

عنوان

*پیش بینی سرطان ریه به کمک روش های مورد استفاده در هوش محاسباتی*

Title

*Lung cancer Prediction using paradigms of*

*Computational Intelligence*

تاریخ تحویل : 17/11/1402

نام درس : هوش محاسباتی

تهیه کنندگان : سینا جوادی و آرمین غفارپور فلاحتی

استاد درس : پروفسور اکبرزاده توتونچی

چکیده و شیوه نامه انجام تحقیق :

سرطان ریه شامل علائم و نشانه هایی است که به کمک آنها میتوان الگوریتم پیش بینی و یادگیری خود را تدوین کرد ؛ لذا ابتدا با استفاده از مقالات و سایت های مختلف ، عوامل و نشانه های بروز سرطان ریه را مطالعه کردیم. این نشانه ها شامل سرفه مداوم یا تغییر در سرفه معمولی ، خون ریزی در ریه ، درد در قفسه سینه ، تنگی نفس ، تب و عفونت های مکرر میباشد. عواملی که میتوانند خطر ابتلا به سرطان ریه را افزایش دهد شامل مصرف دخانیات ، آلودگی هوا و تابش یونیزان و ... میبباشند . منابعی که در این موضوع به ما کمک کرده است در قسمت مراجع آورده شده است.

پس از دانستن این نشانه ها ، ویژگی هایی که به نظر به طور قابل توجهی بر سرطان ریه تاثیر میگذارد را از Dataset در دست داشته انتخاب کردیم.

ما در این تحقیق از الگوریتم هایDecision Trees و Random Forest برای modelling روش خود استفاده کردیم. همچنین از الگوریتم ژنتیک برای بهینه سازی در روش خود بهره بردیم . منابعی که به ما در زمینه یDecision Trees و Random Forest کمک کردند شامل موارد ذیل میباشد :

* Decision trees: from efficient prediction to responsible AI : Hendrik Blockeel & Laurens Devos
* https://betterdatascience.com/mml-random-forest

ابتدا با استفاده از Decision Trees ، مدل سازی و یادگیری داده های موجود در دیتاست را انجام داده و خطای این روش را بدست آوردیم. پس از این کار به سراغ روش Random Forest رفته و با accuracy بیشتر مسئله سرطان ریه را تحلیل کردیم. پس از آنکه خطای کمتر روش Random Forest بر ما آشکار شد ، با کمک الگوریتم ژنتیک برخی از پارامترهای مورد نیاز خود در این روش را بهینه کردیم.

در واقع چالش ما توجه به عوامل و نشانه های سرطان ریه ی افراد مختلفِ dataset و یادگیری این داده ها و سپس استفاده از مدل یادگیری خود برای داده ها و بدست آوردن خطای روش خود میباشد.

یکی از مشکلات ما در روش Random Forest بیش برازش یا overfitting است که ما با استفاده از الگوریتم ژنتیک تا حدی سعی کردیم این مشکل را به حداقل برسانیم.

مقدمه :

سرطان ریه یک بیماری است که در آن سلول‌های سرطانی در بافت ریه شروع به رشد بی‌رویه می‌کنند و توانایی تنفس فرد را کاهش می‌دهند. این بیماری علت اصلی مرگ و میر ناشی از سرطان در زنان و مردان است و درمان قطعی ندارد. سرطان ریه به دو نوع اصلی تقسیم می‌شود: سرطان ریه سلول کوچک و سرطان ریه سلول غیر کوچک. سرطان ریه سلول غیر کوچک شایع‌ترین نوع سرطان ریه است و خود به چند زیرنوع مانند آدنوکارسینوم، سرطان سلول سنگفرشی و کارسینوم سلول بزرگ تقسیم می‌شود. سرطان ریه سلول کوچک نوعی سرطان سریع‌الانتشار و صعب‌العلاج است که معمولا در افراد سیگاری دیده می‌شود. علت اصلی سرطان ریه مصرف دخانیات است، اما عوامل دیگری مانند قرار گرفتن در معرض دود سیگار، آلودگی هوا، تابش یونیزان، آسبست و برخی عوامل ژنتیکی نیز می‌توانند خطر ابتلا به این بیماری را افزایش دهند. علائم سرطان ریه ممکن است در مراحل اولیه ظاهر نشوند یا با علائم بیماری‌های ریوی دیگر اشتباه گرفته شوند. برخی از علائم شایع سرطان ریه عبارتند از: سرفه مداوم یا تغییر در سرفه معمول، خون‌ریزی از ریه، درد قفسه سینه، تنگی نفس، خستگی، از دست دادن وزن، تب و عفونت‌های مکرر. تشخیص سرطان ریه با استفاده از روش‌های مختلفی مانند آزمایش‌های تصویربرداری، آزمایش‌های آزمایشگاهی، بیوپسی و غربالگری انجام می‌شود. مرحله‌بندی سرطان ریه بر اساس اندازه و محل تومور و گسترش آن به بافت‌های اطراف و نواحی دوردست تعیین می‌شود. درمان سرطان ریه بستگی به نوع، مرحله و وضعیت سلامتی بیمار دارد. روش‌های مختلف درمان سرطان ریه شامل جراحی، شیمی درمانی، پرتودرمانی، درمان‌های هدفمند و ایمونوتراپی هستند. طول عمر بیماران سرطان ریه به عوامل مختلفی مانند نوع، مرحله، درمان و واکنش بدن بیمار بستگی دارد. به طور کلی، بیماران سرطان ریه سلول کوچک شانس کمتری برای زنده ماندن دارند.

Decision Trees یا درخت‌های تصمیم، مدل‌های یادگیری ماشین هستند که برای حل مسائل طبقه‌بندی و رگرسیون استفاده می‌شوند. این مدل‌ها به صورت ساختارهای شاخه‌دار نمایش داده می‌شوند که در آن هر گره داخلی یک سوال منطقی و هر گره برگ یک پاسخ است. این مدل‌ها با پیمایش از ریشه تا برگ، می‌توانند پیش‌بینی‌هایی بر اساس داده‌های ورودی ارائه دهند. این مدل‌ها دارای مزایایی مانند سادگی، قابل فهم بودن، تطبیق‌پذیری و کارایی هستند. اما همچنین دارای معایبی مانند بی‌ثباتی، بیش‌برازش، حساسیت به نویز و محدودیت در پردازش داده‌های پیوسته هستند.

چندین الگوریتم برای ساخت درخت تصمیم وجود دارد که از معیارهای مختلفی برای انتخاب بهترین ویژگی در هر گره استفاده می کنند. برخی از این الگوریتم ها عبارتند از ID3, C4.5, CHAID, CART و …

Random forest یا جنگل تصادفی یک روش یادگیری ترکیبی است که برای دسته بندی، رگرسیون و سایر وظایف از ترکیب چندین درخت تصمیم استفاده می کند. [این روش با ایجاد تنوع و کاهش وابستگی بین درخت ها، دقت و پایداری بیشتری نسبت به درخت تصمیم تنها دارد](https://en.wikipedia.org/wiki/Random_forest). برای ساخت یک جنگل تصادفی، ابتدا چندین نمونه زیرمجموعه از داده های آموزشی با روش جایگزینی تصادفی انتخاب می شوند. سپس برای هر نمونه، یک درخت تصمیم با روش CART ساخته می شود. با این تفاوت که در هر گره، تنها یک زیرمجموعه تصادفی از ویژگی ها برای انتخاب بهترین تقسیم بندی در نظر گرفته می شود. این کار باعث می شود که درخت ها از هم مستقل و متفاوت باشند.  [در نهایت، برای پیش بینی یک نمونه جدید، همه درخت ها رای می دهند و رای بیشترین کلاس یا میانگین پیش بینی ها به عنوان خروجی جنگل تصادفی انتخاب می شود](https://en.wikipedia.org/wiki/Random_forest). جنگل تصادفی یک روش ساده، انعطاف پذیر و قدرتمند است که در بسیاری از مسائل یادگیری ماشین کاربرد دارد. این روش مزایایی مانند کاهش بیش برازش، اهمیت ویژگی ها، تعامل ویژگی ها، تشخیص ناهنجاری و … دارد.

الگوریتم ژنتیک یک روش جستجوی تصادفی است که بر اساس فرایندهای تکاملی در طبیعت کار می‌کند. این روش، با استفاده از مفاهیمی مانند وراثت، جهش و انتخاب طبیعی، سعی می‌کند بهترین راه‌حل را برای یک مسئله بهینه‌سازی یا یادگیری ماشین پیدا کند. الگوریتم ژنتیک، جواب‌های ممکن را به صورت کروموزوم‌هایی کد می‌کند که حامل ژن‌هایی هستند که خصوصیات راه‌حل را مشخص می‌کنند. سپس، با اعمال عملگرهایی مانند ترکیب، جهش و انتخاب، جمعیتی از کروموزوم‌ها را در نسل‌های مختلف تولید می‌کند که به مرور زمان به سمت راه‌حل بهینه همگرا می‌شوند. الگوریتم ژنتیک، برای حل مسائل گسسته و غیر خطی بسیار کارآمد است.

توصیف تحقیق :

برای پیاده سازی یک درخت تصمیم برای پیش بینی سرطان ریه ، ابتدا ما باید یک مجموعه داده از بیماران مبتلا به سرطان ریه داشته باشیم که شامل ویژگی های مربوط به بیماری و برچسب های نشان دهنده وضعیت بیمار بودن یا نبودن آنها باشد.

سپس ما باید داده های خود را پیش پردازش کنیم. این شامل تمیز کردن داده ها از نویز و خطا، حذف یا جایگزینی داده های گم شده، تبدیل داده های ریزی به عددی، انتخاب ویژگی های مهم و کاهش بعد داده ها می شود. بعد باید یک الگوریتم برای ساخت درخت تصمیم انتخاب کنیم. الگوریتم های مختلفی برای این کار وجود دارند که از معیارهای مختلفی برای انتخاب بهترین ویژگی در هر گره استفاده می کنند. برخی از این الگوریتم ها عبارتند از ID3, C4.5, CHAID, CART و …

هر کدام از این الگوریتم ها مزایا و معایب خود را دارند که باید با توجه به ماهیت داده ها و هدف پژوهش در نظر گرفته شوند. سپس باید یک مدل درخت تصمیم با استفاده از الگوریتم انتخاب شده بر روی داده های آموزشی بسازیم. برای این کار، باید داده های خود را به دو بخش آموزشی و آزمون تقسیم کنیم و الگوریتم را بر روی بخش آموزشی اجرا کنیم. الگوریتم شروع به تقسیم داده ها از ریشه به سمت برگ ها می کند و در هر گره یک تست انجام می دهد تا بهترین ویژگی را برای تقسیم بندی نمونه ها پیدا کند. این کار تا زمانی ادامه می یابد که همه نمونه ها در یک گره همگن باشند یا شرایط توقفی مانند حداکثر عمق یا حداقل تعداد نمونه در یک گره برقرار شوند. پس از آن باید مدل خود را ارزیابی کنیم. برای این کار، باید مدل را بر روی داده های آزمون که قبلا استفاده نشده اند، اعمال کنیم و معیارهایی مانند دقت، صحت و حساسیت و … را محاسبه کنید. همچنین می توانیم از روش های هرس کردن برای کاهش پیچیدگی و جلوگیری از بیش برازش استفاده کنیم. هرس کردن یک فرآیند است که شاخه هایی را که بر روی ویژگی هایی با اهمیت کم تقسیم می شوند حذف می کند.

در آخر باید مدل خود را به کار ببریم. برای این کار، باید ویژگی های یک نمونه جدید را به مدل وارد کنید و مدل بر اساس تست هایی که در هر گره انجام می دهد، نمونه را به یکی از برچسب های موجود در برگ ها رده بندی می کند. برچسب نهایی نشان دهنده احتمال بیمار بودن یا نبودن است.

اما برای بکارگیری یک جنگل تصادفی برای پیش بینی سرطان ریه، پس از پیش پردازش داده های دردست داشته باید تعدادی درخت تصمیم را با استفاده از روش bagging یا Bootstrapping از داده‌ها بسازیم. این روش، با انتخاب تصادفی یک زیرمجموعه از داده‌ها با حفظ اندازه اصلی، یک نمونه‌برداری با جایگزینی از داده‌ها انجام می‌دهد. سپس برای هر زیرمجموعه، یک درخت تصمیم را با استفاده از یک الگوریتم مانند ID3، C4.5، CART و ... بسازیم. در این مرحله، باید تعداد درخت‌ها و تعداد ویژگی‌هایی که در هر گره بررسی می‌شوند را مشخص کنیم. معمولاً تعداد درخت‌ها بین ۱۰۰ تا ۵۰۰ و تعداد ویژگی‌ها برابر با ریشه‌ی تعداد کل ویژگی‌ها انتخاب می‌شوند. در نهایت، باید جنگل تصادفی ساخته شده را ارزیابی و بهینه‌سازی کنیم. برای این کار، می‌توانیم از معیارهایی مانند دقت، صحت، فراخوانی، معیار F1، منحنی ROC و ... استفاده کنیم.

پیاده سازی و نتایج :

برای پیاده سازی الگوریتم Random Forest و سپس ترکیب آن با GA برای بهینه سازی از زبان برنامه نویسی پایتون استفاده شده است که به توضیح آن میپردازیم:

در پیاده سازی پایتون ابتدا Feature های مدنظر را از دیتاست استخراج میکنیم که برای این کار از کتابخانه Pandas استفاده شده است، سپس داده ها را پیش پردازش میکنیم و داده ها را با مقادیر باینری برای سادگی بیشتر جایگزین میکنیم.

در مرحله بعد برای پیاده سازی Random Forest ابتدا نیاز است الگوریتم Decision Tree را پیاده سازی کنیم. برای پیاده سازی Decision Tree یک کلاس تعریف شده است. به توضیح Function های مختلف این کلاس میپردازیم:

پیش از معرفی کلاس Decision Tree، کلاس Node تعریف شده که برای تعریف Node های تصمیم گیری به معنی شرایط تقسیم بندی است و Node های برگ به معنی کلاس اختصاص داده شده به آن، همچنین این کلاس Information Gain هر نود را هم ذخیره میکند.

Constructor:

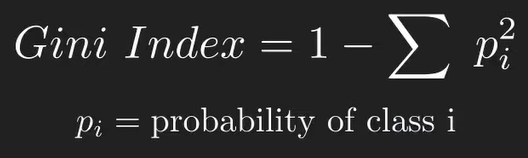
در این بخش شرایط توقف الگوریتم تعریف میشود.

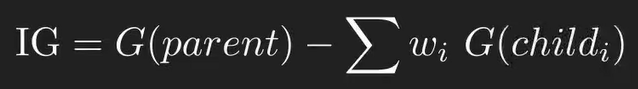
build\_tree:

مهم ترین بخش در پیاده سازی الگوریتم Decision Tree می باشد. در این فانکشن ابتدا فیچرها را از نتیجه نهایی جدا میکنیم در متغیرهای X, Y، سپس به صورت بازگشتی این تابع را صدا میزنیم تا درخت ما ساخته شود. شرایط ادامه این الگوریتم کمتر شدن نمونه ها از حداقل نمونه های یک نود یا بیشتر شدن عمق درخت از ماکزیمم عمق تعریف شده است. در این قسمت ابتدا شاخه ای سمت چپ محاسبه میشوند و سپس سمت راست و در نهایت نود تصمیم گیری که شامل شرایط جداسازی است برگشت می شود و در انتها کلاس نودهای نهایی (leaf nodes) مشخص میشود.

get\_best\_split:

در این فانکشن به محاسبه بهترین جداسازی ممکن میپردازیم. ابتدا متغیر max\_information\_gain را برابر -inf قرار میدهیم سپس بین تمامی فیچرها حرکت میکنیم و تمامی مقادیر ممکن و یکتا برای فیچرها را استخراج میکنیم در حلقه دوم مرز جداسازی را برابر تک به تک مقادیر یکتای فیچر قرار میدهیم و در نهایت information gain به دست آمده از این جداسازی را محاسبه میکنیم و انقدر ادامه میدهیم تا به بیشترین information gain برسیم.

برای محاسبه information gain از رابطه زیر استفاده شده است:



پیاده سازی Random Forest

ابتدا در init متغیرهای اولیه از جمله تعداد درخت ها و نسبت نمونه های استفاده شده در درخت به نمونه های موجود مشخص میشود.

Fit:

این تابع برای آموزش الگوریتم استفاده میشود. در این بخش ابتدا برای هر درخت یک نمونه تصادفی از داده ها انتخاب میشود و امکان وجود نمونه تکراری وجود دارد. این درخت توسط کلاس قبلی Train میشود و به مجموعه درخت های تصمیم اضافه میشود.

سپس در توابع predict و aggregate\_prediction جوابی که بیشترین تکرار را داشته انتخاب میکنیم.

بهینه سازی با الگوریتم ژنتیک:

برای بهینه سازی از الگوریتم ژنتیک استفاده شده است. این الگوریتم در کنار Random Forest اثر هم افزایی ایجاد میکند که به طور کلی بازدهی را افزایش میدهد.

ما از الگوریتم ژنتیک برای بهینه سازی تعداد درخت ها و بیشترین عمق درخت استفاده کردیم. روش پیاده سازی الگوریتم به این صورت است که ابتدا یک مقدار رندوم برای تعداد درخت و یک مقدار رندوم برای ماکزیمم عمق درخت انتخاب می شود و سپس با اجرای الگوریتم ژنتیک و محاسبه fitness به مرور حالت های بهینه تر را پیدا میکنیم.

Fitness:

در این تابع مقدار دقت را با مقادیر داده شده به تابع به عنوان ماکزیمم عمق و تعداد درخت محاسبه میکنیم و برمی گردانیم.

Crossover:

این تابع عملیات کراس اوور را برای زوج داده شده انجام میدهد به این صورت که امکان دارد مقدار تعداد درخت از یک فیچر را با ماکزیمم عمق از زوج دیگر ترکیب کند و یک فرزند جدید تولید کند.

Mutation:

این تابع با احتمالی عملیات جهش را انجام میدهد و مقداری رندوم را به تعداد درخت ها یا ماکزیمم عمق اضافه میکند.

Genetic\_algorithm

در این تابع با محاسبه fitness\_score تمامی individual ها را به سه دسته good، normal و bad تقسیم بندی میکنیم سپس بر روی تعدادی از اعضای هر کدام از این بخش ها عملیات های ترکیب و جهش انجام میشود و نسل بعدی تولید میشود و دوباره در نسل بعدی مقدار دقت کلی اگوریتم را میسنجیم و این عملیات را 5 نسل تکرار میکنیم که قابل تغییر است.

در نهایت با استفاده از الگوریتم ژنتیک بهترین پارامترها را برای الگوریتم Random Forest انتخاب میکنیم و با استفاده از آن این الگوریتم را آموزش میدهیم و با استفاده از داده های تست مورد ارزیابی قرار میدهیم که نتایج به صورت زیر است:

Best Hyperparameters: {'n\_trees': 203, 'max\_depth': 5}

Random Forest Accuracy with Best Hyperparameters: 0.9193548387096774

نتیجه گیری:

همون طور که مشاهده میشود به دقتی در حدود 92% رسیدیم در صورتی که با استفاده از الگوریتم Decision Tree فقط 80% درست پیش بینی میکردیم. و میتوان اثر هم افزای این دو الگوریتم و همچنین مزیت های استفاده از Random Forest را به خوبی مشاهده کرد. همچنین ماتریس Confusion به صورت زیر است.

Confusion Matrix

|  |  |
| --- | --- |
| 5 | 2 |
| 55 | 0 |

بخش دوم

در این بخش با اجرای الگوریتم با Feature های مختلف تاثیر هر کدام را در فرایند تصمیم گیری بررسی میکنیم:

ابتدا با smoking شروع میکنیم:

Best Hyperparameters: {'n\_trees': 42, 'max\_depth': 3}

Random Forest Accuracy with Best Hyperparameters: 0.8870967741935484

همان طور که مشاهده میشود حدود 88 درصد دقیق هستیم.

Confusion Matrix

[ 0 7]

[ 0 55]

سپس اثر خس خس سینه (WHEEZING) را بررسی میکنیم:

Best Hyperparameters: {'n\_trees': 46, 'max\_depth': 1}

Random Forest Accuracy with Best Hyperparameters: 0.8709677419354839

حدود 87 درصد دقت داریم

Confusion Matrix

[ 0 7]

[ 1 54]

و در آخر درد در سینه را بررسی میکنیم:

Best Hyperparameters: {'n\_trees': 5, 'max\_depth': 7}

Random Forest Accuracy with Best Hyperparameters: 0.8709677419354839

همان طور که میبینیم مهم ترین عامل سیگار کشیدن است.

مراجع :

* <https://www.who.int/news-room/fact-sheets/detail/lung-cancer>
* <https://www.mayoclinic.org/diseases-conditions/lung-cancer/symptoms-causes/syc-20374620>
* https://amarpishro.com/data-analysis/decision-tree
* [What Is Random Forest? A Complete Guide | Built In](https://builtin.com/data-science/random-forest-algorithm)
* https:// jsums.medsab.ac.ir/article\_710.html
* Lynch CM, Abdollahi B, Fuqua JD, de Carlo AR, Bartholomai JA, Balgemann RN, van Berkel VH, Frieboes HB. Prediction of lung cancer patient survival via supervised machine learning classification techniques. