مقایسه کنیم که در درخت تصمیم این داده ها به صورت زیر است.

accuracy: 77.97%

	true 1	true 0	class precision
pred. 1	20	6	76.92%
pred. 0	7	26	78.79%
class recall	74.07%	81.25%	

و حال بجای ابزار درخت تصمیم ما از نوی بیزین استفاده میکنیم



و خروجی آن به صورت زیر است.

accuracy: 84.75%

	true 1	true 0	class precision
pred. 1	21	3	87.50%
pred. 0	6	29	82.86%
class recall	77.78%	90.62%	

که در اینجا میبینم در روش بیزین نسبت به روش درخت تصمیم accuracy و recall آن افزایش پیدا کرده است.