



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

عنوان:

پروژه نرم افزار ریدمایندر

نگارنده:

پدرام پیرو اصفیا - ۹۸۲۵۰۰۶

استاد: دکتر فاطمه شاکری

درس: داده کاوی

بهار ۱۴۰۲

فهرست مطالب

۱. تمرین اول.....	۳
۱.۱. بررسی داده‌های گمشده.....	۳
۱.۲. نحوه رفتار با داده‌های پرت.....	۴
۱.۳. نرمالایز کردن داده‌ها.....	۷
۱.۴. شناسایی داده‌های پرت.....	۸
۲. تمرین دوم: پیش‌بینی بیماران قلبی.....	۹
۲.۱. پیش‌پردازش دیتاست.....	۹
۲.۱.۱. تغییر تایپ ویژگی‌ها.....	۹
۲.۱.۲. نرمالایز کردن ویژگی‌ها.....	۱۱
۲.۱.۳. آماده‌سازی ویژگی‌ها برای مدل.....	۱۵
۲.۲. پیاده‌سازی مدل‌ها.....	۱۶
۲.۲.۱. تقسیم دیتاست به دو بخش.....	۱۶
۲.۲.۲. پیاده‌سازی 5-fold cross validation.....	۱۸
۲.۳. بررسی نتایج و ارزیابی.....	۱۹
۲.۴. بررسی اسکترپلات ویژگی‌ها.....	۲۱
۳. منابع.....	۲۳

۱. تمرین اول

۱.۱. بررسی داده‌های گمشده

ابتدا داده‌ها را از بخش **data>Sample** وارد بخش **Design** می‌کنیم. سپس با اجرا مدل و دیدن بخش **statistics**، سعی می‌کنیم فهم کلی از فیچرها بدست آوریم و ببینیم که برای هر کدام بهترین روش جاگذاری مقادیر گمشده چیست. در ادامه نمایی از این بخش آمده است، هرچند که در یک جدول اطلاعات به صورت خلاصه تر و بررسی بهتر آورده شده است:

Name	Type	Missing	Statistics	Filter (17 / 17 attributes):
Label class	Nominal	0	Least bad (14)	Most good (26) Values good (26), bad (14)
duration	Integer	1	Min 1	Max 3 Average 2.103
wage-inc-1st	Real	1	Min 2	Max 6.900 Average 3.621
wage-inc-2nd	Real	10	Min 2	Max 7 Average 3.913
wage-inc-3rd	Real	28	Min 2	Max 5.100 Average 3.767
col-adj	Nominal	16	Least tcf (4)	Most none (14) Values none (14), tc (6), ...[1 more]
working-hours	Integer	3	Min 27	Max 40 Average 37.811

تصویر ۱ - بررسی داده‌های گمشده

اطلاعات خلاصه شده در زیر آمده است:

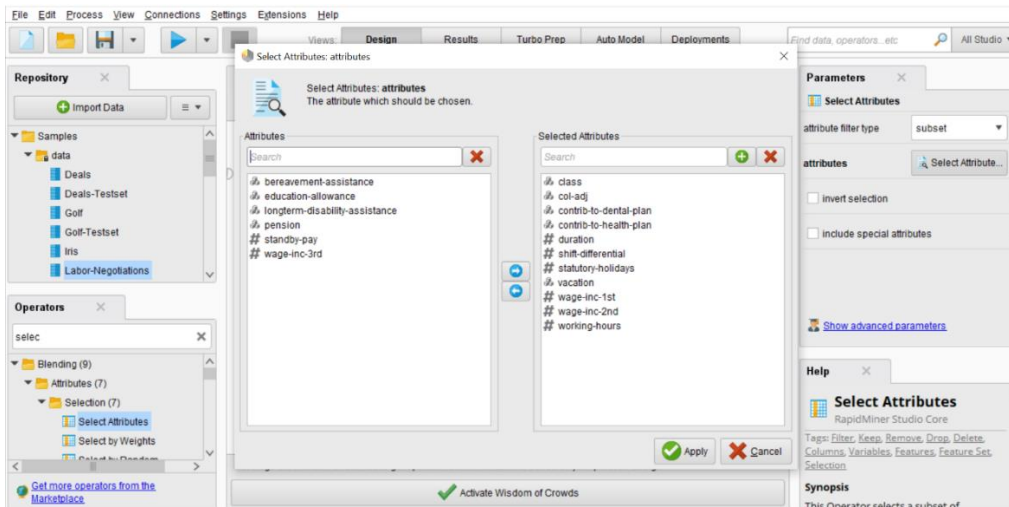
جدول ۱ - تعداد داده‌های گمشده به ازای هر ویژگی

نام ویژگی	تعداد داده‌های گمشده
Class	0
Duration	1
Wage-inc-1 st	1
Wage-inc-2 nd	10
Wage-inc-3 rd	28
Col-adj	16
Working-hours	3

Pension	22
Standby-pay	33
Shift-differential	16
Education-allowancee	22
Statutory-holidays	2
Vacation	3
Longterm-disability-assistance	24
Contrib-to-dental-plan	15
Bereavement-assistance	20
Contrib-to-health-plan	16

۱.۲. نحوه رفتار با داده‌های پرت

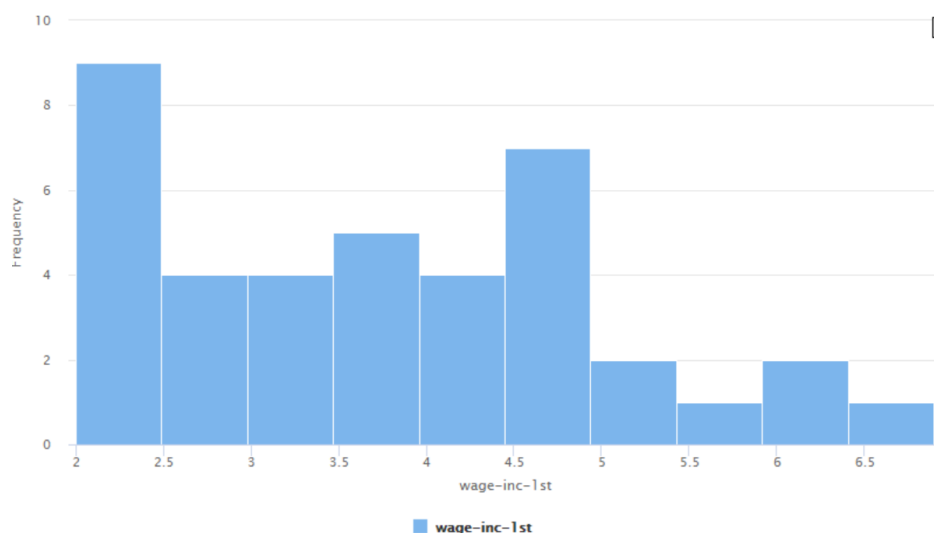
از آنجایی که تعداد کل رکوردها ۴۰ تاست، نگهداشتن برخی ستون‌ها که تعداد زیادی داده گم شده دارند نه تنها کمکی نم‌کیند، بلکه جاگذاری مقدار آن‌ها باعث میشود که در تصمیم‌گیری‌های آتیماں دچار خطا شویم به علت bias زیادی که در داده‌ها ایجاد می‌شود. به همین علت با توجه به (van Buuren, 2011)، یک آستانه ۵۰٪ در نظر گرفتیم که اگر تعداد داده‌های گم شده بیشتر از ۲۰ تا بود، آن ستون را حذف کنیم. با توجه به این موضوع ستون‌های Wage-inc-3rd، Pension، Standby-pay، Education-allowancee، Longterm-disability-assistance، Bereavement-assistance حذف میشوند. برای این کار از بخش operators>Selection، ویژگی‌های مد نظر را انتخاب میکنیم. سپس مدل را اجرا می‌کنیم. در تصویر ۲، بخش انتخاب ویژگی‌ها را میبینیم.



تصویر ۲- انتخاب ویژگی‌های مد نظر

توجه داشته باشید با توجه به این که تعداد رکوردهایی که داریم خیلی کم هستند، هر کدام از رکوردها اهمیت زیادی برای ما دارد (و مثل حالتی نمیباشد که ۱۰۰۰۰۰ داده داشته باشیم و از دست دادن ۱۰۰۰ تا از آن ها خیلی برایمان اهمیت نداشته باشد). به همین علت حذف کردن رکوردی که حاوی داده گمشده است، به نظر رفتار مناسبی نمی‌آید. تنها راه باقی مانده این است که داده‌های گمشده را با میانگین/میانه/مد پر کنیم.

- ستون duration حاوی داده‌های categorical می‌باشد که به همین دلیل مقدار مناسبش را با مد پر می‌کنیم.
- ستون Wage-inc-1st به نوعی skewed است، به همین علت با میانه پر می‌شود. توجه کنید که نمودار آن در زیر آمده است. (از آوردن نمودارهای ویژگی‌های دیگر خودداری شده است.)



تصویر ۳- کشیدگی در wage-inc-1st

- ستون Wage-inc-2nd به نظر شبه گاوسی می‌آید به همین علت با میانگین پر می‌شود.

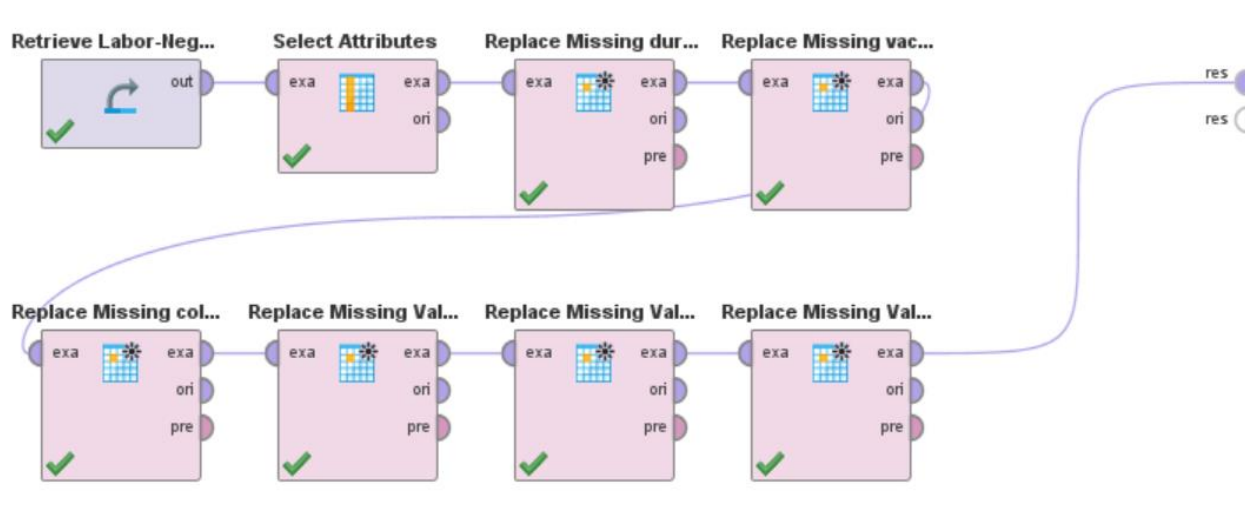
- ستون col-adj حاوی داده‌های categorical می‌باشد که به همین دلیل مقدار مناسبش را با مد پر می‌کنیم.
- ستون working-hours به نوعی skewed است، به همین علت با میان پر می‌شود.
- ستون shift-differential به نوعی skewed است، به همین علت با میان پر می‌شود.
- ستون Statutory-holidays نیز نشاندهنده روز است پس به نوعی کتگوریکال است و باید با مد پر شود.
- ستون vacation حاوی داده‌های categorical می‌باشد که به همین دلیل مقدار مناسبش را با مد پر می‌کنیم.
- ستون contrib-to-dental-plan حاوی داده‌های categorical می‌باشد که به همین دلیل مقدار مناسبش را با مد پر می‌کنیم.
- ستون contrib-to-health-plan حاوی داده‌های categorical می‌باشد که به همین دلیل مقدار مناسبش را با مد پر می‌کنیم.

ابتدا با ستون‌هایی که باید با مد پر شوند شروع می‌کنیم. از بخش operators>Cleansing>Replace Missing Values انتخاب می‌کنیم. از آنجایی که به صورت خودکار نمی‌توان مد را جاگذرای کرد، به صورت دستی این کار را انجام می‌دهیم و با انتخاب کردن value از آپشن‌های موجود، خودمان مقدار مشخص می‌کنیم.

جدول ۲ - مقدار مد برای ستون‌ها

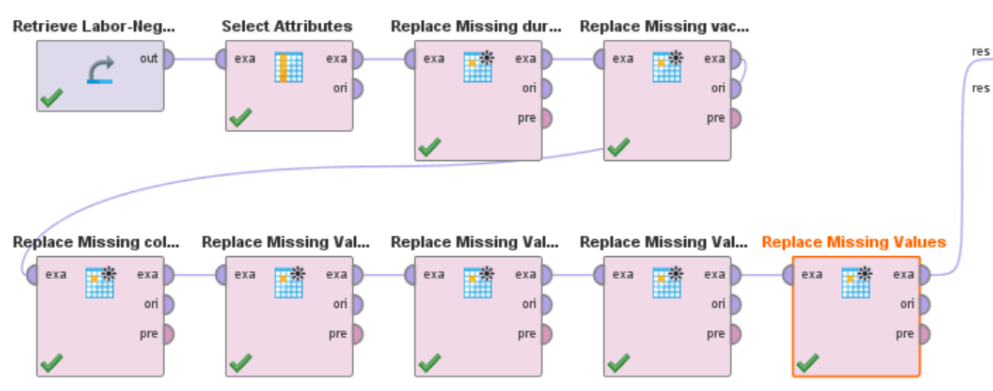
مقدار <u>مد</u>	نام ویژگی
2	Duration
none	Col-adj
below-average vacation	Vacation
half	Contrib-to-dental-plan
full	Contrib-to-health-plan
11	Statutory-holidays

بعد از اضافه کردن اپراتورها خواهیم داشت:



تصویر ۴- اضافه کردن اپراتورهای مربوط به جاگذاری مد

حال از انجایی که در رپیدمایتر قابلیت جاگذاری میانه وجود نداره، هم فیچرهایی که می‌خواستیم با میانه جاگذاری کنیم و هم آن‌هایی که می‌خواستیم با میانگین جاگذاری کنیم را با میانگین جاگذاری می‌کنیم.

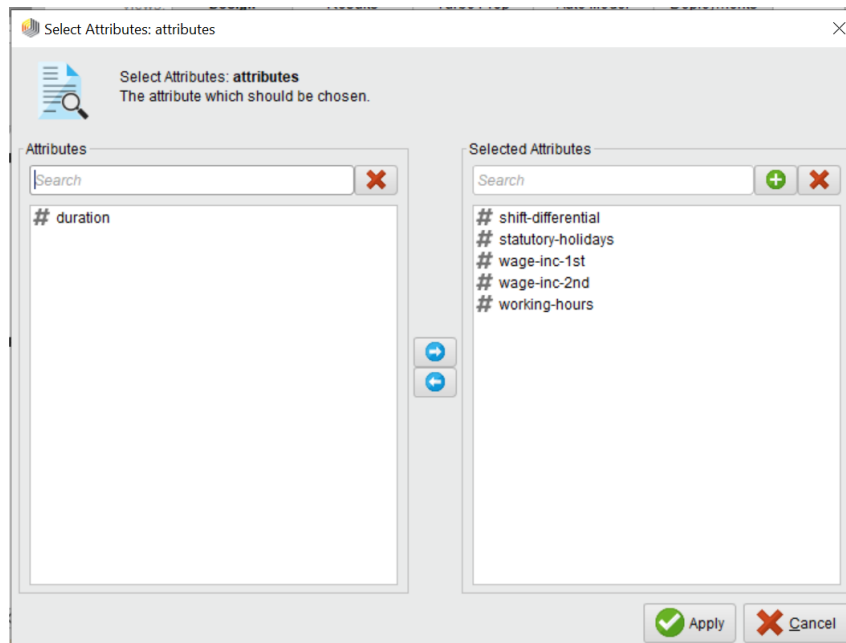


تصویر ۵- اضافه کردن اپراتورهای مربوط به جاگذاری میانگین

همچنین از تب Results میتوانیم چک کنیم که آیا داده گمشده باز موجود است یا خیر.

