



دانشگاه کاشان
دانشکده برق و کامپیوتر

گزارش پروژه درس «مبانی داده کاوی»
در رشته کامپیوتر گرایش نرم افزار

توسط:
سبا پورعجم
۹۳۲۱۱۷۰۰۴

استاد درس:
دکتر سید مهدی وحیدی پور

۱۳۹۶/۱۱/۱۱

فهرست مطالب

بخش ۱ - ساخت پروسه و وارد کردن دیتاست به رپیدماینر

بخش ۲ - سوال ۱ :

- حالت a
- حالت b
- حالت c
- تعیین label برای دیتاست
- حالت d
- حالت
- حالت e
- حالت f
- حالت g
- حالت h

بخش ۳ - سوال ۳

- حالت e
- حالت f

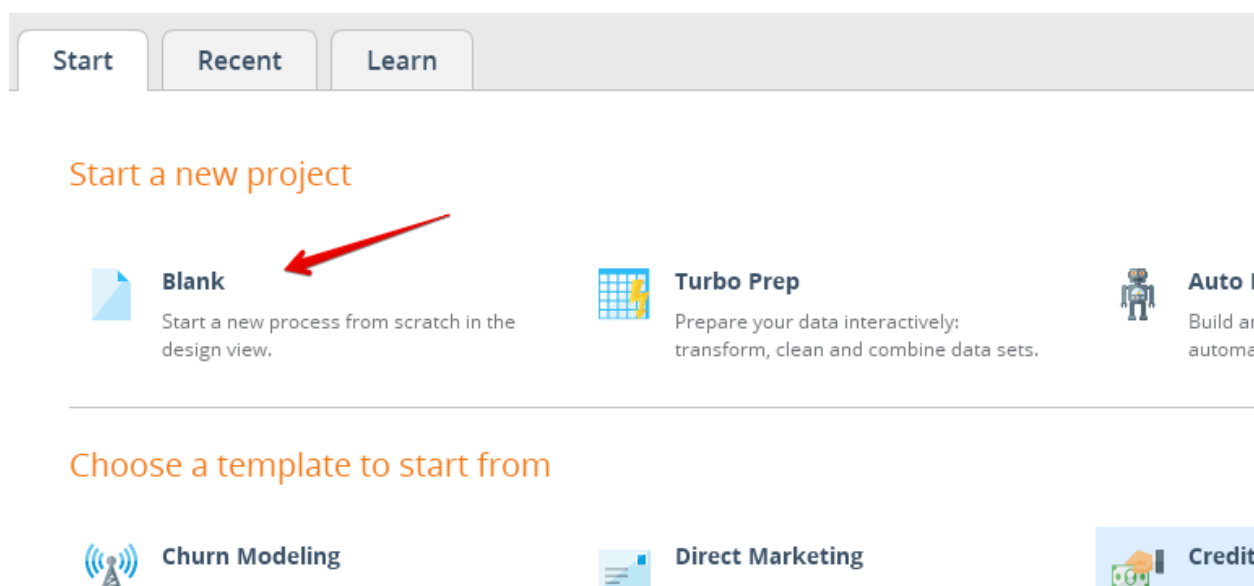
بخش ۴ - سوال ۴

- حالت g
- حالت h

بخش ۱

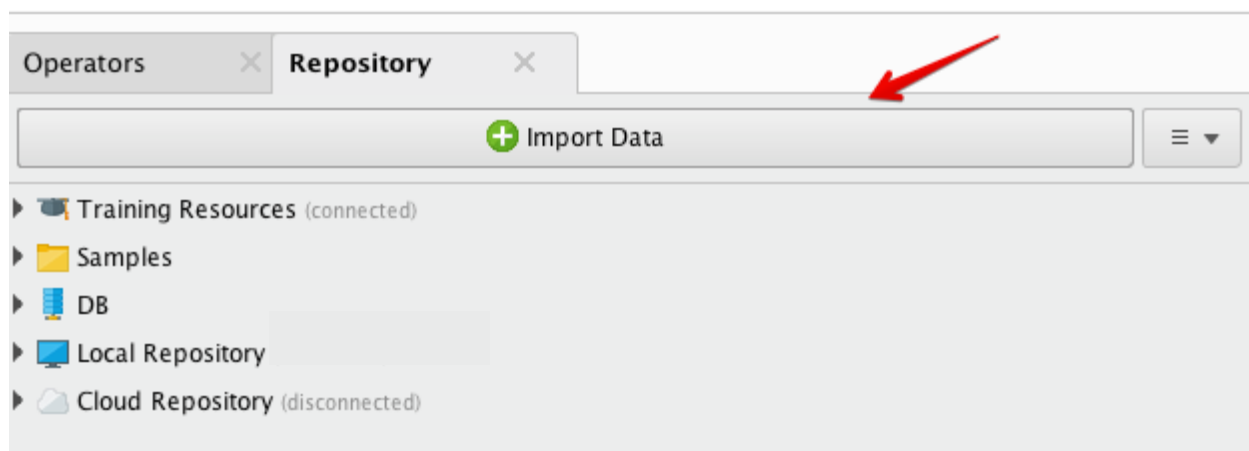
ساخت پروسه و وارد کردن دیتاست به ریپیدماینر

پس از ورود به محیط ریپیدماینر ۹.۱، در منوی اصلی، در قسمت **File** گزینه **New Process** را انتخاب کرده و سپس در صفحه‌ی باز شده گزینه **Blank** را انتخاب می‌کنیم (شکل ۱-۱).

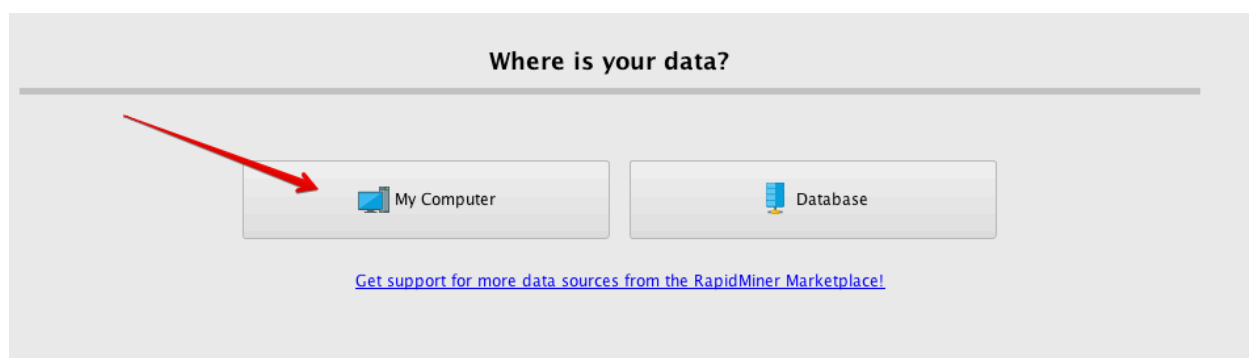


شکل ۱-۱

در پنجره‌ی **Repository** روی **Import Data** کلیک کرده (شکل ۲-۱) و سپس **My Computer** را می‌زنیم (شکل ۳-۱) و سپس به محل فایل اکسل دیتاست رفته و آن را انتخاب می‌کنیم و **Next** را می‌زنیم. در صفحه‌ی بعد اطلاعات فایل اکسل نمایش داده می‌شود (شکل ۴-۱). روی **Next** کلیک کرده و در صفحه‌ی بعد به آخرین ستون (ستون **c**) رفته و آن روی آن راست کلیک کرده و در قسمت **Change Type** **binominal** را انتخاب می‌کنیم. چون ستون **c** ستون **label** است و نوع آن باید **nominal** و یا **binominal** باشد و چون در اینجا دو کلاس بیشتر نداریم **binominal** را انتخاب می‌کنیم (شکل ۵-۱). سپس روی **Next** کلیک می‌کنیم و سپس **Finish** را می‌زنیم تا دیتاست وارد ریپیدماینر شود.



شکل ۲-۱



شکل ۳-۱

Select the cells to import.

Sheet: Sheet1 Cell range: A1:N598 Select All Define header row: 1

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Age (ag...	sex	chest p...	blood p...	cholest...	blood s...	electroc...	heart ra...	exercise...	depress...	slope	ca	thal
2	63.000	1.000	1.000	145.000	233.000	1.000	2.000	150.000	0.000	2.300	3.000	0.000	6.00
3	37.000	1.000	3.000	130.000	250.000	0.000	0.000	187.000	0.000	3.500	3.000	0.000	3.00
4	41.000	0.000	2.000	130.000	204.000	0.000	2.000	172.000	0.000	1.400	1.000	0.000	3.00
5	56.000	1.000	2.000	120.000	236.000	0.000	0.000	178.000	0.000	0.800	1.000	0.000	3.00
6	57.000	0.000	4.000	120.000	354.000	0.000	0.000	163.000	1.000	0.600	1.000	0.000	3.00
7	57.000	1.000	4.000	140.000	192.000	0.000	0.000	148.000	0.000	0.400	2.000	0.000	6.00
8	56.000	0.000	2.000	140.000	294.000	0.000	2.000	153.000	0.000	1.300	2.000	0.000	3.00
9	44.000	1.000	2.000	120.000	263.000	0.000	0.000	173.000	0.000	0.000	1.000	0.000	7.00
10	52.000	1.000	3.000	172.000	199.000	1.000	0.000	162.000	0.000	0.500	1.000	0.000	7.00
11	57.000	1.000	3.000	150.000	168.000	0.000	0.000	174.000	0.000	1.600	1.000	0.000	3.00
12	54.000	1.000	4.000	140.000	239.000	0.000	0.000	160.000	0.000	1.200	1.000	0.000	3.00
13	48.000	0.000	3.000	130.000	275.000	0.000	0.000	139.000	0.000	0.200	1.000	0.000	3.00
14	49.000	1.000	2.000	130.000	266.000	0.000	0.000	171.000	0.000	0.600	1.000	0.000	3.00
15	64.000	1.000	1.000	110.000	211.000	0.000	2.000	144.000	1.000	1.800	2.000	0.000	3.00
16	58.000	0.000	1.000	150.000	283.000	1.000	2.000	162.000	0.000	1.000	1.000	0.000	3.00
17	50.000	0.000	3.000	120.000	219.000	0.000	0.000	158.000	0.000	1.600	2.000	0.000	3.00
18	58.000	0.000	3.000	120.000	340.000	0.000	0.000	172.000	0.000	0.000	1.000	0.000	3.00
19	66.000	0.000	1.000	150.000	226.000	0.000	0.000	114.000	0.000	2.600	3.000	0.000	3.00
20	43.000	1.000	4.000	150.000	247.000	0.000	0.000	171.000	0.000	1.500	1.000	0.000	3.00
21	60.000	0.000	1.000	140.000	238.000	0.000	0.000	151.000	0.000	1.800	1.000	0.000	3.00

Previous Next Cancel

شکل ۴-۱

slope	ca	thal	c
integer	polynomial	polynomial	binomial
3	0	6	0
3	0	3	0
1	0	3	0
1	0	3	0
1	0	3	0
2	0	6	0
2	0	3	0
1	0	7	0
1	0	7	0
1	0	3	0
1	0	3	0
1	0	3	0
1	0	3	0
2	0	3	0
1	0	3	0
2	0	3	0
1	0	3	0
3	0	3	0
1	0	3	0

Repository (disconnected)

- polynomial
- binomial**
- real
- integer
- date_time
- date
- time

no problems.

Previous Next Cancel

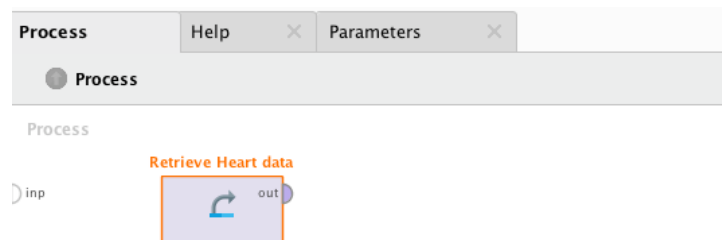
شکل ۵-۱

بخش ۲

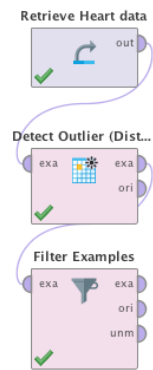
سوال ۱

حالت a

ابتدا از قسمت Repository دیتاست را به روی پروسه کشیده و رها می‌کنیم. (شکل ۱-۲). سپس اپراتور Detect Outlier Distances را از پنجره‌ی Operators انتخاب کرده و به داخل پروسه آورده و خروجی out دیتاست را به ورودی exa آن وصل می‌کنیم. سپس اپراتور Filter Examples را آورده و خروجی exa از اپراتور Detect Outlier Distances را به ورودی exa از اپراتور Filter Examples وصل می‌کنیم. (شکل ۲-۲). پارامترهای Detect Outlier Distances را مطابق شکل ۲-۳ تنظیم می‌کنیم. این اپراتور باعث به وجود آمدن ستونی به نام outlier در داده‌ها می‌شود که اگر true باشد معنی آن اینست که داده نویز است و باید حذف شود. برای انجام این کار در قسمت Parameters از اپراتور Filter Examples روی Add Filters... کلیک کرده (شکل ۲-۴) و فیلتری مشابه شکل ۲-۵ تنظیم کرده و OK را می‌زنیم.




شکل ۱-۲




شکل ۲-۲



Operators	Repository	Parameters
Detect Outlier (Distances)		
number of neighbors	10	ⓘ
number of outliers	30	ⓘ
distance function	squared distance	ⓘ

شکل ۳-۲




Operators	Repository	Parameters
Filter Examples		
filters		Add Filters... ⓘ
condition class	custom_filters	ⓘ
<input type="checkbox"/> invert filter		ⓘ

شکل ۴-۲

 Create Filters: **filters**
Defines the list of filters to apply.

outlier equals false  

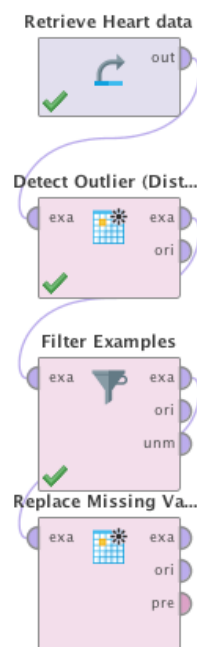
☒ Match all ☐ Match any ☒ Preselect comparators

 Add Entry  OK  Cancel

شکل ۵-۲

حالت b

اپراتور **Replace missing value** را وارد کرده و خروجی **exa** از اپراتور **Filter Examples** را به ورودی **exa** آن وصل می‌کنیم (شکل ۶-۲).



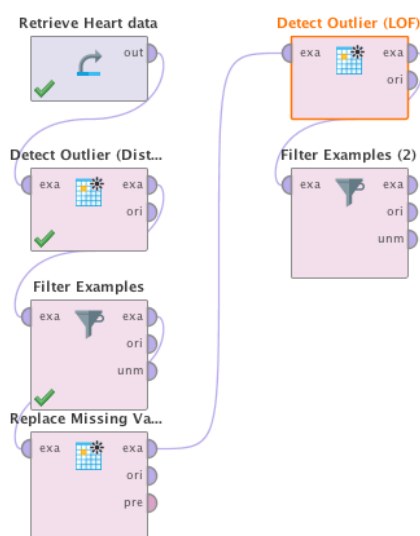
شکل ۶-۲

حالت c

اپراتور Detect Outlier (LOF) را وارد پروسه کرده و خروجی exa از اپراتور قبلی را به ورودی exa آن وصل می‌کنیم. سپس تنظیمات پارامترهای آن را مطابق شکل ۷-۲ انجام می‌دهیم. سپس اپراتور Filter Examples را وارد پروسه کرده و خروجی exa از آن وصل می‌کنیم (شکل ۸-۲).

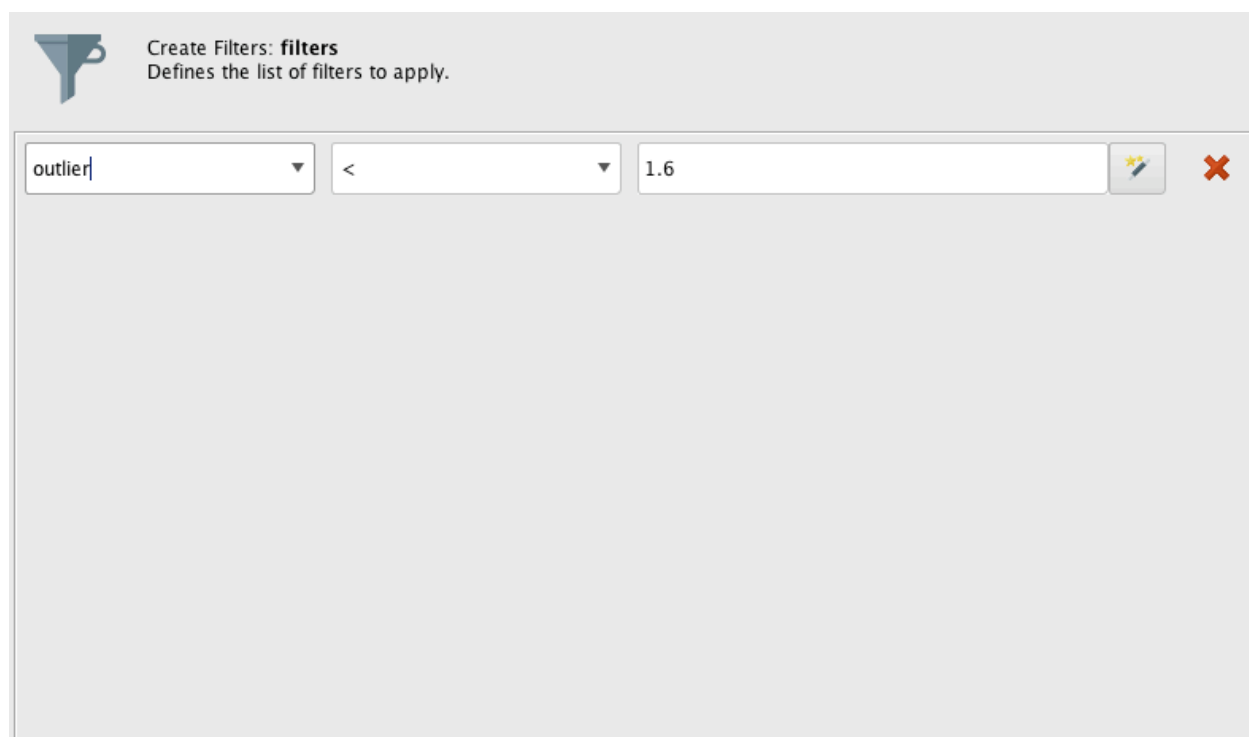
Operators	Repository	Parameters
Detect Outlier (LOF)		
minimal points lower bound	3	
minimal points upper bound	7	
distance function	squared distance	

شکل ۷-۲



شکل ۸-۲

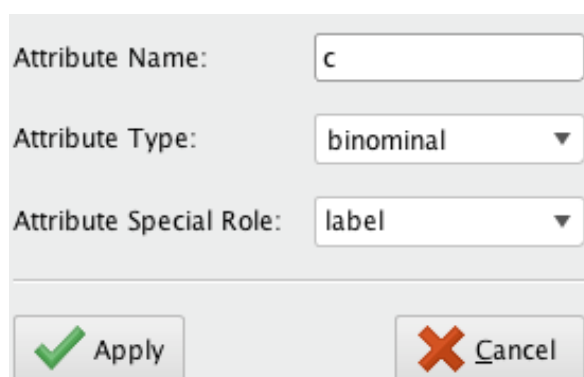
اپراتور LOF یک فیلد outlier به داده‌ها اضافه می‌کند که نشان‌دهنده‌ی میزان دور بودن داده از داده‌های اطراف آن است. با یک مشاهده‌ی کلی روی داده‌ها متوجه می‌شویم که تقریباً نیمی از آن‌ها مقدار outlier صفر را دارند که یعنی کاملاً در نزدیکی بقیه داده‌های اطرافشان قرار دارند. اما نیمی دیگر مقدار outlier غیر صفر را دارند. الان سوال این است که مرز جداسازی داده‌های نویز با داده‌های غیرنویز را چه عددی از outlier مشخص می‌کند. با نگاه گذرا روی داده‌ها درمیابیم عدد ۱.۶ مقدار مناسبی برای تعیین این مرز است. پس در تنظیمات پارامترهای Filter Examples (که پس از LOF قرار دادیم) روی Add Filters... کلیک کرده و فیلتری مطابق شکل ۹-۲ تنظیم می‌کنیم. این فیلتر داده‌هایی که مقادیر outlier آن‌ها بیشتر از ۱.۶ است را از داده‌ها حذف می‌کنند.



شکل ۹-۲

تعیین label برای دیتاست

به منظور وارد کردن دیتاست به مدل‌ها برای انجام classification (در قسمت‌های بعدی) باید یکی از ستون‌های دیتاست را با عنوان label به رپیدماینر بشناسانیم. برای این کار از پنجره‌ی Repository روی اسم دیتاست راست کلیک کرده و Edit را می‌زنیم. در پنجره‌ی باز شده (Data Editor) در روی ستون c راست کلیک کرده و Modify Attribute را می‌زنیم. سپس در پنجره‌ی باز شده در بخش Attribute Special Role گزینه‌ی label را انتخاب می‌کنیم (شکل ۲-۲۸)

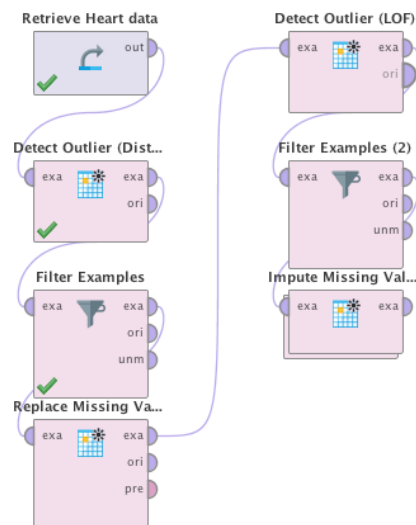


شکل ۲-۲۸

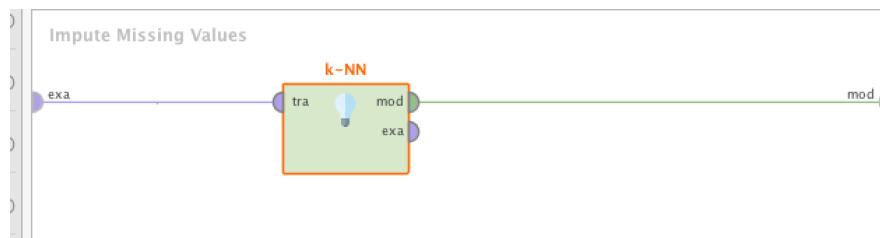
و سپس در منوی پنجره‌ی Data Editor علامت ذخیره را کلیک می‌کنیم تا تغییرات اعمال شوند.

حالت d

اپراتور Impute Missing Values را وارد پروسه کرده و سپس خروجی exa از اپراتور قبلی را به ورودی exa آن وصل می‌کنیم (شکل ۲-۱۰). سپس روی آن دوبار کلیک می‌کنیم تا وارد subprocess آن بشویم. در داخل زیرپروسه، اپراتور K-NN را وارد کرده و ورودی exa زیرپروسه را به ورودی tra آن وصل کرده و خروجی mod از K-NN را به خروجی mod از زیر پروسه وصل می‌کنیم (شکل ۲-۱۱). سپس در قسمت پارامترهای K-NN مقدار k را طبق توضیح پروژه برابر با ۱۰ قرار می‌دهیم.



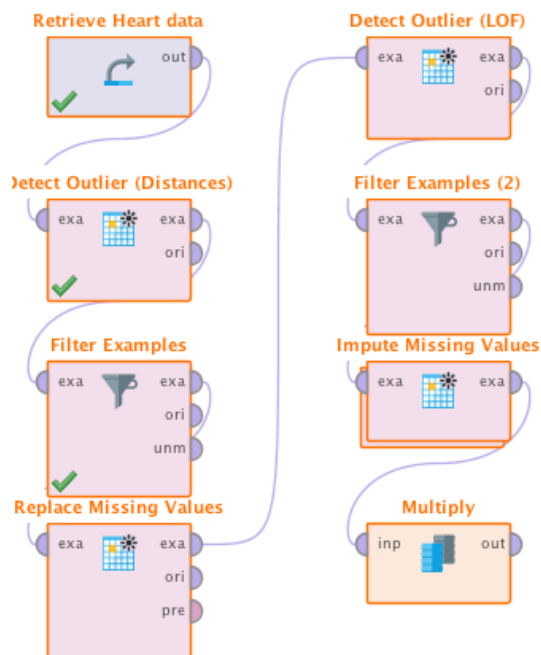
شکل ۲-۱۰



شکل ۲-۱۱

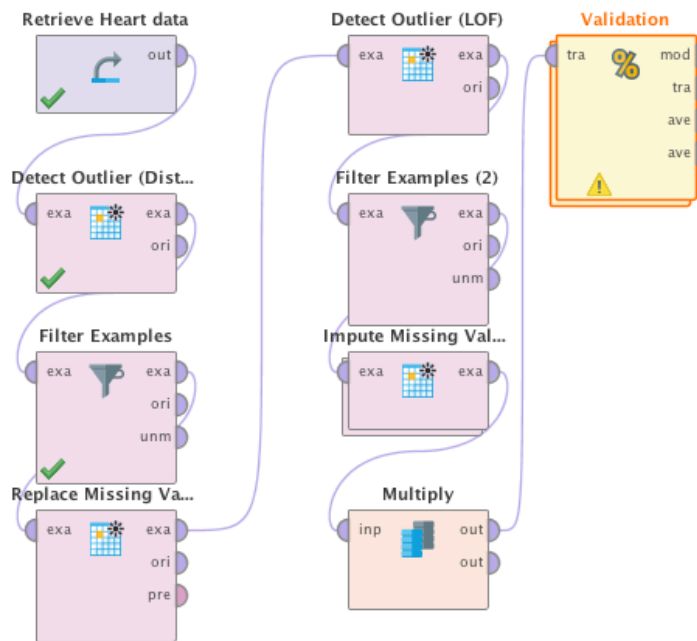
حالت e

ابتدا اپراتور **Multiply** را وارد پروسه می‌کنیم تا در مراحل بعدی از آن برای دیگر اپراتورها استفاده کنیم. کار این اپراتور صرفاً این است که مقدار ورودی را در بی‌شمار خروجی به ما میدهد و بنابراین خروجی قسمت قبل که دیتاست نویزگیری شده بود را توسط این اپراتور می‌توانیم در تعداد بی‌شماری اپراتور دیگر استفاده کنیم (شکل ۲-۱۲).



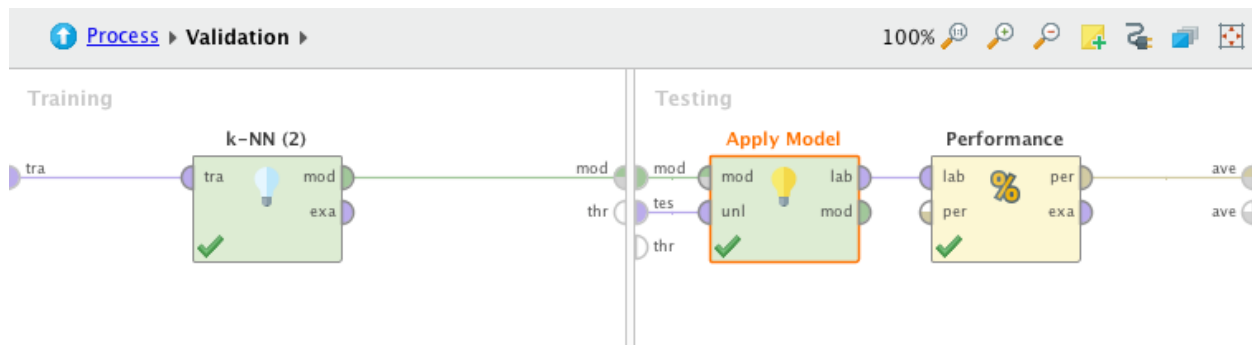
شکل ۱۲-۲

حال اپراتور Split Validation را وارد پروسه می‌کنیم. و اولین خروجی out از اپراتور Multiply را به ورودی tra آن وصل می‌کنیم (شکل ۱۳-۲). سپس در بخش پارامترهای آن مقدار split ratio را برابر ۰.۸ قرار می‌دهیم (طبق توضیح پروژه).



شکل ۲-۱۳

حال روی اپراتور Split Validation دوبار کلیک می‌کنیم و وارد زیرپروسه‌ی آن می‌شویم. سپس در Training آن اپراتور K-nn را وارد می‌کنیم. مقدار k آن را برابر ۸ قرار می‌دهیم و سپس در بخش Testing از زیرپروسه ابتدا اپراتور Apply Model را برای دادن داده‌های تست (۲۰ درصد کل داده‌ها) به مدل، وارد کرده و سپس اپراتور Performance (Classification) را وارد می‌کنیم و مطابق شکل ۲-۱۴ آن‌ها را متصل می‌کنیم.



شکل ۲-۱۴

برای اپراتور (Classification) Performance در قسمت پارامترها تیک‌های Accuracy و weighted mean recall و mean precision را می‌زنیم. پس از اجرا، مقادیر خروجی مشابه شکل ۱۵-۲ هستند.

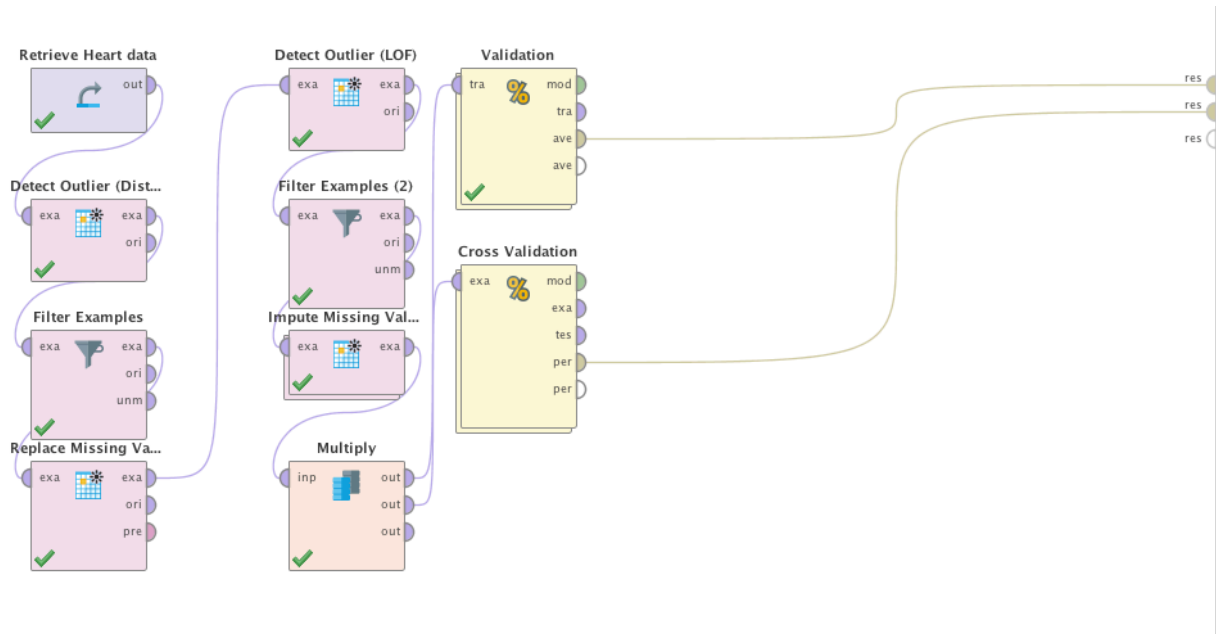
PerformanceVector

```
PerformanceVector:
accuracy: 67.33%
ConfusionMatrix:
True:  0      1
0:     49     21
1:     12     19
weighted_mean_recall: 63.91%, weights: 1, 1
ConfusionMatrix:
True:  0      1
0:     49     21
1:     12     19
weighted_mean_precision: 65.65%, weights: 1, 1
ConfusionMatrix:
True:  0      1
0:     49     21
1:     12     19
```

شکل ۱۵-۲

حالت f

شبیه به حالت e ابتدا اپراتور Cross Validation را وارد پروسه کرده و مطابق شکل ۱۶-۲ متصل می‌کنیم. سپس مقادیر پارامترهای آن را مطابق شکل ۱۷-۲ تنظیم می‌کنیم. با دوبار کلیک روی آن وارد زیرپروسه شده و اپراتورهای K-NN و Apply Model و Performance (Classification) را وارد کرده و مطابق شکل ۱۸-۲ متصل می‌کنیم. برای K-NN مقدار k را در پارامترهای آن برابر ۸ قرار داده و برای Performance (Classification) تیک گزینه‌های Accuracy و weighted mean recall و weighted mean precision را می‌زنیم. پس از اجرای پروسه مقادیر معیارهای کارایی در شکل ۱۹-۲ قابل مشاهده‌اند.



شکل ۱۶-۲

Cross validation

☐ split on batch attribute

☐ leave one out

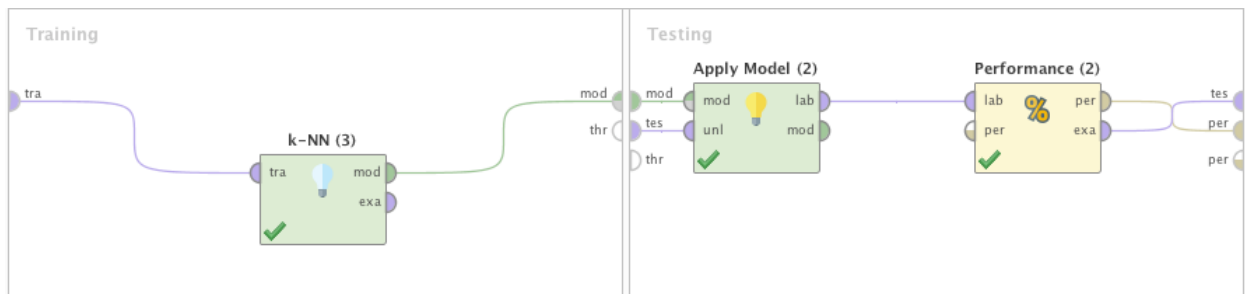
number of folds

sampling type

☐ use local random seed

☒ enable parallel execution

شکل ۱۷-۲



شکل ۱۸-۲

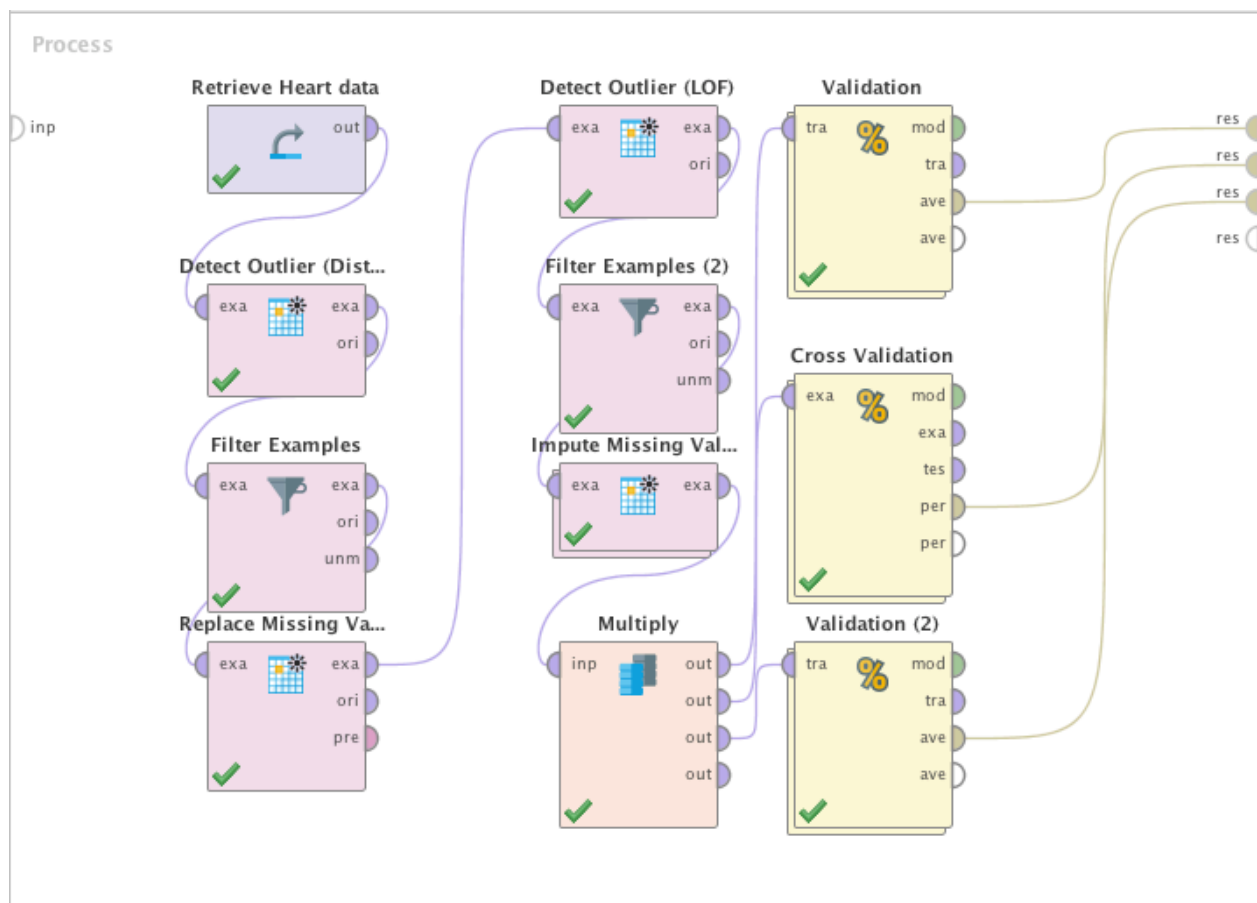
PerformanceVector

```
PerformanceVector:
accuracy: 64.04% +/- 5.77% (micro average: 64.03%)
ConfusionMatrix:
True:  0      1
0:    230    105
1:     77     94
weighted_mean_recall: 61.02% +/- 5.44% (micro average: 61.08%), weights: 1, 1
ConfusionMatrix:
True:  0      1
0:    230    105
1:     77     94
weighted_mean_precision: 62.19% +/- 6.43% (micro average: 61.81%), weights: 1, 1
ConfusionMatrix:
True:  0      1
0:    230    105
1:     77     94
```

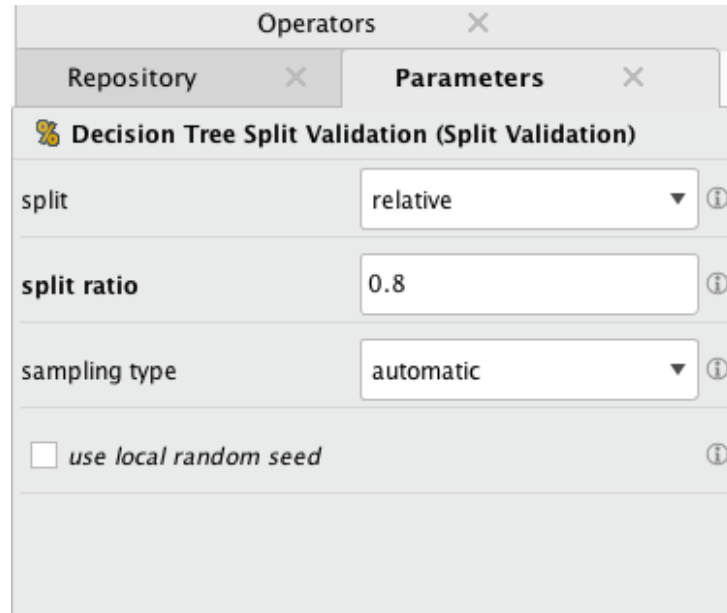
شکل ۱۹-۲

حالت g

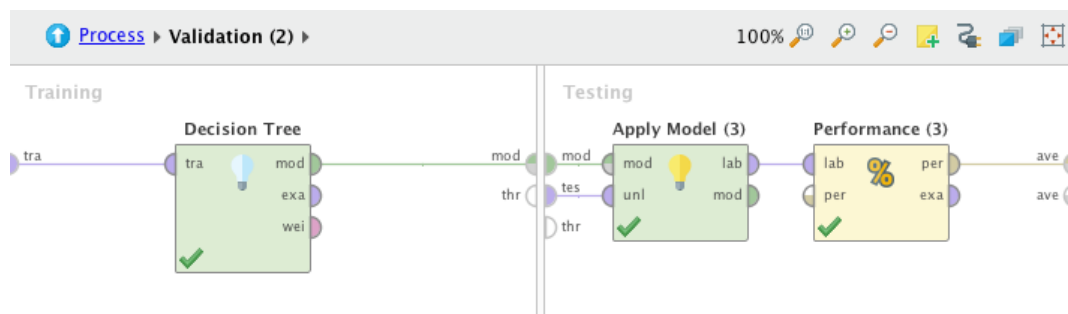
شبیه به حالت e ابتدا اپراتور Split Validation را وارد پروسه کرده و مطابق شکل ۲۰-۲ متصل می‌کنیم. سپس مقادیر پارامترهای آن را مطابق شکل ۲۱-۲ تنظیم می‌کنیم. با دوبار کلیک روی آن وارد زیرپروسه شده و اپراتورهای Decision Tree و Apply Model و Performance (Classification) را وارد کرده و مطابق شکل ۲۲-۲ متصل می‌کنیم. برای Decision Tree مقدار maximal depth را در پارامترهای آن برابر ۸ قرار داده و مقدار criterion آن را برابر gini_index می‌گذاریم و برای Performance (Classification) تیک گزینه‌های Accuracy و weighted mean recall و weighted mean precision را می‌زنیم. پس از اجرای پروسه مقادیر معیارهای کارایی در شکل ۲۳-۲ قابل مشاهده‌اند.



شکل ۲۰-۲



شکل ۲-۲۱



شکل ۲-۲۲

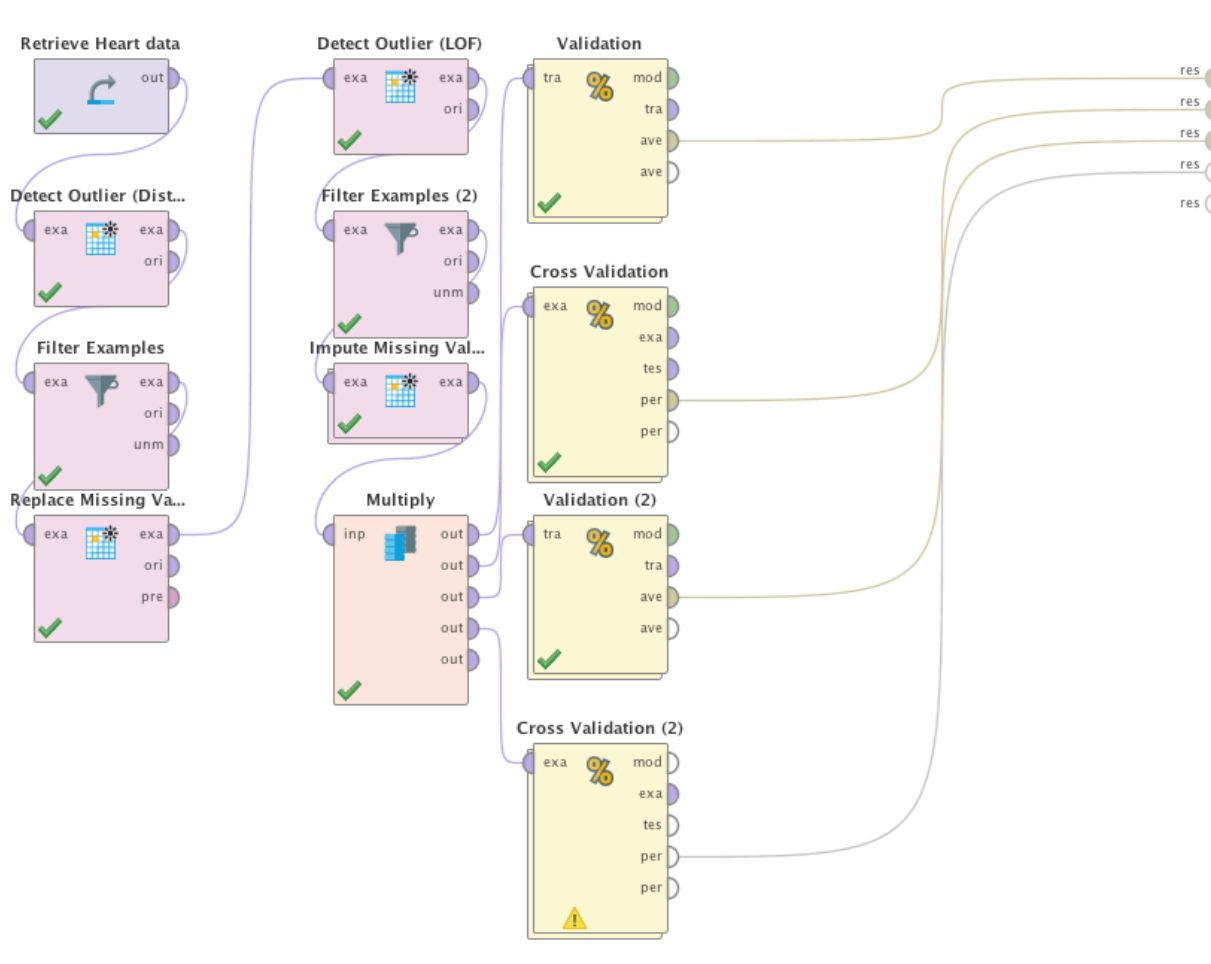
PerformanceVector

```
PerformanceVector:  
accuracy: 72.37%  
ConfusionMatrix:  
True:  0      1  
0:     78     28  
1:     14     32  
weighted_mean_recall: 69.06%, weights: 1, 1  
ConfusionMatrix:  
True:  0      1  
0:     78     28  
1:     14     32  
weighted_mean_precision: 71.58%, weights: 1, 1  
ConfusionMatrix:  
True:  0      1  
0:     78     28  
1:     14     32
```

شکل ۲-۲۳

حالت h

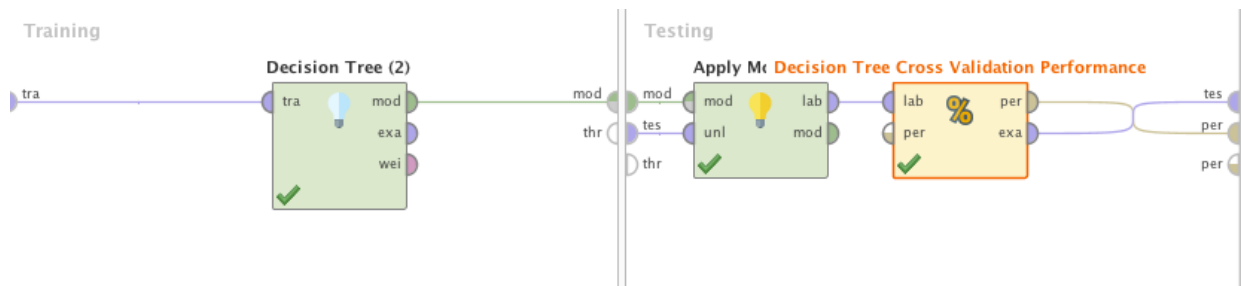
شبیه به حالت g ابتدا اپراتور Cross Validation را وارد پروسه کرده و مطابق شکل ۲-۲۴ متصل می‌کنیم. سپس مقادیر پارامترهای آن را مطابق شکل ۲-۲۵ تنظیم می‌کنیم. با دوبار کلیک روی آن وارد زیرپروسه شده و اپراتورهای Decision Tree و Apply Model و Performance (Classification) را وارد کرده و مطابق شکل ۲-۲۶ متصل می‌کنیم. برای Decision Tree مقدار maximal depth را در پارامترهای آن برابر ۸ قرار داده و مقدار criterion آن را برابر gini_index می‌گذاریم و برای Performance (Classification) تیک گزینه‌های Accuracy و weighted mean recall و weighted mean precision را می‌زنیم. پس از اجرای پروسه مقادیر معیارهای کارایی در شکل ۲-۲۷ قابل مشاهده‌اند.



شکل ۲-۲۴

Repository	Parameters
Decision Tree Cross Validation (Cross Validation)	
<input type="checkbox"/>	split on batch attribute
<input type="checkbox"/>	leave one out
number of folds	8
sampling type	stratified sampling
<input type="checkbox"/>	use local random seed
<input checked="" type="checkbox"/>	enable parallel execution

شکل ۲-۲۵



شکل ۲-۲۶

PerformanceVector

PerformanceVector:

accuracy: 74.94% +/- 7.24% (micro average: 74.90%)

ConfusionMatrix:

True: 0 1

0: 256 76

1: 51 123

weighted_mean_recall: 72.67% +/- 7.36% (micro average: 72.60%), weights: 1, 1

ConfusionMatrix:

True: 0 1

0: 256 76

1: 51 123

weighted_mean_precision: 74.11% +/- 7.76% (micro average: 73.90%), weights: 1, 1

ConfusionMatrix:

True: 0 1

0: 256 76

1: 51 123

شکل ۲-۲۷

بخش ۳

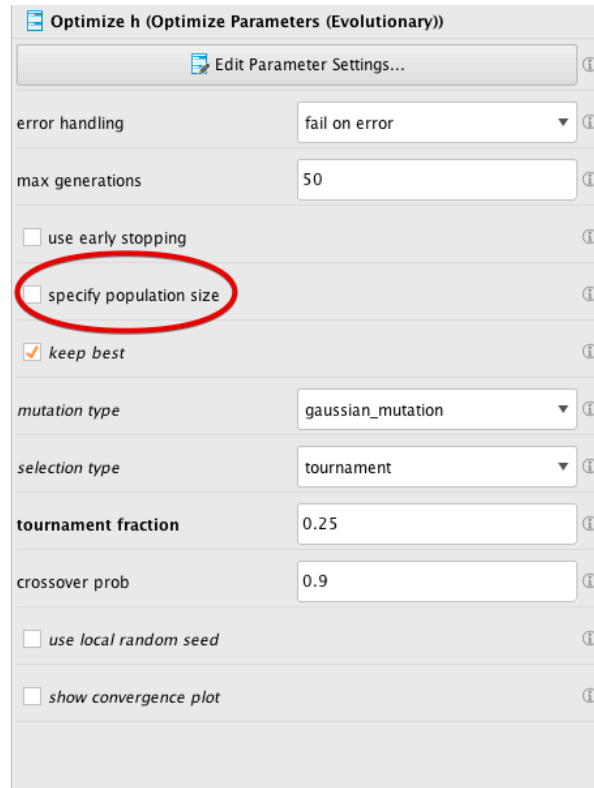
سوال ۳

حالت e

برای پیدا کردن بهترین پارامترها به منظور بیشینه کردن معیارهای کارایی (که برای ما اینجا سه معیار accuracy و recall و precision مد نظر است) رپیدمایر سه اپراتور در اختیار ما قرار می‌دهد. این سه اپراتور عبارتند از:

- Optimize Parameters (Grid)
- Optimize Parameters (Quadratic)
- Optimize Parameters (Evolutionary)

البته اپراتور دیگری نیز به نام Loop Parameters وجود دارد که با آن می‌توان تمام حالت‌های ممکن تمام پارامترها را تست کرد و بهترین جواب را بدست آورد که به دلیل ماهیت نمایی بودن زمان اجرای آن، برای تعداد زیادی پارامتر کار معقولی نیست و در ضمن این اپراتور بیشتر برای پارامترهایی که ماهیت انتخابی دارند مناسب است تا پارامترهای عددی مثل k در k -NN. طبق راهنمای خود رپیدمایر برای Optimize Parameters (Evolutionary)، این اپراتور در مواقعی که محدوده‌ی عددی پارامترها و وابستگی آن‌ها مشخص نیست، بهتر از دو اپراتور دیگر عمل می‌کند و لذا بهترین گزینه برای ما این اپراتور است. این اپراتور با الگوریتم‌های تکاملی بهترین ترکیب ممکن از تنظیمات پارامترها را برای ما پیدا می‌کند. تنظیمات پارامترهای خود این اپراتور را به صورت پیشفرض قرار می‌دهیم و فقط گزینه‌ی specify population size آن را غیرفعال می‌کنیم تا خود اپراتور میزان جمعیت اولیه‌ی الگوریتم را تعیین کند (شکل ۳-۱).



شکل ۳-۱

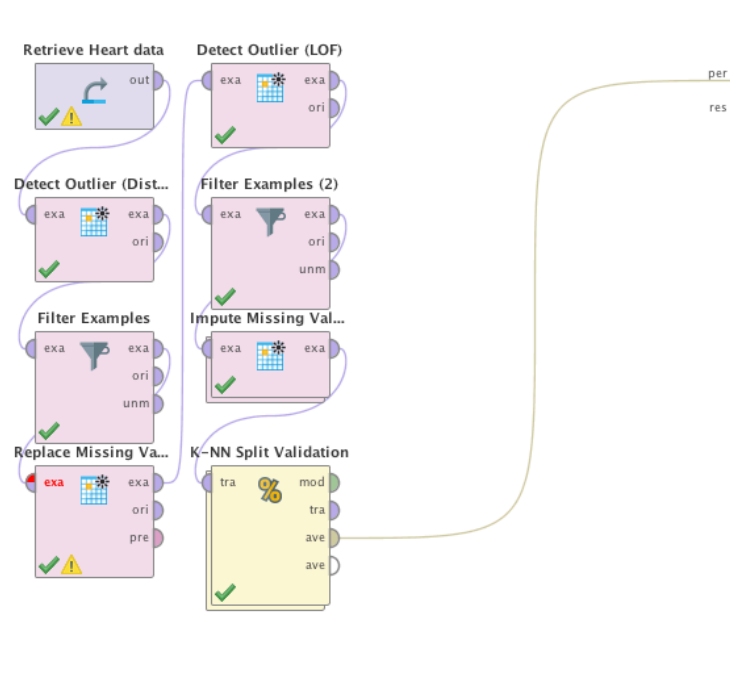
نحوهی کار این اپراتور به این گونه است که ابتدا پروسه‌ای را داخل زیرپروسه‌ی آن طراحی میکنیم و سپس یک خروجی از یک اپراتور **performance** به آن می‌دهیم و در تنظیمات این اپراتور تمام پارامترهایی از اپراتورهای زیرپروسه‌ی آن را که می‌خواهیم مقدار بهینه‌شان را پیدا کنیم، انتخاب می‌کنیم و سپس پروسه را اجرا می‌کنیم و در آخر این اپراتور مقادیر بهینه‌ی پارامترهای انتخاب شده را به ما می‌دهد.

برای هر یک از حالات **e** و **f** و **g** و **h** یک **Optimize Parameters (Evolutionary)** قرار میدهم که در ادامه به بررسی هریک از آن‌ها می‌پردازیم.

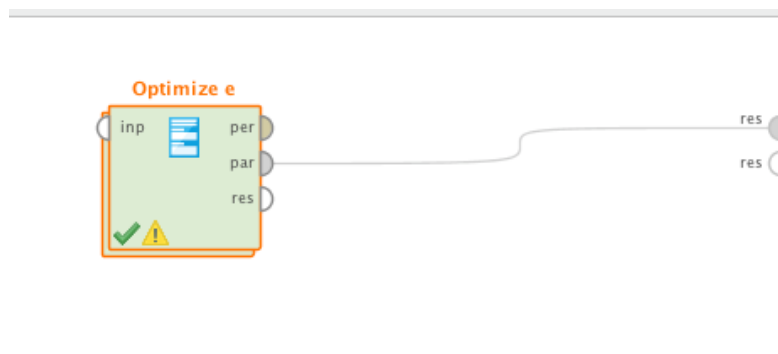
برای اجرای بخش ۳ و ۴ (بخش بهینه‌سازی) پروسه‌ی ریپدماینر جدیدی ایجاد می‌کنیم (فایل **project-optimization.rmp**) و این فایل جدای فایل پروسه‌ای است که در بخش ۲ آن را ساختیم. اما در این پروسه از چیزی که در بخش ۲ درست کردیم استفاده خواهیم کرد.

برای حالت **e** ابتدا یک اپراتور **Optimize Parameters (Evolutionary)** ایجاد کرده و وارد زیرپروسه‌ی اپراتور آن می‌شویم و مطابق شکل (۳-۲) دقیقاً پروسه‌ی که در بخش ۲ ایجاد کردیم را با این تفاوت که اپراتورهای **Cross Validation** و **Split Validation** که مربوط به بخش‌های **f** و **g** و **h** هستند را حذف می‌کنیم (در نتیجه می‌توانیم اپراتور **Multiply** را نیز حذف کنیم) و آن را داخل زیرپروسه **paste** می‌کنیم.

سپس همانطور که پیداست پروسه‌ی اصلی به شکل ۳-۳ درمی‌آید.

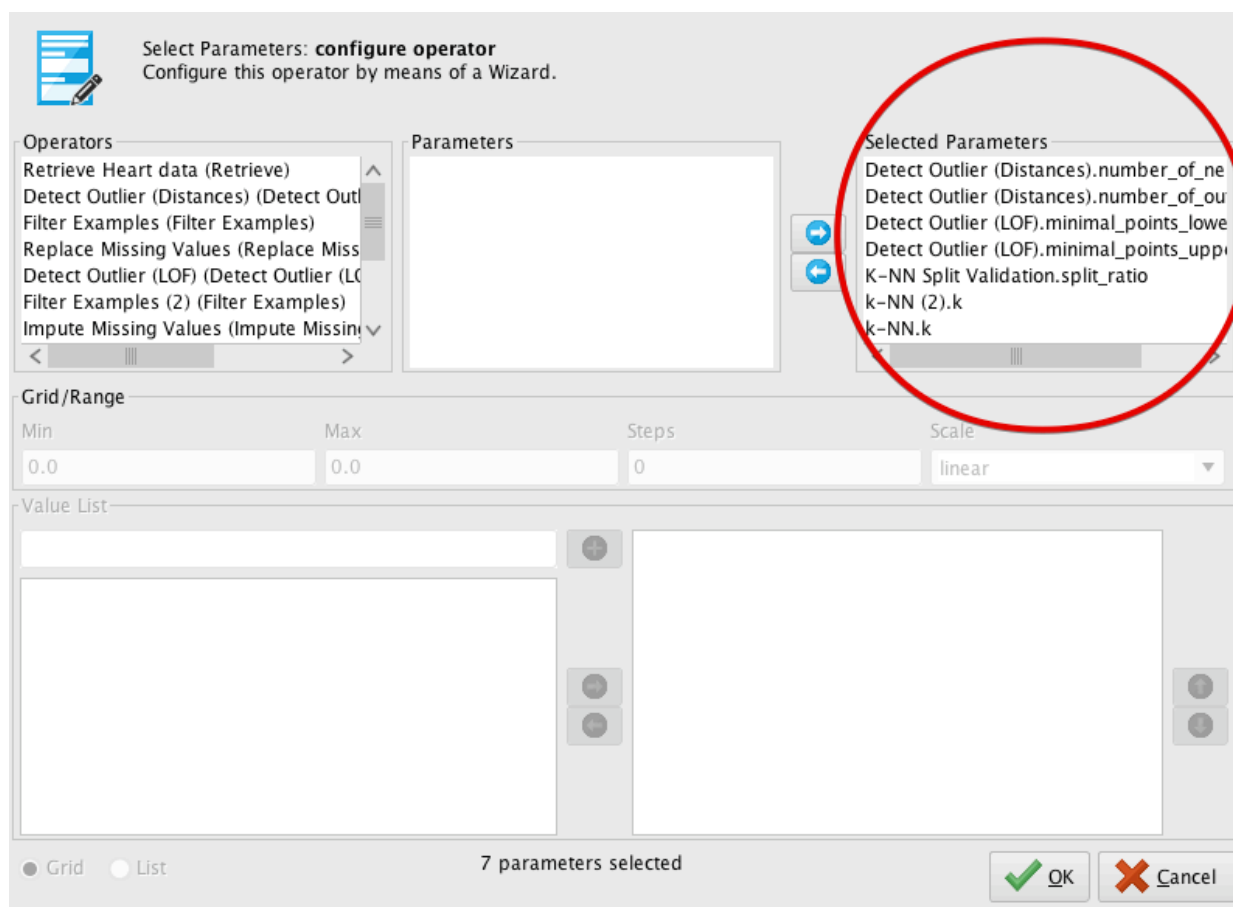


شکل ۲-۳



شکل ۳-۳

حال در بخش پارامترهای اپراتور بهینه‌سازی، روی **Edit Parameter Settings...** کلیک کرده و در این بخش پارامترهایی را که می‌خواهیم از زیرپروسه، بهینه کنیم، انتخاب می‌کنیم (شکل ۳-۴).



شکل ۳-۴

حال پروسه را اجرا می‌کنیم و چند دقیقه صبر می‌کنیم تا الگوریتم اجرا شده و خروجی بدهد (شکل ۳-۵).

ParameterSet

Parameter set:

Performance:

PerformanceVector [

-----accuracy: 70.37%

ConfusionMatrix:

True: 0 1
0: 33 15
1: 1 5

-----weighted_mean_recall: 61.03%, weights: 1, 1

ConfusionMatrix:

True: 0 1
0: 33 15
1: 1 5

-----weighted_mean_precision: 76.04%, weights: 1, 1

ConfusionMatrix:

True: 0 1
0: 33 15
1: 1 5

]

Detect Outlier (Distances).number_of_neighbors = 41

Detect Outlier (Distances).number_of_outliers = 55

Detect Outlier (LOF).minimal_points_lower_bound = 50

Detect Outlier (LOF).minimal_points_upper_bound = 77

K-NN Split Validation.split_ratio = 0.8917510333414714

k-NN (2).k = 52

k-NN.k = 81

شکل ۵-۳

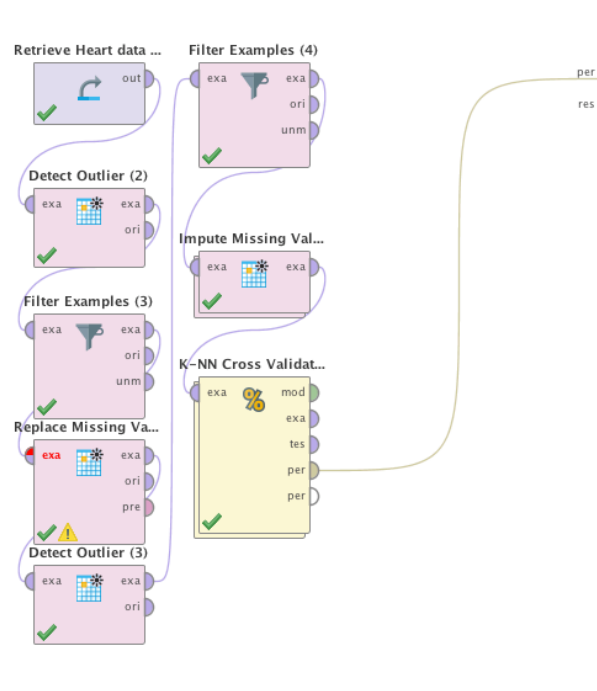
مقادیری که با بیضی قرمز مشخص شده‌اند، مقادیر بهینه‌ی معیارهای کارایی و مقادیری که با خط قرمز زیر آن‌ها مشخص شده‌اند، مقادیر بهینه‌ی پارامترها هستند.

پارامتر k-NN (2).k مربوط به مدل اصلی و k-NN.k مربوط به Impute Missing Value می‌باشد.

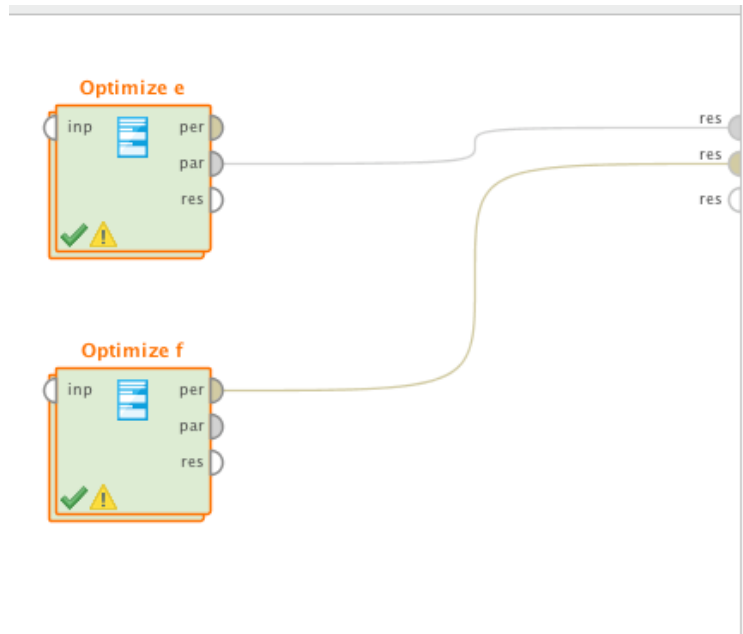
حالت f

برای حالت f شبیه به حالت e عمل می‌کنیم (برای خواندن بیشتر درمورد اپراتور بهینه‌سازی و تنظیمات آن به بخش ۳-e مراجعه کنید)، بدین شکل که ابتدا یک اپراتور **Optimize Parameters (Evolutionary)** ایجاد کرده و وارد زیرپروسه‌ی اپراتور آن می‌شویم و مطابق شکل (۳-۶) دقیقاً پروسه‌ی که در بخش ۲ ایجاد کردیم را با این تفاوت که اپراتورهای **Cross Validation** ها و **Split Validation** که مربوط به بخش‌های e و g و h هستند را حذف می‌کنیم (در نتیجه می‌توانیم اپراتور **Multiply** را نیز حذف کنیم) و آن را داخل زیرپروسه **paste** می‌کنیم.

سپس همانطور که پیداست پروسه‌ی اصلی به شکل ۳-۷ درمی‌آید.

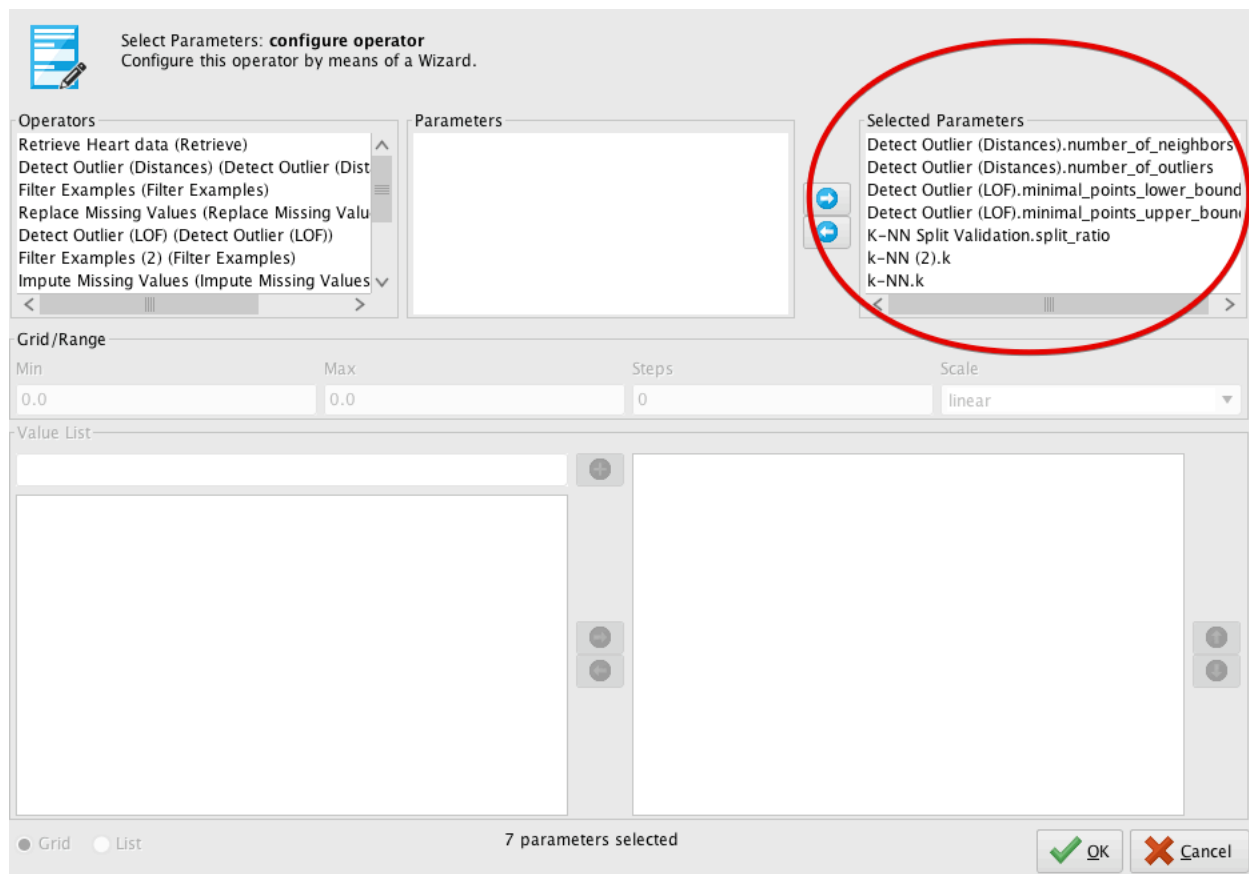


شکل ۳-۶



شکل ۷-۳

حال در بخش پارامترهای اپراتور بهینه‌سازی، روی **Edit Parameter Settings...** کلیک کرده و در این بخش پارامترهایی را که می‌خواهیم از زیرپروسه، بهینه کنیم، انتخاب می‌کنیم (شکل ۸-۳).



شکل ۸-۳

حال پروسه را اجرا می‌کنیم و چند دقیقه صبر می‌کنیم تا الگوریتم اجرا شده و خروجی بدهد (شکل ۹-۳).

ParameterSet

Parameter set:

Performance:

PerformanceVector [

-----accuracy 66.91% +/- 17.98% (micro average: 66.86%)

ConfusionMatrix:

True: 0 1

0: 241 103

1: 65 98

-----weighted_mean_recall 63.85% +/- 19.43% (micro average: 63.76%), weights: 1, 1

ConfusionMatrix:

True: 0 1

0: 241 103

1: 65 98

-----weighted_mean_precision: 63.76% +/- 23.15% (micro average: 65.09%), weights: 1, 1

ConfusionMatrix:

True: 0 1

0: 241 103

1: 65 98

]

Detect Outlier (2).number_of_neighbors = 48

Detect Outlier (2).number_of_outliers = 14

Detect Outlier (3).minimal_points_lower_bound = 5

Detect Outlier (3).minimal_points_upper_bound = 43

k-NN (3).k = 13

K-NN Cross Validation.number_of_folds = 78

k-NN (5).k = 10

شکل ۹-۳

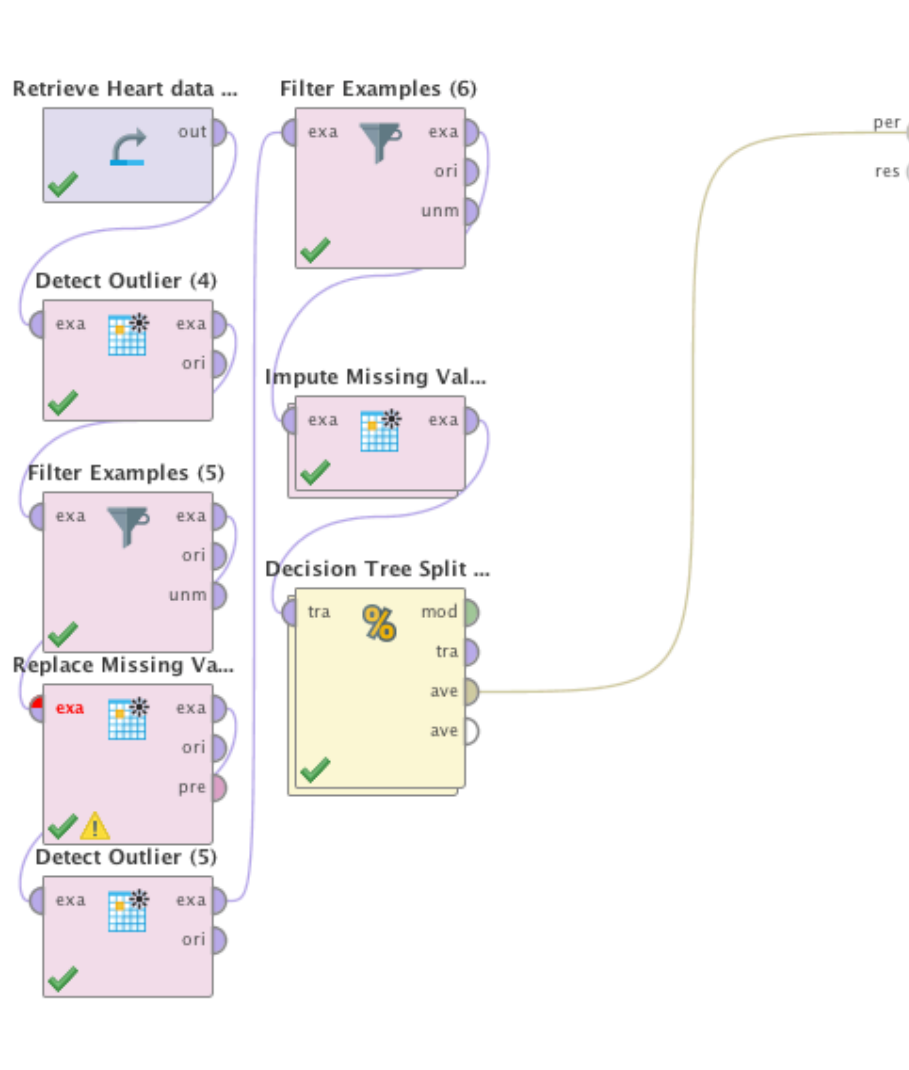
مقادیری که با بیضی قرمز مشخص شده‌اند، مقادیر بهینه‌ی معیارهای کارایی و مقادیری که با خط قرمز زیر آن‌ها مشخص شده‌اند، مقادیر بهینه‌ی پارامترها هستند.

پارامتر k -NN (5). k مربوط به مدل اصلی و k -NN (3). k مربوط به Impute Missing Value می‌باشد.

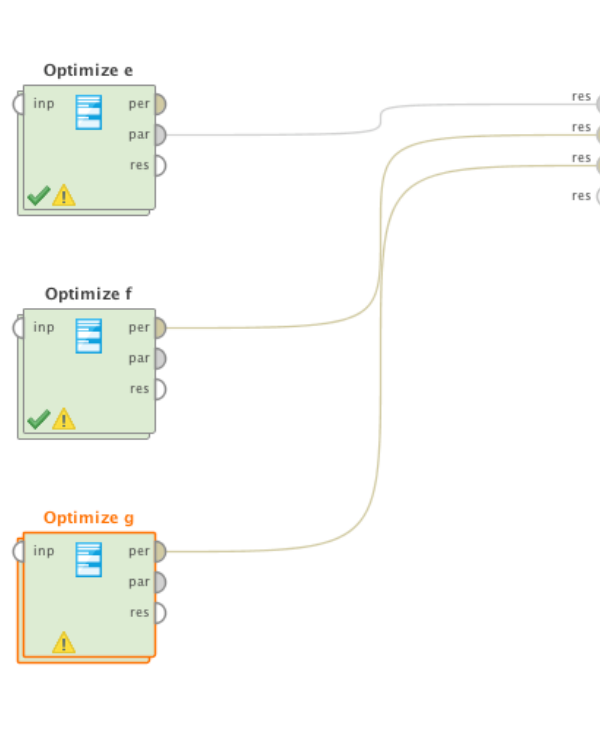
حالت g

برای حالت g شبیه به حالت e عمل می‌کنیم (برای خواندن بیشتر درمورد اپراتور بهینه‌سازی و تنظیمات آن به بخش e-۳ مراجعه کنید)، بدین شکل که ابتدا یک اپراتور **Optimize Parameters (Evolutionary)** ایجاد کرده و وارد زیرپروسی اپراتور آن می‌شویم و مطابق شکل (۳-۱۰) دقیقاً پروسی که در بخش ۲ ایجاد کردیم را با این تفاوت که اپراتورهای **Cross Validation** و **Split Validation** که مربوط به بخش‌های e و f و h هستند را حذف می‌کنیم (در نتیجه می‌توانیم اپراتور **Multiply** را نیز حذف کنیم) و آن را داخل زیرپروسی **paste** می‌کنیم.

سپس همانطور که پیداست پروسی اصلی به شکل ۳-۱۱ درمی‌آید.

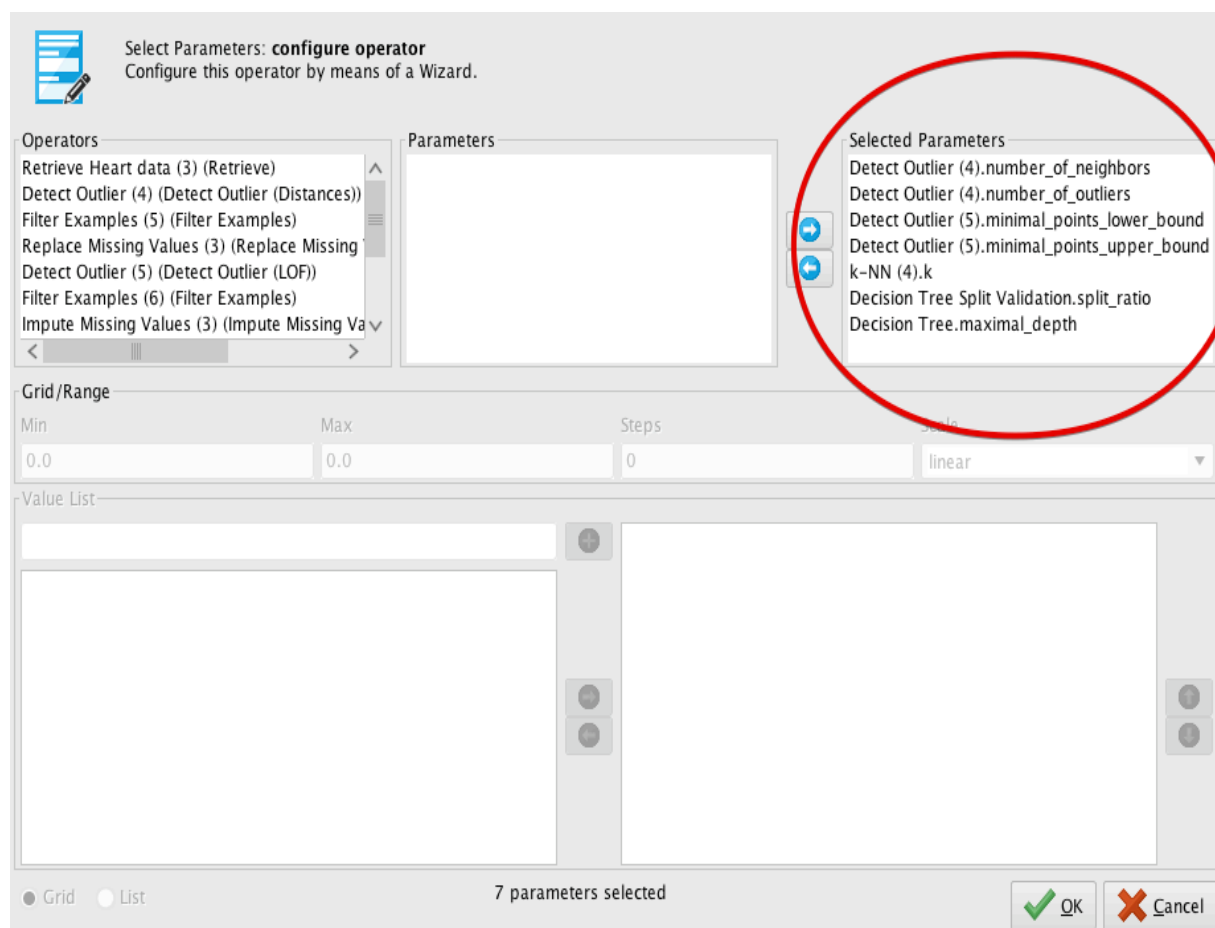


شکل ۳-۱۰



شکل ۱۱-۳

حال در بخش پارامترهای اپراتور بهینه‌سازی، روی **Edit Parameter Settings...** کلیک کرده و در این بخش پارامترهایی را که می‌خواهیم از زیرپروسه، بهینه کنیم، انتخاب می‌کنیم (شکل ۱۲-۳).



شکل ۱۲-۳

حال پروسه را اجرا می‌کنیم و چند دقیقه صبر می‌کنیم تا الگوریتم اجرا شده و خروجی بدهد (شکل ۱۳-۳).

ParameterSet

Parameter set:

Performance:

PerformanceVector [

-----accuracy: 78.81%

ConfusionMatrix:

True: 0 1

0: 293 91

1: 16 105

-----weighted_mean_recall: 74.20%, weights: 1, 1

ConfusionMatrix:

True: 0 1

0: 293 91

1: 16 105

-----weighted_mean_precision: 81.54% weights: 1, 1

ConfusionMatrix:

True: 0 1

0: 293 91

1: 16 105

]

Detect Outlier (4).number_of_neighbors = 81

Detect Outlier (4).number_of_outliers = 40

Detect Outlier (5).minimal_points_lower_bound = 46

Detect Outlier (5).minimal_points_upper_bound = 24

k-NN (4).k = 67

Decision Tree Split Validation.split_ratio = 0.03514528124153623

Decision Tree.maximal_depth = 13

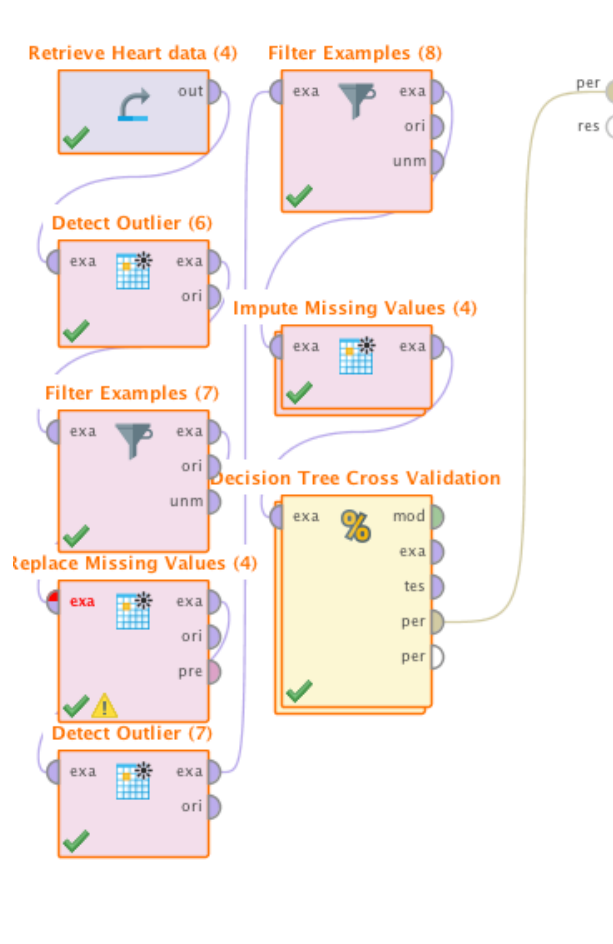
شکل ۳-۱۳

مقادیری که با بیضی قرمز مشخص شده‌اند، مقادیر بهینه‌ی معیارهای کارایی و مقادیری که با خط قرمز زیر آن‌ها مشخص شده‌اند، مقادیر بهینه‌ی پارامترها هستند.

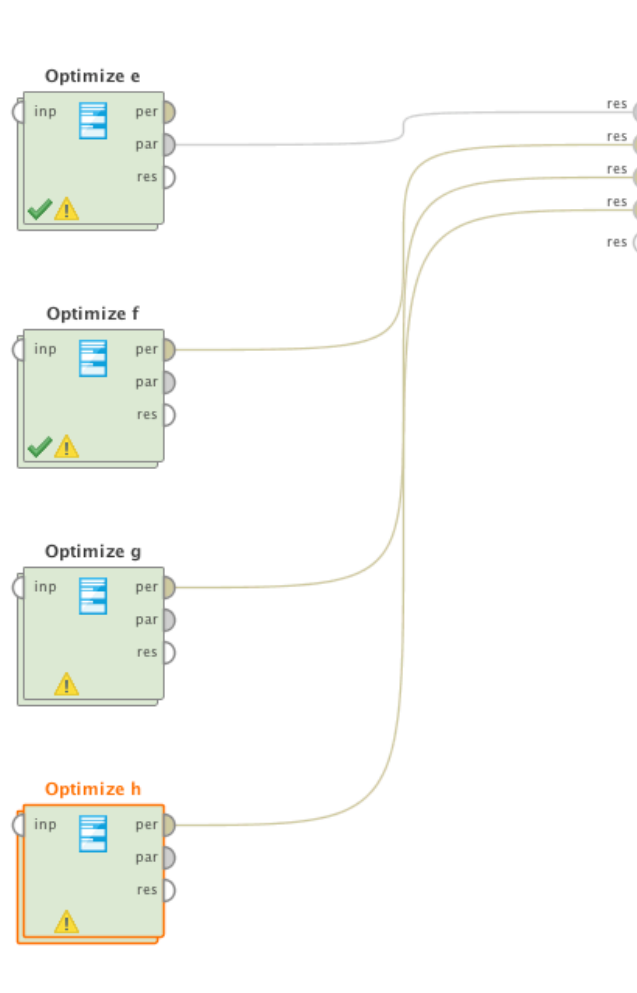
حالت h

برای حالت h شبیه به حالت e عمل می‌کنیم (برای خواندن بیشتر درمورد اپراتور بهینه‌سازی و تنظیمات آن به بخش ۳-e مراجعه کنید)، بدین شکل که ابتدا یک اپراتور **Optimize Parameters (Evolutionary)** ایجاد کرده و وارد زیرپروسی اپراتور آن می‌شویم و مطابق شکل (۳-۱۴) دقیقاً پروسی که در بخش ۲ ایجاد کردیم را با این تفاوت که اپراتورهای **Cross Validation** ها و **Split Validation** که مربوط به بخش‌های e و f و g هستند را حذف می‌کنیم (در نتیجه می‌توانیم اپراتور **Multiply** را نیز حذف کنیم) و آن را داخل زیرپروسی **paste** می‌کنیم.

سپس همانطور که پیداست پروسی اصلی به شکل ۳-۱۵ درمی‌آید.




شکل ۳-۱۴



شکل ۱۵-۳

حال در بخش پارامترهای اپراتور بهینه‌سازی، روی **Edit Parameter Settings...** کلیک کرده و در این بخش پارامترهایی را که می‌خواهیم از زیرپروسه، بهینه کنیم، انتخاب می‌کنیم (شکل ۱۶-۳).

 **Select Parameters: configure operator**
Configure this operator by means of a Wizard.

Operators
Retrieve Heart data (4) (Retrieve)
Detect Outlier (6) (Detect Outlier (Distances))
Filter Examples (7) (Filter Examples)
Replace Missing Values (4) (Replace Missing Values)
Detect Outlier (7) (Detect Outlier (LOF))
Filter Examples (8) (Filter Examples)
Impute Missing Values (4) (Impute Missing Values)

Parameters

Selected Parameters
Detect Outlier (6).number_of_neighbors
Detect Outlier (6).number_of_outliers
Detect Outlier (7).minimal_points_lower_bound
Detect Outlier (7).minimal_points_upper_bound
k-NN (6).k
Decision Tree Cross Validation.number_of_fold
Decision Tree (2).maximal_depth

Grid/Range

Min	Max	Steps	Scale
0.0	0.0	0	linear

Value List

☒ Grid ☐ List

7 parameters selected

☒ OK ☐ Cancel

شکل ۳-۱۶

حال پروسه را اجرا می‌کنیم و چند دقیقه صبر می‌کنیم تا الگوریتم اجرا شده و خروجی بدهد (شکل ۳-۱۷).

ParameterSet

Parameter set:

Performance:

PerformanceVector [

-----accuracy: 77.26% +/- 6.10% (micro average: 77.26%)

ConfusionMatrix:

True: 0 1

0: 265 65

1: 56 146

-----weighted_mean_recall: 75.82% +/- 6.35% (micro average: 75.87%), weights: 1, 1

ConfusionMatrix:

True: 0 1

0: 265 65

1: 56 146

-----weighted_mean_precision: 76.90% +/- 6.35% (micro average: 76.29%), weights: 1, 1

ConfusionMatrix:

True: 0 1

0: 265 65

1: 56 146

]

Detect Outlier (6).number_of_neighbors = 97

Detect Outlier (6).number_of_outliers = 19

Detect Outlier (7).minimal_points_lower_bound = 35

Detect Outlier (7).minimal_points_upper_bound = 66

k-NN (6).k = 79

Decision Tree Cross Validation.number_of_folds = 14

Decision Tree (2).maximal_depth = 60

شکل ۳-۱۷

مقادیری که با بیضی قرمز مشخص شده‌اند، مقادیر بهینه‌ی معیارهای کارایی و مقادیری که با خط قرمز زیر آن‌ها مشخص شده‌اند، مقادیر بهینه‌ی پارامترها هستند.

پایان.

