به نام حق

تمرین سوم داده کاوی

دکتر ناظرفرد

حسین محمدی

۹۵۳۳۰۸۱

ترم بهار ۹۸-۹۹

بخش تشریحی

سوال اول

هر دو الگوریتم جنگل تصادفی و درخت گرادیان تقویتی از جمله الگوریتم های طبقه بندی ensemble یا همان ترکیبی میباشند. طبقه بندهای ترکیبی از ترکیب چندین طبقه بند (classifier) استفاده می‌کنند. در واقع این طبقه بندها، هر کدام مدل خود را بر روی داده‌ها ساخته و این مدل را ذخیره می‌کنند. در نهایت برای طبقه بندی نهایی یک رای گیری در بین این طبقه بندها انجام می‌شود و آن طبقه یا کلاسی که بیشترین رای را بیاورد، طبقه یا کلاس نهایی محسوب می‌شود.طبقه بند ترکیبی به جای اینکه خود یک مدل بسازد از مدل‌های ساخته شده توسط بقیه ی طبقه بندها استفاده کرده و با یک رای گیری، مشخص می‌کند که کدام طبقه را برای نمونه ی آزمایشی باید انتخاب کند.

الگوریتم جنگل تصادفی جزو الگوریتم های Bagging یا کیسه گذاری است و الگوریتم گرادیان تقویتی جزو الگوریتم های Boosted یا تقویت شده است که کیسه گذای و تقویت شده از زیرشاخه های الگوریتم ensemble هستند.

جنگل تصادفی:

در میان الگوریتم‌های فعلی از نظر دقت بی‌نظیر است.

روی داده‌های بسیار بزرگ قابل اجراست.

می‌تواند هزاران متغیر را بدون حذف متغیرها مدیریت کند.

برآوردی از مهمترین متغیرها در طبقه‌بندی می‌دهد.

راه کارایی برای برآورد داده‌ها گم‌شده دارد.

جنگل تصادفی، تصادفی بودن افزوده‌ای را ضمن رشد درختان به مدل اضافه می‌کند. این الگوریتم، به جای جست‌وجو به دنبال مهم‌ترین ویژگی‌ها هنگام تقسیم کردن یک «گره(Node) «، به دنبال بهترین ویژگی‌ها در میان مجموعه تصادفی از ویژگی‌ها می‌گردد. این امر منجر به تنوع زیاد و در نهایت مدل بهتر می‌شود. بنابراین، در جنگل تصادفی، تنها یک زیر مجموعه از ویژگی‌ها توسط الگوریتم برای تقسیم یک گره در نظر گرفته می‌شود. با استفاده افزوده از آستانه تصادفی برای هر ویژگی به جای جست‌وجو برای بهترین آستانه ممکن، حتی می‌توان درخت‌ها را تصادفی‌تر نیز کرد (مانند کاری که درخت تصمیم نرمال انجام می‌دهد.

جنگل تصادفی درخت تصمیم‌های زیادی را تولید می‌کند. برای طبقه‌بندی یک شیء جدید از برداری ورودی را در انتهای هر یک از درختان جنگل تصادفی قرار می‌دهد. هر درخت به ما یک طبقه‌بندی می‌هد و می‌گوییم این درخت به

آن کلاس "رای" می‌دهد. جنگل طبقه‌بندی‌ای که بیشترین رای را داشته باشد (بین همه درخت های جنگل) انتحاب می‌کند.

هر درخت به صورت زیر تشکیل می‌شود:

۱. اگر N تعداد حالت‌ها در مجموعه داده‌های) trainمجموعه‌ی کار) باشد، N حالت را به صورت تصادفی با جایگذاری از داده‌های اصلی، نمونه‌گیری می‌کنیم.این نمونه مجموعه‌ی کار برای این درخت می‌باشد.

۲. اگر M متغیر داشته باشیم و m را کوچکتر از M در نظر بگیریم به طوری که در هر گره، m متغیر به صورت تصادفی از M انتخاب می‌شوند و بهترین جداسازی روی این m متغیر برای جداسازی گره استفاده می‌شود. مقدار m در طول ساخت جنگل ثابت در نظر گرفته می‌شود.

۳. هر درخت به اندازه‌ی ممکن بزرگ می‌شود. هیچ هرسی وجود ندارد.

نرخ خطای جنگل به دو مورد زیر بستگی دارد:

همبستگی بین هر دو درخت در جنگل. افزایش همبستگی نرخ خطای جنگل را افزایش می‌دهد.

قدرت هر یک از درختان در جنگل. هر درخت با نرخ خطای کم یک طبقه بند قوی است. افزایش قدرت هر یک از درختان نرخ خطای جنگل را کاهش می‌دهد.

کاهش m هم همبستگی و هم قدرت را کاهش می‌دهد. و افزایشش هر دو را افزایش می‌دهد.

درخت گرادیان تقویتی:

مدل گرادیان تقویتی ، ترکیبی خطی از یک سری مدل‌های ضعیف است که به صورت تناوبی برای ایجاد یک مدل نهایی قوی ساخته شده است.

در این الگوریتم ، هر درخت جدید متناسب با نسخه اصلاح شده از مجموعه داده های اصلی است.

این الگوریتم بسیاری از مدل ها را به صورت تدریجی ، افزودنی و پی در پی آموزش می دهد.

در این الگوریتم ابتدا یک درخت ساخته شده و سپس داده هایی که درخت اول به اشتباه تشخیص داده را شناسایی میکند. سپس احتمال انتخاب مجموعه داده هایی که اشتباه تشخیص داده شده اند در درخت های بعدی بیشتر میشود تا بتوان بهتر درخت بعدی را ساخت و نرخ خطا را پایین اورد. در اینجا برای مثال تابع ارور یا loss function بر اساس توابع لگاریتمیک نوشته میشوند.

اگر در داده ها نویز وجود داشته باشد این الگوریتم نسبت به overfitting شدن بیشتر حساس خواهد بود.

به دلیل این که درختان بطور پی در پی ساخته می شوند ، ساختن مدل بیشتر طول می کشد.

تنظیم آن نسبت به جنگل تصادفی سخت تر است. در واقع اگر پارامترها درست مقدار دهی شوند آنگاه نسبت به جنگل تصادفی بهتر عمل میکند. به طور معمول سه پارامتر وجود دارد: تعداد درختان ، عمق درختان و سرعت یادگیری و هر درخت ساخته شده عموما کم عمق است.

پس بر خلاف جنگل تصادفی که در پایان مدلها را ترکیب میکند ، در درخت گرادیان تقویتی در طول ساخت درختها صورت میپذیرد.

سوال دوم

مشکل این درخت تصمیم آن است که بایاس شده و overfitting رخ داده است.

دو راهکار برای حل این مشکل وجود دارد.

اول آنکه از بررسی همه داده های آموزشی خودداری کنیم تا ابعاد درخت زیاد نشده و برای مجموعه داده های آموزشی از overfitting جلوگیری کنیم.

دوم آنکه بر خلاف راهکار اول همه داده ها بررسی میکنیم اما در پایان از ابعاد درخت میکاهیم و آن را هرس میکنیم.

سوال سوم

معیار accuracy برای فهمیدن عملکرد کلی مناسب است اما در مواقعی به درستی نمیتواند ما را از نتیجه چیزی که داریم مطمئن سازد. برای مثال اگر در داده های آموزشی چند کلاس داشته باشیم و تعداد داده هایی که متعلق به یک کلاس مثلا کلاس آ ، ۹۰ درصد کل داده ها را شامل شود آنگاه به دلیل آنکه داده ها متعادل نیستند و در واقع از همه کلاس ها به یک اندازه داده نداریم. بنابراین داده های بعدی اکثرا کلاس آ تشخیص داده خواهند شد.

بنابراین نیازمند معیارهای دیگری هستیم که آنها را معرفی میکنیم.

حساسیت و تشخیص‌پذیری دو شاخص مهم برای ارزیابی آماری عملکرد نتایج آزمون‌های طبقه‌بندی باینری (دودویی یا دوحالته) هستند، که در آمار به عنوان توابع طبقه‌بندی شناخته می‌شوند. زمانی که بتوان داده‌ها را به دو گروه مثبت و منفی تقسیم کرد، عملکرد نتایج یک آزمایش که اطلاعات را به این دو دسته تقسیم می‌کند با استفاده از شاخص‌های حساسیت و تشخیص‌پذیری قابل اندازه‌گیری و توصیف است .

پارامتر تشخیص‌پذیری را نیز اصطلاحا دقت(Precision) ، و حساسیت را نیز اصطلاحا صحت (Recall) می‌نامند.

دسته بندی داده ها

۱ -مثبت صحیح(True Positive)

۲ -مثبت کاذب(False Positive)

۳ -منفی صحیح(True Negative)

۴ -منفی کاذب(False Negative)

مثال های از "مثبت های کاذب" و "منفی های کاذب"

- **بخش امنیت فرودگاه :**یک "مثبت کاذب" هنگامی است که اشیاء معمولی مانند کلیدها و یا سکه ها به اشتباه اسلحه تشخیص داده می شوند)و ماشین صدای "بیپ" را ایجاد می کند(
- **کنترل کیفیت :**یک "مثبت کاذب" هنگامی است که محصول با کیفیت خوب، مردود می شود و یک "منفی کاذب" هنگامی است که محصول بی کیفیت مورد قبول واقع می شود
- **نرم افزار ضد ویروس :**یک "مثبت کاذب" هنگامی است که یک فایل عادی بعنوان یک ویروس شناخته می شود
- **آزمایش پزشکی :**گرفتن آزمایش های ارزان قیمت و بررسی آن توسط تعداد زیادی از پزشکان می تواند مثبتهای کاذب زیادی به بار آورد)یعنی جواب تست بگوید که بیمار هستید در حالی که چنین نیست)، و سپس خواسته شود که دوباره آزمایش های با دقت بیشتر بگیرید.

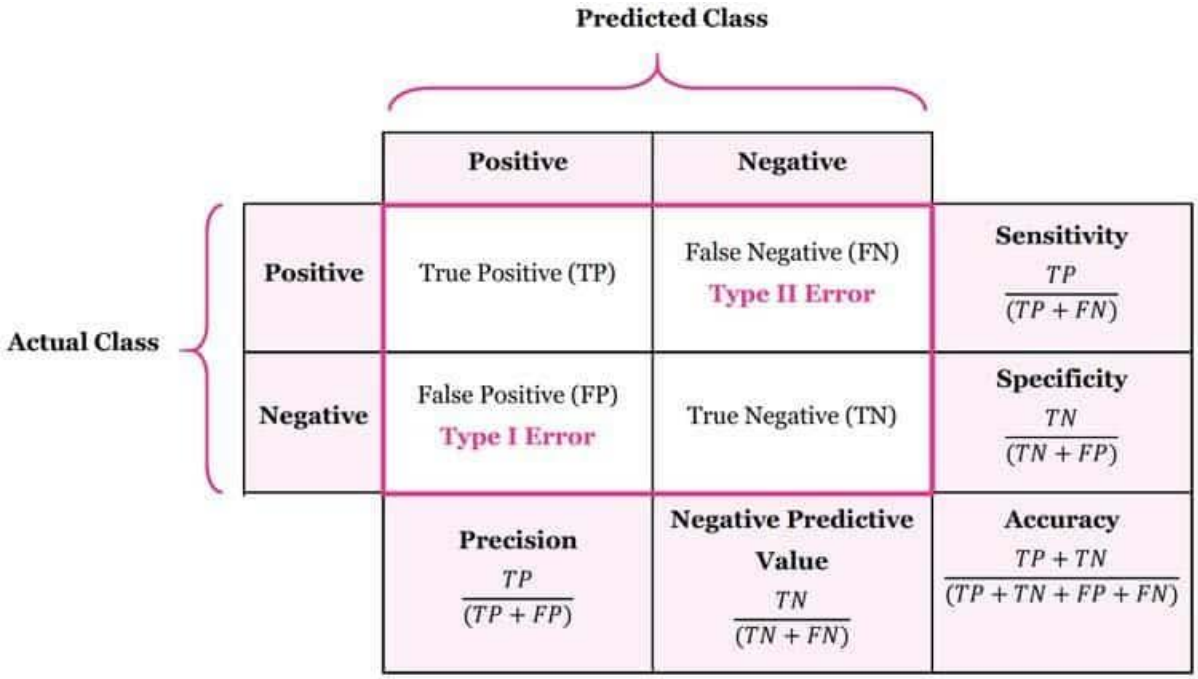
کدام معیارها باید بهتر باشند:

۱ -مثبت صحیح = (True Positive) درست شناسایی شده است.

۲ -مثبت کاذب = (False Positive) اشتباه شناسایی شده است (خطای نوع یک در انجام آزمون).

۳ -منفی صحیح = (True Negative) به درستی رد شد.

۴ -منفی کاذب = (False Negative) اشتباه رد شد (خطای نوع دوم در انجام آزمون).



روش های ارزیابی دقت و صحت

درنهایت نتایج به‌دست‌آمده مورد ارزیابی قرارگرفته و برای موارد مختلف تفسیر و استفاده می‌شود. در ارزیابی معمولا معیارهای زیر متصور است.

۱. **تشکیل ماتریس اختلاط(confusion matrix)**

۲. **دقت(Accuracy)**

۳. **صحت(Precision)**

۴. **Recall:** زمانی که ارزش false negatives بالا باشد، معیارRecall ، معیار مناسبی خواهد بود.

۵. **F1 Score**

۶. **MCC:** پارامتر دیگری است که برای ارزیابی کارایی الگوریتم‌های یادگیری ماشین از آن استفاده می‌شود. این پارامتر بیان‌گر کیفیت کلاس‌بندی برای یک مجموعه باینری می‌باشد.

دقت:(Accuracy)

به طور کلی، دقت به این معناست که مدل تا چه اندازه خروجی را درست پیش‌بینی می‌کند. با نگاه کردن به دقت ، بلافاصله می‌توان دریافت که آیا مدل درست آموزش دیده است یا خیر و کارآیی آن به طور کلی چگونه است. اما این معیار اطلاعات جزئی در مورد کارآیی مدل ارائه نمی‌دهد.

Accuracy = (TP+TN) / (TP+FN+FP+TN)

صحت: (Precision)

وقتی که مدل نتیجه را مثبت (positive) پیش‌بینی می‌کند، این نتیجه تا چه اندازه درست است؟ زمانی که ارزش false positivesبالا باشد، معیار صحت، معیار مناسبی خواهد بود. فرض کنید، مدلی برای تشخیص سرطان داشته باشیم و این مدل Precision پایینی داشته باشد. نتیجه این امر این است که این مدل، بیماری بسیاری از افراد را به اشتباه سرطان تشخیص می‌دهد. نتیجه این امر استرس زیاد، آزمایش‌های فراوان و هزینه‌های گزافی را برای بیمار به دنبال خواهد داشت.

در واقع، «حساسیت» معیاری است که مشخص می‌کند دسته‌بند، به چه اندازه در تشخیص تمام افراد مبتلا به بیماری موفق بوده‌است. همانگونه که از رابطه فوق مشخص است، تعداد افراد سالمی که توسط دسته‌بند به اشتباه به عنوان فرد بیمار تشخیص داده شده‌اند، هیچ تاثیری در محاسبه این پارامتر ندارد و در واقع زمانی که پژوشهگر از این پارامتر به عنوان پارامتر ارزیابی برای دسته‌بند خود استفاده می‌کند، هدفش دستیابی به نهایت دقت در تشخیص نمونه‌های کلاس مثبت است.

در واقع نسبت مقداری موارد صحیح طبقه‌بندی‌شده توسط الگوریتم از یک کلاس مشخص، به کل تعداد مواردی که الگوریتم چه به‌صورت صحیح و چه به‌صورت غلط، در آن کلاس طبقه‌بندی کرده است که به‌صورت زیر محاسبه می‌شود:

Precision =TP / (TP+FP)

فراخوانی یا حساسیت یا: Recall

در نقطه مقابل این پارامتر، ممکن است در مواقعی دقت تشخیص کلاس منفی حائز اهمیت باشد. از متداول‌ترین پارامترها که معمولا در کنار حساسیت بررسی می‌شود، پارامتر خاصیت(Specificity) ، است که به آن «نرخ پاسخ‌های منفی درست (True Negative Rate) «نیز می‌گویند. خاصیت به معنی نسبتی از موارد منفی است که آزمایش آن‌ها را به درستی به عنوان نمونه منفی تشخیص داده است. این پارامتر به صورت زیر محاسبه می‌شود.

زمانی که ارزش false negatives بالا باشد، معیارRecall ، معیار مناسبی خواهد بود. فرض کنیم مدلی برای تشخیص بیماری کشنده ابولا داشته باشیم. اگر این مدل Recall پایینی داشته باشد چه اتفاق خواهد افتاد؟ این مدل افراد زیادی که آلوده به این بیماری کشنده هستند را سالم در نظر می‌گیرد و این فاجعه است. نسبت مقداری موارد صحیح طبقه‌بندی شده توسط الگوریتم از یک کلاس به تعداد موارد حاضر در کلاس مذکور که به‌صورت زیر محاسبه می‌شود:

Recall = Sensitivity = (TPR) = TP / (TP+FN)

معیارهای ارزیابی F1 Score یاF-measure

معیارF1 ، یک معیار مناسب برای ارزیابی دقت یک آزمایش است. این معیار Precisionو Recall را با هم در نظر می‌گیرد. معیار F1 در بهترین حالت، یک و در بدترین حالت صفر است.

F-measure= 2 * (Recall * Precision) / (Recall + Precision)

معیار:(Specificity)

در نقطه مقابل این پارامتر، ممکن است در مواقعی دقت تشخیص کلاس منفی حائز اهمیت باشد. از متداول‌ترین پارامترها که معمولا در کنار حساسیت بررسی می‌شود، پارامتر خاصیت(Specificity) ، است که به آن «نرخ پاسخ‌های منفی درست (True Negative Rate) «نیز می‌گویند. خاصیت به معنی نسبتی از موارد منفی است که آزمایش آن‌ها را به درستی به عنوان نمونه منفی تشخیص داده است. این پارامتر به صورت زیر محاسبه می‌شود.

Specificity (TNR) = TN / (TN+FP)

معیارهای ارزیابی:MCC

پارامتر دیگری است که برای ارزیابی کارایی الگوریتم‌های یادگیری ماشین از آن استفاده می‌شود. این پارامتر بیان‌گر کیفیت کلاس‌بندی برای یک مجموعه باینری می‌باشدMCC (Matthews correlation coefficient) .، سنجه‌ای است که بیان‌گر بستگی مابین مقادیر مشاهده شده از کلاس باینری و مقادیر پیش‌بینی شده از آن می‌باشد. مقادیر مورد انتظار برای این کمیت در بازه ۱- و ۱ متغیر می‌باشد. مقدار ۱+، نشان دهنده پیش‌بینی دقیق و بدون خطای الگوریتم یادگیر از کلاس باینری می‌باشد. مقدار ۰ ، نشان دهنده پیش‌بینی تصادفی الگوریتم یادگیر از کلاس باینری می‌باشد. مقدار ۱-، نشان دهنده عدم تطابق کامل مابین موارد پیش‌بینی شده از کلاس باینری و موارد مشاهده شده از آن می‌باشد .

سوال چهارم

از آنجا که طبقه بند ها از یکدیگر مستقل هستند و جواب هریک روی جواب دیگری تاثیری ندارد پس میتوانیم از قانون احتمالات مستقل استفاده کنیم.

برای آن که الگوریتم تشخیص اشتباه داشته باشد بیشتر از نصف ۲۵ طبقه بند یعنی ۱۳ طبقه بند جواب اشتباه بدهند.

حال احتمال این که جواب اشتباه دهد شامل تمام حالاتی است که بیشتر از ۱۳ طبقه بند اشتباه تشخیص داده باشند. بنابراین داریم:

$$\sum_{i=13}^{25} \binom{25}{i} (0.35)^i (1-0.35)^{25-i}$$

سوال پنجم

برای ساخت درخت تصمیم طبق فرمولهای زیر پیش میرویم.

$$IG(Y,X) = E(Y) - E(Y|X)$$

Positive class = Edible , Negative class = Poisonous

Info(D) = I(9,5) = - 9/14 * log(9/14) - 5/14 * log(5/14) = 0.94

Habitat.Info(D) = 5/14 * (-3/5 * log(3/5) - 2/5 * log(2/5) + 4/14 * (-4/4 *log(4/4)) + 5/14 * (-3/5 * log(3/5) - 2/5 * log(2/5)) = 0.694

Gain(Habitat) = Info(D) - Habitat.Info(D) = 0.246

CapColor.Info(D) = 4/14 * (-2/4 * log(2/4) - 2/4 * log(2/4) + 6/14 * (-4/6 *log(4/6) - 2/6 *log(2/6)) + 4/14 * (-3/4 * log(3/4) - 1/4 * log(1/4)) = 0.91

Gain(Cap Color) = Info(D) - CapColor.Info(D) = 0.03

CapShape.Info(D) = 0.787

Gain(Cap Shape) = Info(D) - CapShape.Info(D) = 0.152

Order.Info(D) = 0.892

Gain(Odor) = Info(D) - Odor.Info(D) = 0.048

1. Habitat 2. Cap Shape 3. Odor 4. Cap Color

سوال ششم

$$P(y|a\,.\,b.\,c) = \frac{P(a|y) * P \quad (b|y) * P(c|y) * P(y)}{\sum_{y'} P(a|y') * P(b|y') * P(c|y') * P(y')}$$

$P(y|a', b, c') = 1/18$

$P(y'|a', b, c') = 4/25$

پس bad نتیجه گیری میشود.

بخش پیاده سازی

پیاده سازی اول

در این بخش برای آنکه بتوانیم از کتابخانه sklearn برای ساخت درخت تصمیم استفاده کنیم نیازمند آن هستیم تا تغییراتی روی داده های خامی که در اختیار داریم انجام دهیم.

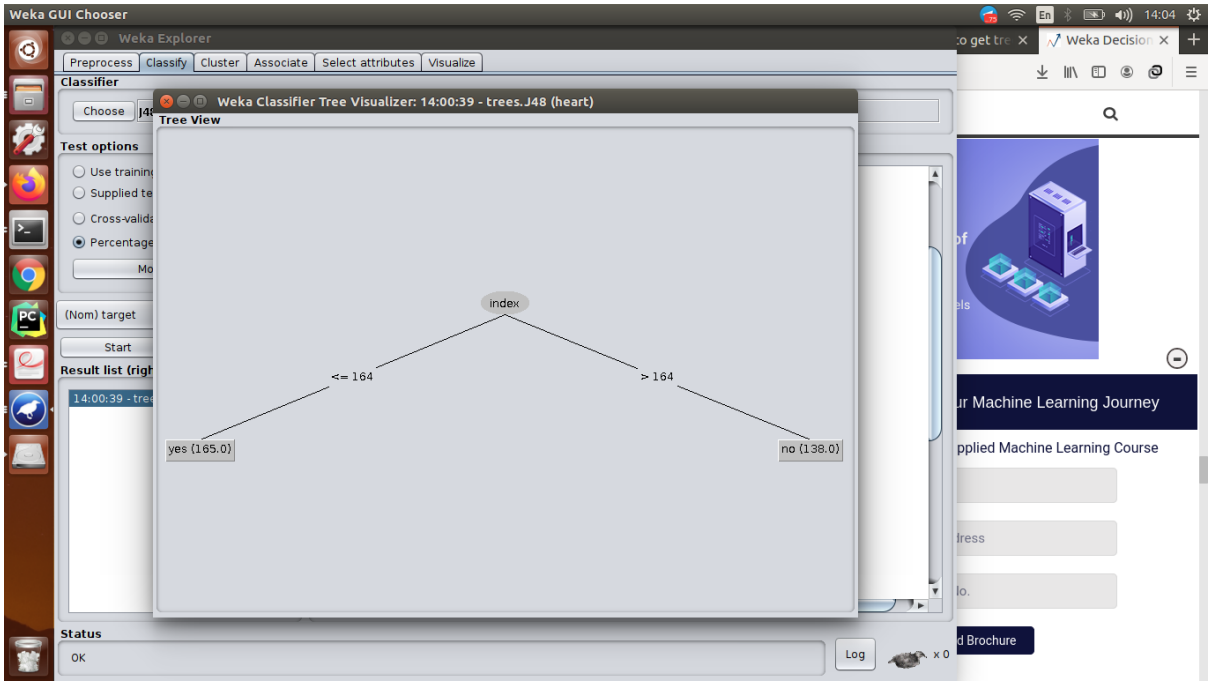
این تغییرات عبارتند از تبدیل اعداد ستون سن به دسته های جدا از هم ، و تبدیل داده های رشته و بولین به تایپ های مناسب برای استفاده از کتابخانه.
برای آموزش و آزمایش درخت از ۸۰ درصد داده ها برای آموزش و ۲۰ درصد داده ها برای آزمایش استفاده میکنیم.

نتایج و درخت های ساخته شده برای هر درخت با مقادیر مختلف در پارامترهای ساخت درخت در فایل jupyter موجود هستند.

پیاده سازی دوم

برای استفاده از weka از ابزار خود این نرم افزار برای تبدیل به فرمت ARFF استفاده میکنیم. مشاهده میکنیم هنگام استفاده از این فایل جدید با ارور خطا در مغایرت تعداد ویژگی ها در اول فایل و تعداد ویژگی ها در برای هر داده میشویم.
که علت آن یک ستون اضافه است که ایندکس داده هاست اما نام ستون در اول فایل ذکر نشده.
پس از رفع ارور و ساخت درخت میبینیم که با دقت بسیار خوبی همه چیز جلو میرود!

میبینیم که درخت به شکل زیر است.

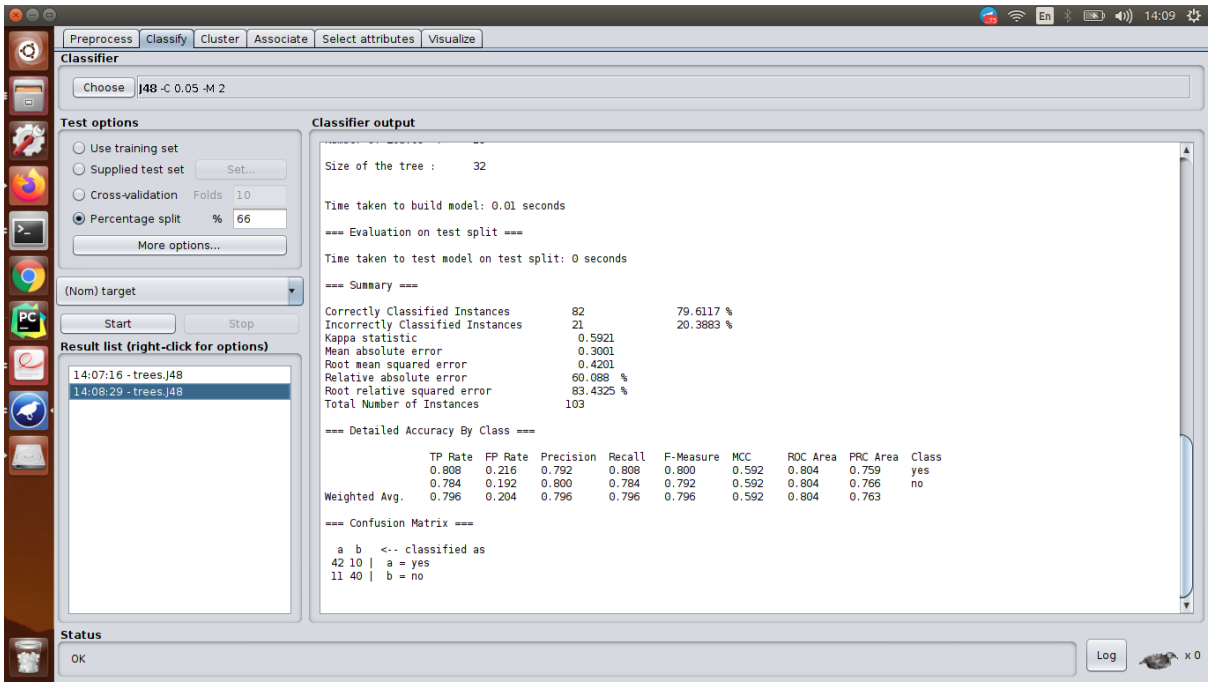


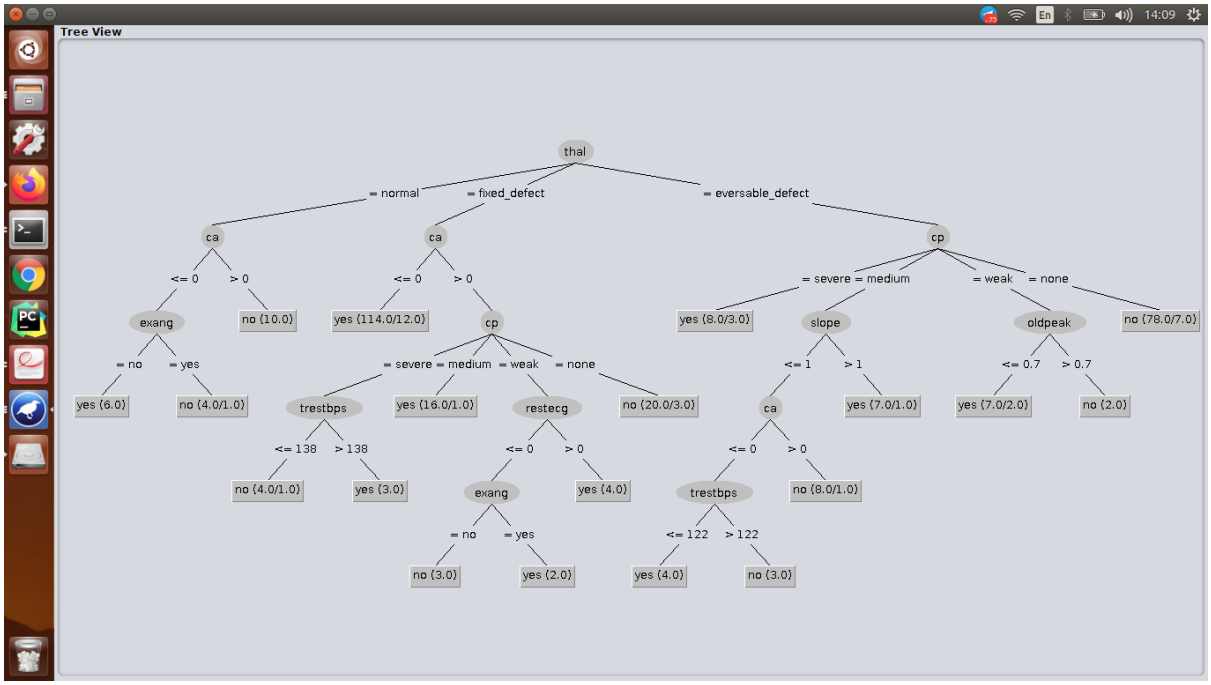
در واقع درخت به شدت بایاس شده است و چون داده های با کلاس yes پشت هم و داده های با کلاس no پشت هم هستند با دقت ۱۰۰ درصدی عمل میکند.

برای رفع این مشکل ستون index را به کلی از داده ها حذف میکنیم.چون این ستون gain بسیار بالایی دارد و طبیعی است که این ویژگی توسط نرم افزار انتخاب شود.

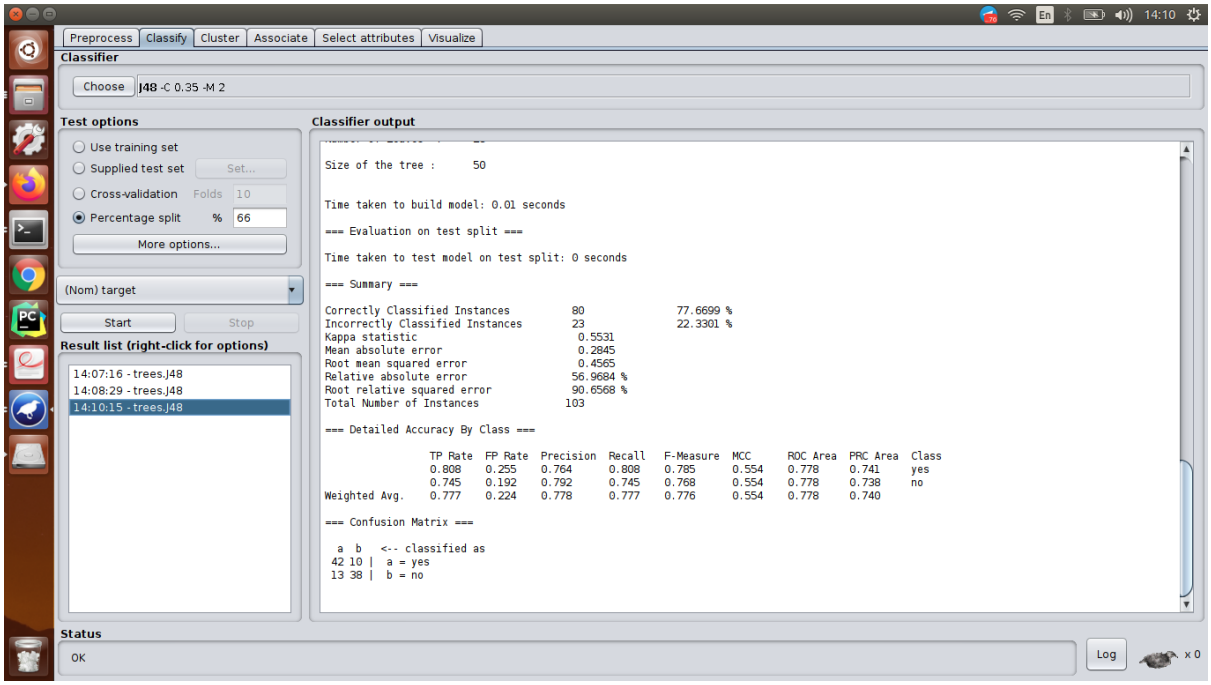
پس از حذف ستون index درخت ها و نتایج با پارامترهای درخواستی در صورت سوال به صورت زیر خواهند بود.

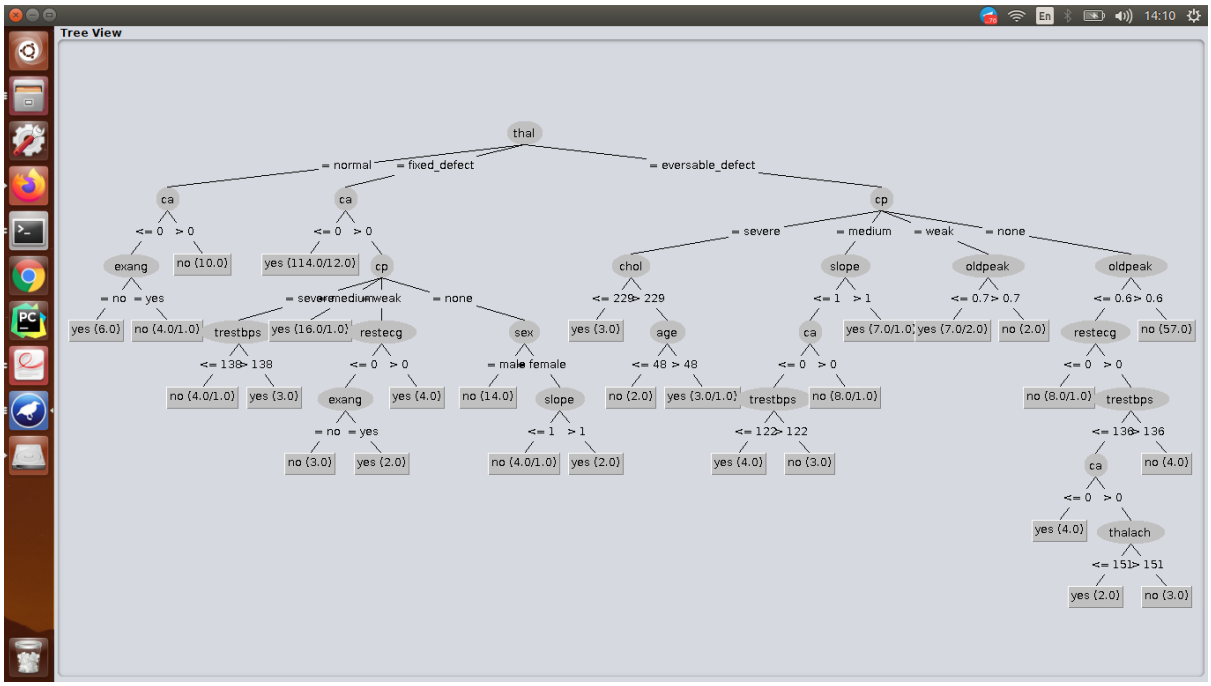
۰/۰۵





۰/۳۵


Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.35 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) target
Start
Stop
Result list (right-click for options)
14:07:16 - trees.J48
14:08:29 - trees.J48
14:10:15 - trees.J48
Classifier output
Size of the tree : 50
Time taken to build model: 0.01 seconds
=== Evaluation on test split ===
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 80 77.6699 %
Incorrectly Classified Instances 23 22.3301 %
Kappa statistic 0.5531
Mean absolute error 0.2845
Root mean squared error 0.4565
Relative absolute error 56.9684 %
Root relative squared error 90.6568 %
Total Number of Instances 103
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.808 0.255 0.764 0.808 0.785 0.554 0.778 0.741 yes
0.745 0.192 0.792 0.745 0.768 0.554 0.778 0.738 no
Weighted Avg. 0.777 0.224 0.778 0.777 0.776 0.554 0.778 0.740
=== Confusion Matrix ===
a b <-- classified as
42 10 | a = yes
13 38 | b = no
Status
OK
Log
x 0


Tree View
thal
= normal
= fixed_defect
= reversable_defect
ca
ca
cp
<= 0
> 0
exang
no (10.0)
yes (114.0/12.0)
cp
= severe
= medium
= weak
= none
chol
slope
oldpeak
oldpeak
= no
= yes
yes (6.0)
no (4.0/1.0)
trestbps
yes (16.0/1.0)
restecg
sex
yes (3.0)
age
<= 229
> 229
<= 1
> 1
<= 0.7
> 0.7
<= 0.6
> 0.6
ca
yes (7.0/1.0)
yes (7.0/2.0)
no (2.0)
restecg
no (57.0)
<= 138
> 138
no (4.0/1.0)
yes (3.0)
<= 0
> 0
exang
yes (4.0)
= male
= female
no (14.0)
slope
<= 48
> 48
no (2.0)
yes (3.0/1.0)
<= 0
> 0
trestbps
no (8.0/1.0)
no (8.0/1.0)
trestbps
<= 0
> 0
no (8.0/1.0)
trestbps
= no
= yes
no (3.0)
yes (2.0)
<= 1
> 1
no (4.0/1.0)
yes (2.0)
<= 122
> 122
yes (4.0)
no (3.0)
<= 136
> 136
ca
no (4.0)
<= 0
> 0
yes (4.0)
thalach
<= 151
> 151
yes (2.0)
no (3.0)

پیاده سازی سوم

در این بخش نیز از دسته بند بیز ساده برای تشخیص کلاس متن استفاده میکنیم.

از آنجا که این کار قبلا توسط بنده انجام شده بود تنها اصلاحاتی جزئی برای کار با فایل دیتای موجود و شرایط مسئله انجام شد که گزارش دسته بند و نحوه عملکرد آن در فایلی جدا در پوشه گزارش موجود است و نتایج آن در فایل jupyter نیز موجود است.گفتنی است smoothing نیز انجام شده است.