۱.۳. نرمالایز کردن داده‌ها

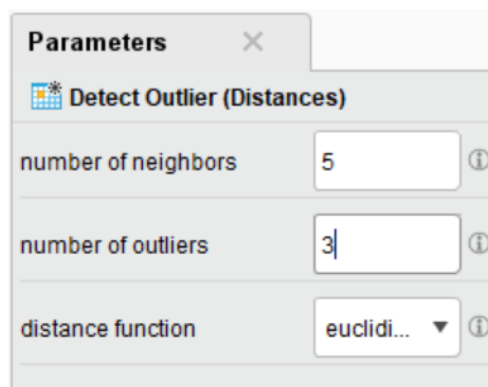
در گام بعد باید داده هارا نرمالایز کنیم. برای این کار `Cleansing>Normalization>Normalize` را انتخاب میکنیم و تنها داده‌هایی که `Categorical` نیستند را انتخاب میکنیم که شامل فیچرهای زیر می‌شود.



تصویر ۶- انتخاب ویژگی‌های مناسب نرمالایز کردن

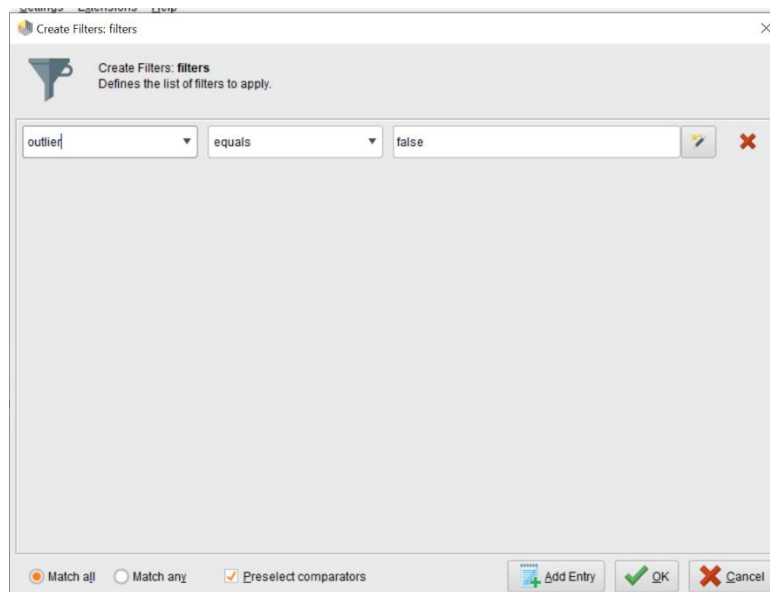
۱.۴. شناسایی داده‌های پرت

بعد از نرمالایز کردن باید داده‌های پرت را شناسایی کنیم. بدین منظور از اپراتور detect outlier استفاده می‌کنیم و فقط ۳ رکوردی که ممکن است پرت باشند را با ۵ همسایگی و فاصله اقلیدسی سعی می‌کنیم شناسایی کنیم.



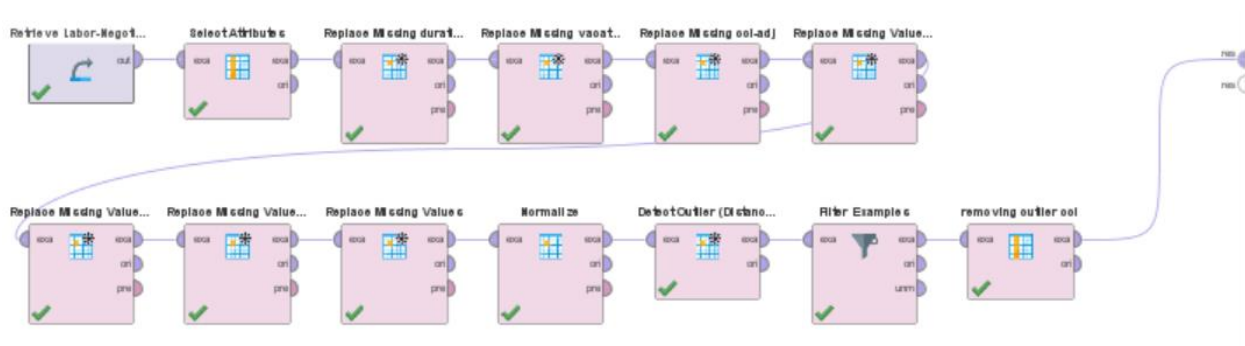
تصویر ۷- انتخاب پارامترهای مربوط به شناسایی داده پرت

در ادامه عملگر filter example را قرار می‌دهیم تا داده‌های پرتی که شناسایی کردیم را از دیتاست حذف کنیم:



تصویر ۸- شرط مربوط به فیلتر کردن داده های پرت

حال ستون outlier را نیز حذف میکنیم و دیتاستمان از مرحله پیش پردازش نیز عبور کردند و آماده پیاده سازی مدل است.



تصویر ۹- شمای کامل پروسه پاکسازی

۲. تمرین دوم: پیشبینی بیماران قلبی

۲.۱. پیش پردازش دیتاست

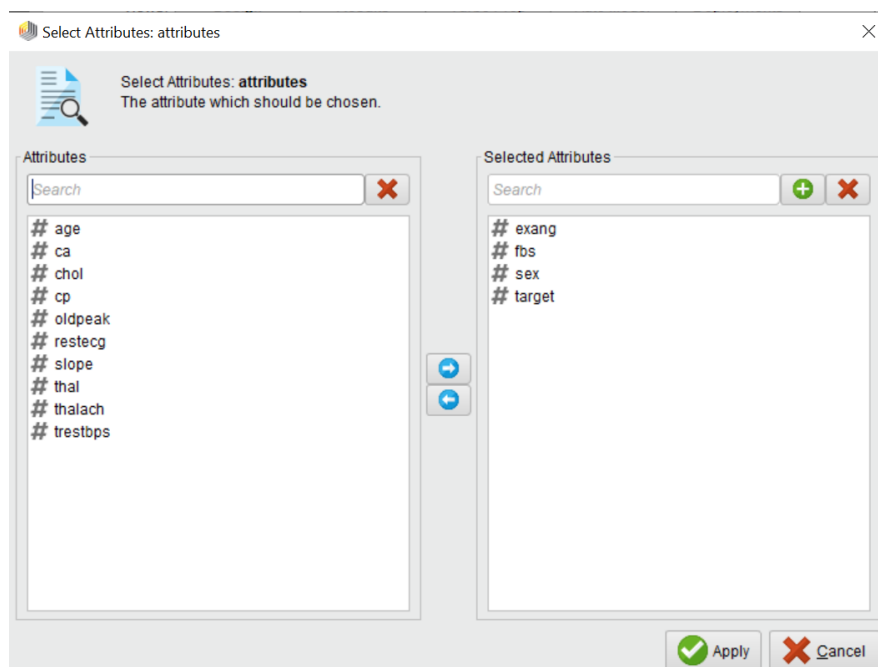
۲.۱.۱. تغییر تایپ ویژگی ها

در گام اول دیتاست را از file>Import Data می‌کنیم. سپس با توجه به meta data ای که حاضر شده است، باید تایپ برخی ستون‌ها را تغییر دهیم و از حالت int به categorical تبدیل کنیم. حالت کنتگوریکال در رپیدماینر به دو بخش binomial و polynomial تقسیم میشود. تبدیلات باید به صورت زیر انجام شود:

جدول ۳- تغییر تایپ ستون‌ها

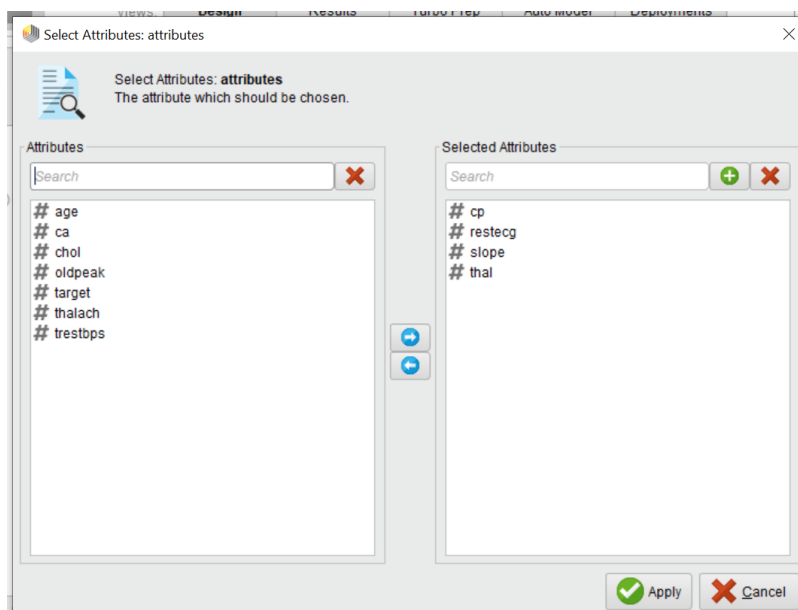
تایپ ویژگی	نام ویژگی
binomial	sex
polynomial	cp
binomial	fbs
polynomial	restecg
binomial	exang
polynomial	slope
polynomial	thal
binomial	target

به همین دلیل از بخش اپراتورها، Blending>Types>Numerical to Binomial تغییرات را برای ستون‌های مشخص شده انجام می‌دهیم.



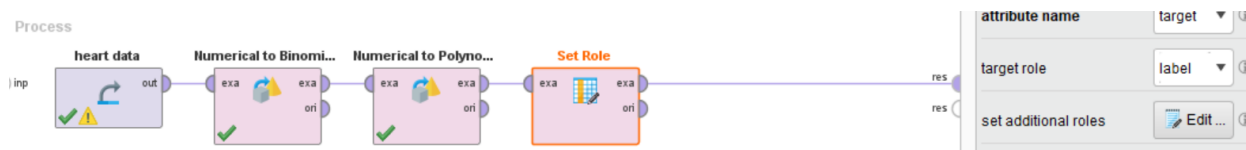
تصویر ۱۰- تبدیل تایپ به باینومیال

همین کار را از بخش Blending>Types>Numerical to Polynomial می‌دهیم. ۳ انجام



تصویر ۱۱- تبدیل تایپ به پولینومیل

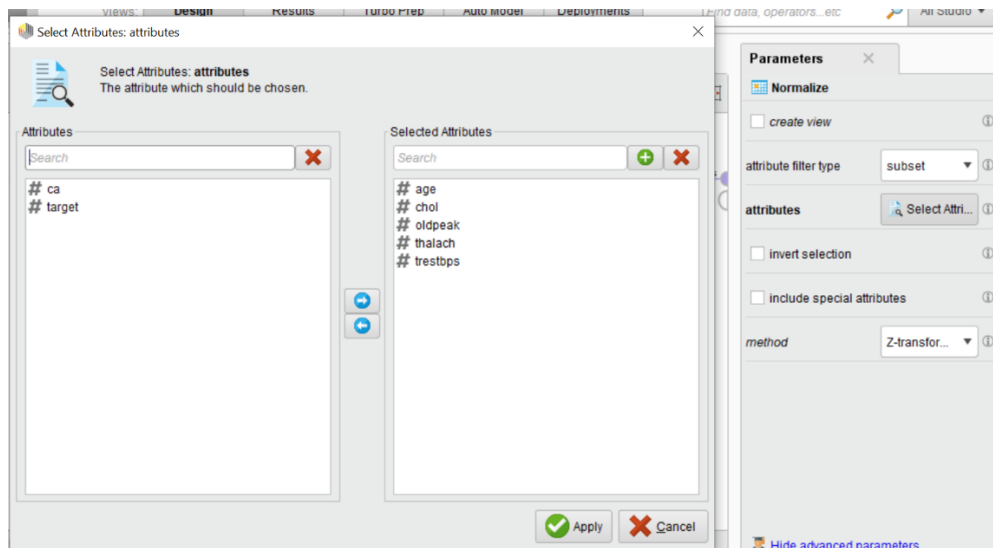
همچنین باید ستونی که تارگت است را برای پیشبینی نقشش را عوض کنیم:



تصویر ۱۲- تغییر نقش ستون تارگت

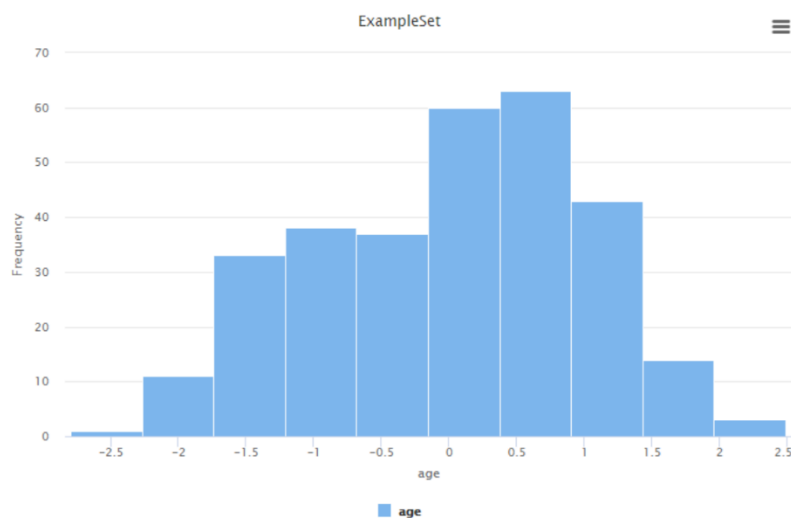
۲.۱.۲. نرمالایز کردن ویژگی‌ها

در ادامه باید داده‌های numerical را نرمالایز کنیم. از آنجایی که در تمرین قبل این پروسه توضیح داده شده است، از توضیح مجدد آن خودداری کرده و صرفاً پیاده سازی می‌کنیم. برای نرمالایز کردن ستون‌های زیر را انتخاب می‌کنیم.



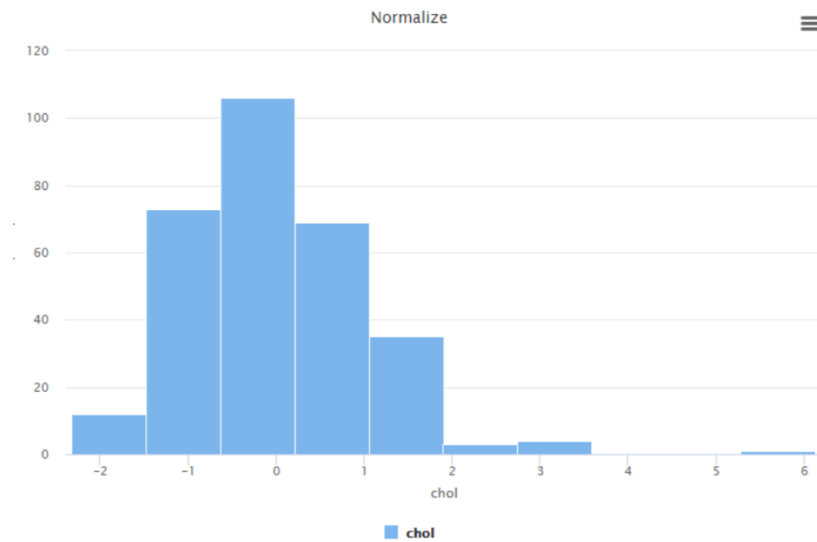
تصویر ۱۳- نرمالایز کردن ستون‌های نومریکال

در گام بعدی به دنبال داده‌های پرت می‌گردیم. در اینجا مهم است که تعداد این رکوردها را به درستی مشخص کنیم. برای این کار به هیستوگرام ویژگی‌هایی که نرمالایز کردیم نگاه می‌کنیم و تک تک آن‌ها را بررسی می‌کنیم.



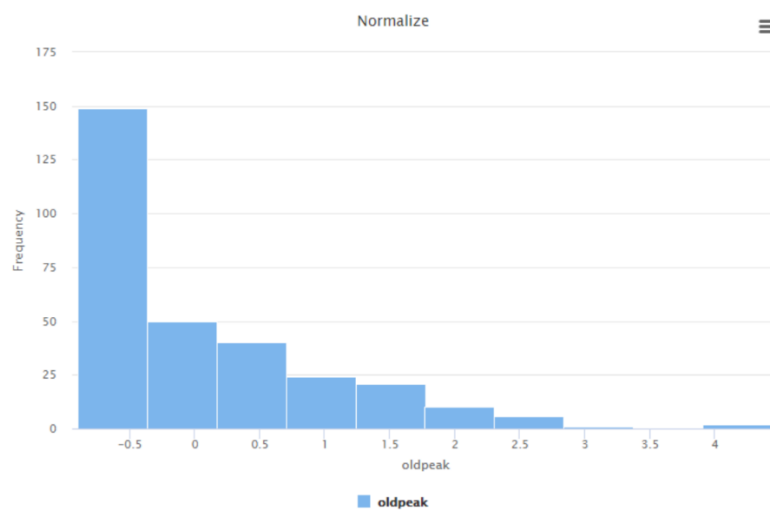
تصویر ۱۴- هیستوگرام ستون age

با توجه به اینکه داده‌های age در بازه‌ی ۳- تا ۳ قرار دارند، به نظر نمی‌آید outlier وجود داشته باشد.



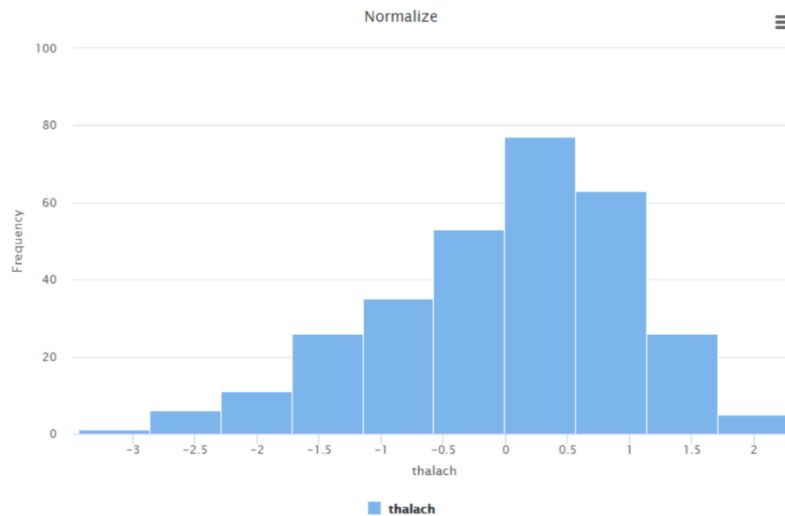
تصویر ۱۵- هیستوگرام ستون chol

می‌توان دید که در ستون chol حدوداً ۵ رکورد بیشتر از ۳ هستند و می‌توان به عنوان داده پرت در نظرشان گرفت.



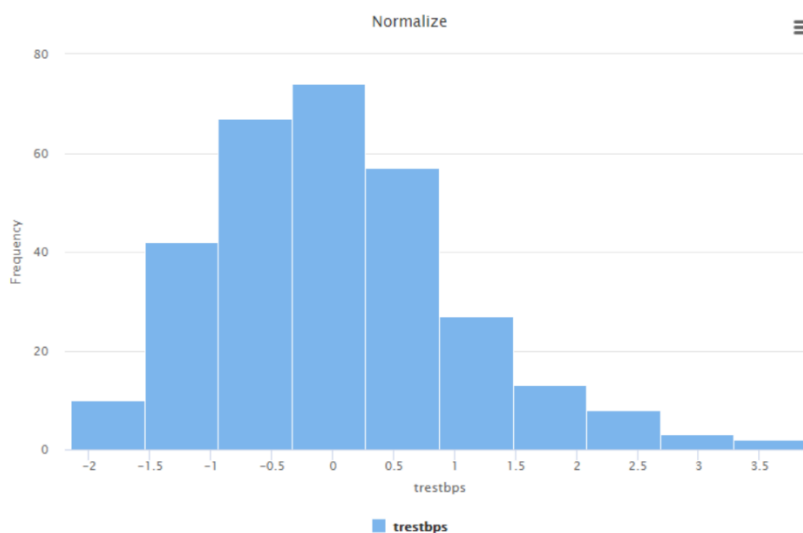
تصویر ۱۶- هیستوگرام ستون oldpeak

می‌توان دید که در ستون oldpeak حدوداً ۳ رکورد بیشتر از ۳ هستند و می‌توان به عنوان داده پرت در نظرشان گرفت.



تصویر ۱۷- هیستوگرام ستون thalach

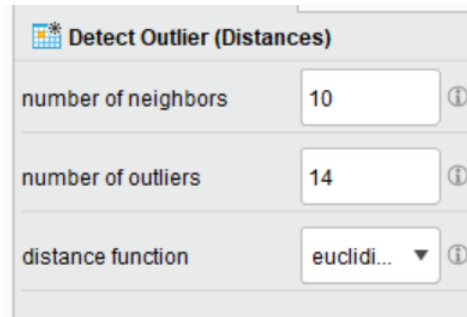
می‌توان دید که در ستون thalach یک رکورد کمتر از ۳- است و می‌توان به عنوان داده پرت در نظر گرفت.



تصویر ۱۸- هیستوگرام ستون trestbps

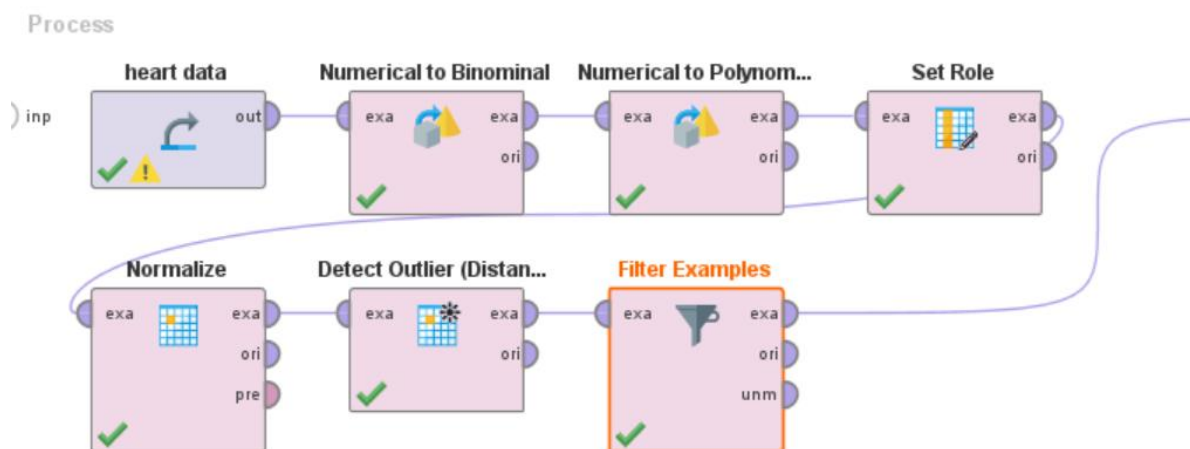
می‌توان دید که در ستون trestbps حدودا ۵ رکورد بیشتر از ۳ هستند و می‌توان به عنوان داده پرت در نظرشان گرفت.

پس مجموعا می‌توان گفت در بدترین حالت ۱۴ رکورد داریم که outlier هستند. به همین دلیل پارامتر مربوط به شناسایی داده‌های پرت را که در تمرین قبل بررسی کردیم، برابر با ۱۴ قرار می‌دهیم.



تصویر ۱۹- پارامترهای شناسایی outlier ها

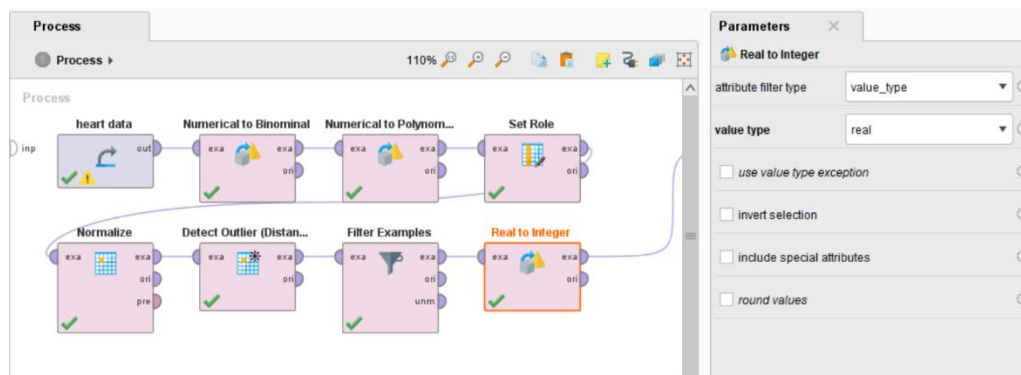
سپس داده‌ها را فیلتر می‌کنیم تا outlier ها در تحلیل نباشند:



تصویر ۲۰- پروسه تا آخر حذف کردن داده‌های پرت

۲.۱.۳. آماده‌سازی ویژگی‌ها برای مدل

همچنین برای پیاده‌سازی مدل‌ها نیاز است تا ویژگی‌ها از real به integer تبدیل شوند و ستون outlier نیز حذف شود.



تصویر ۲۱ - تبدیل real به integer

۲.۲. پیاده سازی مدل ها

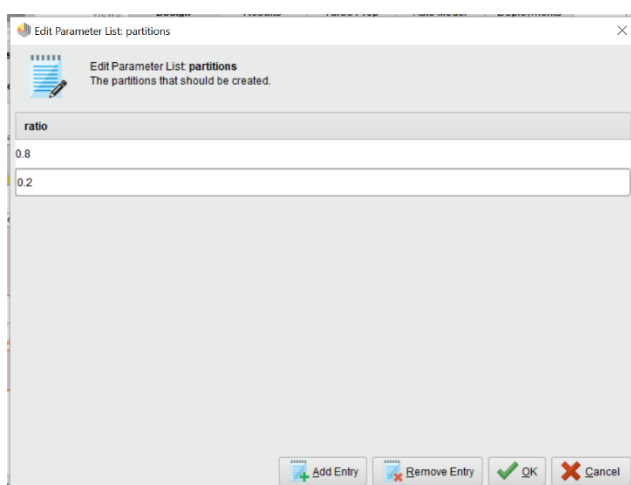
حال باید دو پروسه را پی بگیریم.

۱. تقسیم دیتاست به دو بخش ۸۰-۲۰

۲. پیاده سازی k-fold cross validation

۲.۲.۱. تقسیم دیتاست به دو بخش

ابتدا تقسیم ۸۰-۲۰ را انجام می دهیم:

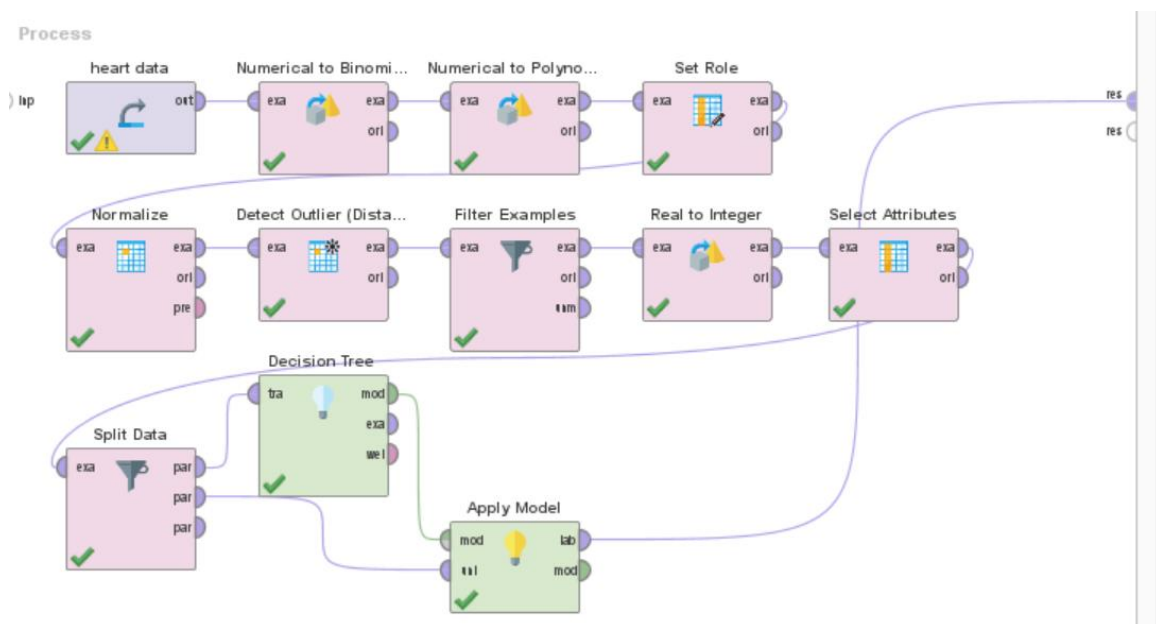


تصویر ۲۲- تقسیم ۸۰-۲۰ دیتاست

سپس مدل decision tree را با پارامتر $\text{maximal depth}=7$ و دیگر پارامترهای دیفالت (مطابق تصویر ۲۳) به همراه apply model اضافه می کنیم.

criterion	gain_ratio ▼ ⓘ
maximal depth	7 ⓘ
<input checked="" type="checkbox"/> apply pruning	ⓘ
confidence	0.1 ⓘ
<input checked="" type="checkbox"/> apply prepruning	ⓘ
minimal gain	0.01 ⓘ
minimal leaf size	2 ⓘ
<i>minimal size for split</i>	4 ⓘ
<i>number of prepruning alternatives</i>	3 ⓘ

تصویر ۲۳- پارامترهای decision tree



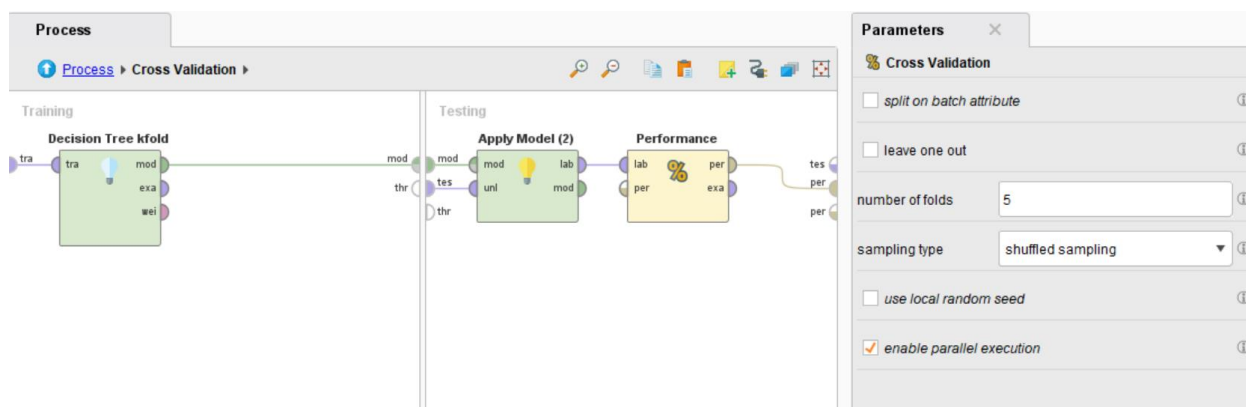
تصویر ۲۴- پروسه بعد از اضافه کردن decision tree

Row No.	target	prediction(L...	confidence(f...	confidence(L...	trestbps	chol	thalach	oldpeak	age
1	true	true	0.049	0.951	0	0	0	0	0
2	true	true	0.049	0.951	0	0	0	0	0
3	true	true	0.049	0.951	1	0	0	0	0
4	true	true	0.049	0.951	1	0	-1	1	1
5	true	true	0.049	0.951	1	0	0	0	0
6	true	true	0.049	0.951	-1	-1	-1	0	0
7	true	true	0.049	0.951	0	-1	0	0	0
8	true	true	0.049	0.951	0	1	0	0	0
9	true	true	0.049	0.951	1	2	0	0	1
10	true	true	0	1	0	1	0	0	0
11	true	true	0.049	0.951	0	0	0	0	0
12	true	true	0.049	0.951	0	0	0	0	0
13	true	false	0.900	0.100	0	0	0	0	0

تصویر ۲۵- دیتاست بعد از اعمال مدل با تقسیم ۸۰-۲۰

۲.۲.۲. پیاده‌سازی 5-fold cross validation

در گام بعدی 5-fold cross validation نیز پیاده سازی می‌کنیم. بعد از اضافه کردن اپراتور cross validation. به sub-process مربوطه رفته و تغییرات زیر را انجام می‌دهیم.

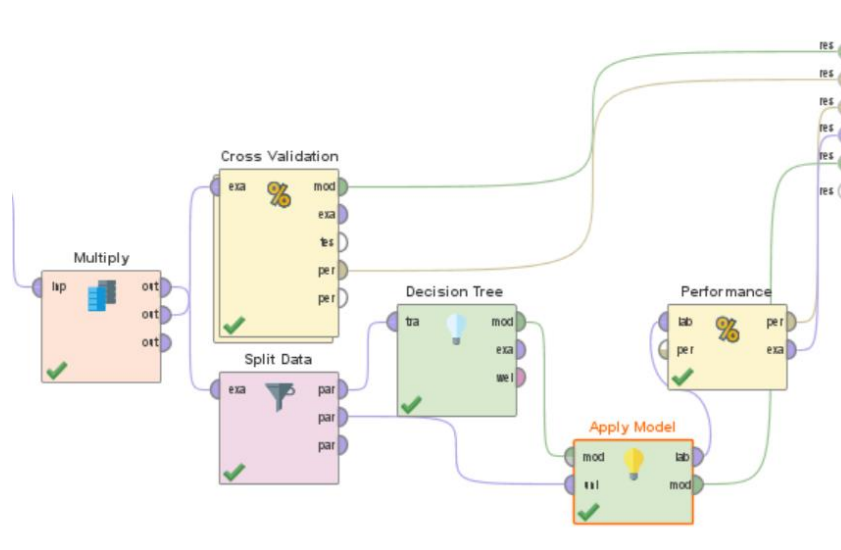


تصویر ۲۶- تغییرات ایجاد شده در sub-process برای 5-fold cross validation

همچنین معیارهای ارزیابی را هم برای روش train-test split برای 5-fold cross validation به صورت زیر انتخاب می‌کنیم:

- Accuracy
- Classification error
- Kappa

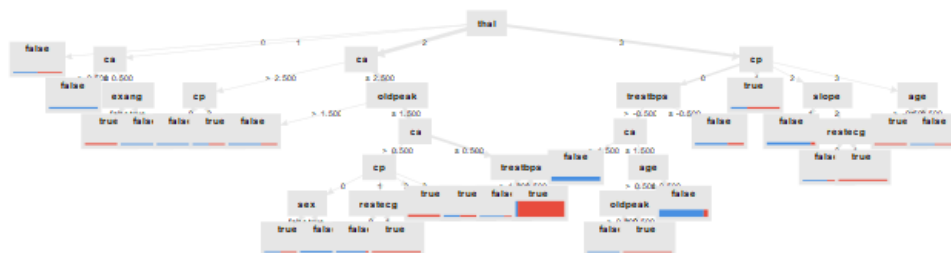
کل پروسه‌های اضافه شده بعد از انتخاب ویژگی‌ها (select attributes) به صورت زیر است:



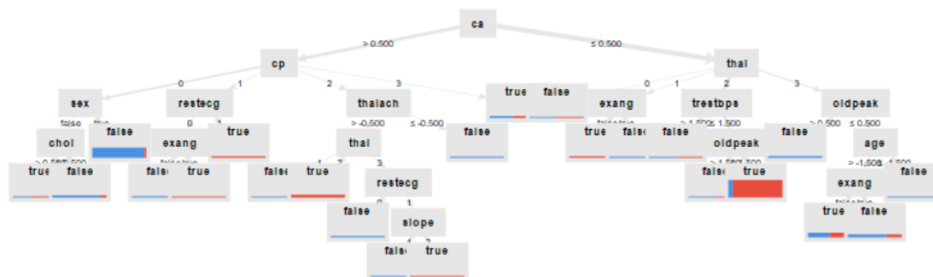
تصویر ۲۷- اپراتورها برای پیاده سازی مدل ها

۲.۳. بررسی نتایج و ارزیابی

حال به بررسی گرافهای حاصله می پردازیم:



تصویر ۲۸- درخت تصمیم حاصل شده از روش تقسیم بندی ۸۰-۲۰



تصویر ۲۹- درخت تصمیم حاصل شده از روش 5-fold cross validation

در گام بعد به مقادیر بدست آمده برای ارزیابی مدل می پردازیم و اول با train/test split شروع می کنیم.

accuracy: 81.03% classification_error: 18.97% kappa: 0.610

	true false	true true	class precision
pred. false	18	3	85.71%
pred. true	8	29	78.38%
class recall	69.23%	90.62%	

تصویر ۳۰- معیارهای ارزیابی مدل train/test split

دقت مدل با توجه به اینکه accuracy بیش از ۸۰٪ است، خوب است و می‌توان حدوداً گفت که مدل قابل قبول است (البته که باید توجه کنیم که ممکن است این امتیاز صرفاً به خاطر سمپلینگ باشد که انجام شده است و باید نتیجه دقیق‌تر را در 5-fold cross validation بررسی کنیم). همچنین با توجه به اینکه معیار کاپا بالاتر از ۶۰٪ در آمده است، قابل قبول است (L., 2012) (هرچند که با حالت ایده آل ۱۰۰٪ فاصله زیادی دارد).

همچنین ۲۹ تا از رکوردهایی که واقعا True بودند، True پیشبینی شده‌اند و ۱۸ تا از آن‌هایی که False بودند، False پیشبینی شده‌اند. در ادامه باید ذکر کرد که ۸ تا از رکوردهایی که واقعا False بودند، اشتباهاً True و ۳ تا از رکوردهایی که واقعا True بودند اشتباهاً False پیشبینی شده‌اند.

معیارهای دیگر مثل Precision, Recall و F1-Score را نیز می‌توانیم حساب کنیم:

$$Precision = \frac{29}{29 + 8} = 78.38\%$$

$$Recall = \frac{29}{29 + 3} = 90.62\%$$

$$F1 - Score = \frac{2 \times 0.7838 \times 0.9062}{0.7838 + 0.9062} = 84.05\%$$

قابل ذکر است که Precision می‌گوید از بین تمام رکوردهایی که مثبت پیشبینی شده‌اند، چه درصدی واقعا درست هستند و Recall می‌گوید از بین تمام رکوردهایی که واقعا مثبت هستند، چه درصدی واقعا درست هستند.

که مقدار خوبی محسوب می‌شود. در ادامه همین کارها را برای 5-fold cross validation انجام می‌دهیم.

accuracy: 73.02% +/- 6.48% (micro average: 73.01%) kappa: 0.451 +/- 0.132 (micro average: 0.452)

	true false	true true	class precision
pred. false	88	38	69.84%
pred. true	40	123	75.46%
class recall	68.75%	76.40%	

تصویر ۳۱- معیارهای ارزیابی مدل 5-fold cross validation

دقت مدل با توجه به اینکه accuracy بیش از ۷۰٪ است، خوب است هرچند می‌بینیم که از حالت قبل کمتر است و تاثیر سمپلینگ که انجام داده ایم این بار کم شده است. همچنین با توجه به اینکه معیار کاپا ۴۵٪ است، مدل از این بابت ضعیف است.

همچنین ۱۲۳ تا از رکوردهایی که واقعا True بودند، True پیشبینی شده اند و ۸۸ تا از آنهایی که False بودند، False پیشبینی شده اند. در ادامه باید ذکر کرد که ۴۰ تا از رکوردهایی که واقعا False بودند، اشتباها True و ۳۸ تا از رکوردهایی که واقعا True بودند اشتباها False پیشبینی شده اند.

معیارهای دیگر مثل Precision, Recall و F1-Score را نیز می توانیم حساب کنیم:

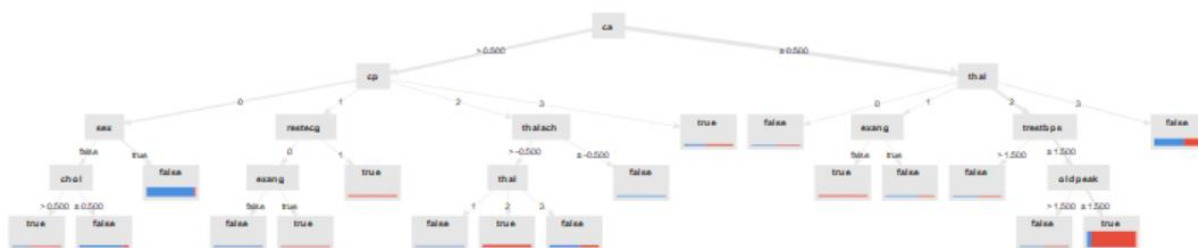
$$Precision = \frac{123}{123 + 40} = 75.46\%$$

$$Recall = \frac{123}{123 + 38} = 76.4\%$$

$$F1 - Score = \frac{2 \times 0.7546 \times 0.764}{0.7546 + 0.764} = 75.92\%$$

که مقدار خوبی محسوب می شود.

صرفا برای بررسی، عمق درخت را از ماکسیمم ۷ به ۵ تغییر میدهم و نتیجه را بررسی میکنیم.



تصویر ۳۲- درخت تصمیم حاصل شده از روش تقسیم بندی 5-fold cross validation بعد از تغییر پارامتر ماکسیمم عمق

accuracy: 73.35% +/- 6.31% (micro average: 73.36%) kappa: 0.452 +/- 0.132 (micro average: 0.457)

	true false	true true	class precision
pred. false	86	35	71.07%
pred. true	42	126	75.00%
class recall	67.19%	78.26%	

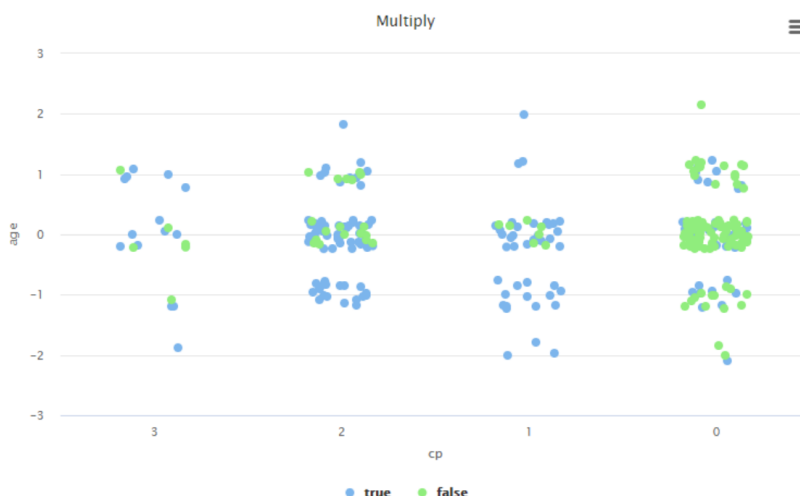
تصویر ۳۳- معیارهای ارزیابی مدل 5-fold cross validation بعد از تغییر پارامتر ماکسیمم عمق

با توجه به اینکه مقدار accuracy و F1-Score افزایش می یابد و همچنین interpretability مدل درخت تصمیم بیشتر می شود، پارامتر maximum depth=5 به ۷، ارجح داده می شود.

$$F1 - Score = \frac{2 \times 0.75 \times 0.7826}{0.75 + 0.7826} = 76.59\%$$

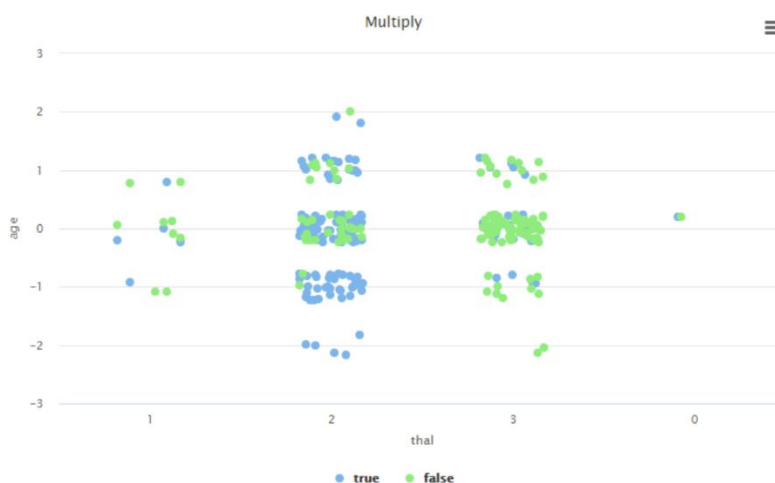
۲.۴. بررسی اسکترپلات ویژگی ها

در نهایت به بررسی برخی اسکترپلات ها می پردازیم.



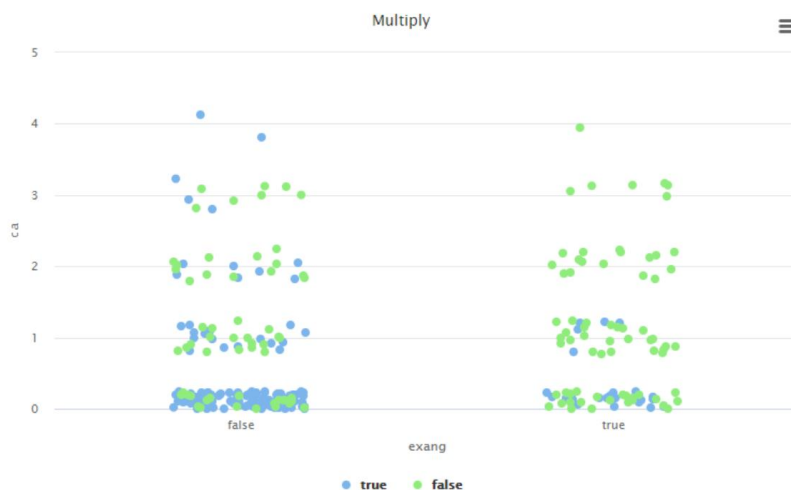
تصویر ۳۴- اسکترپلات سن بر اساس درد قلب به تفکیک بیماری قلبی

در تصویر ۳۴، age بر اساس cp رسم شده است و رنگ‌ها نشان‌دهنده این است که یک نفر آیا بیماری قلبی دارد یا خیر. همانطور که می‌توان مشاهده کرد، اگر یک نفر $cp=0$ باشد که با توجه به متادیتا یعنی typical angina دارد، به احتمال بالا بیماری قلبی ندارد و اگر atypical angina داشته باشد، احتمال بالا بیماری قلبی دارد.



تصویر ۳۵- اسکترپلات سن بر اساس تالاسمی به تفکیک بیماری قلبی

در تصویر ۳۵، age بر اساس thal رسم شده است و رنگ‌ها نشان‌دهنده این است که یک نفر آیا بیماری قلبی دارد یا خیر. همانطور که می‌توان مشاهده کرد، اگر یک نفر $thal=2$ باشد (چون که متادیتا خراب است و ۳ مقدار ۳، ۶ و ۷ برای این فیچر ذکر شده است با وجود اینکه این فیچر مقادیر ۰، ۱، ۲ و ۲ دارد، معنی $thal=2$ نیامده است) احتمال بالا بیماری قلبی دارد و اگر $thal=3$ باشد، احتمال بالا بیماری قلبی ندارد. البته رابطه معناداری بین سن و تالاسمی نمیتوان مشاهده کرد.



تصویر ۳۶- اسکترپلات تعداد رگ‌های اصلی بر اساس آنژین ناشی از ورزش به تفکیک بیماری قلبی

در تصویر ۳۶، ca بر اساس $exang$ رسم شده است و رنگ‌ها نشان‌دهنده این است که یک نفر آیا بیماری قلبی دارد یا خیر. همانطور که می‌توان مشاهده کرد، اگر یک نفر آنژین داشته باشد، صرف نظر از تعداد رگ‌های اصلی، بیماری قلبی نخواهد داشت. اما اگر آنژین نداشته باشد و تعداد رگ‌های اصلی ۰ باشد، احتمال خیلی بالا بیماری قلبی دارد.

۳. منابع

L., M. M. (2012). Interrater reliability: the kappa statistic. *Biochemia medica*, 276–282.

van Buuren, S. &-O. (2011). mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 1–67.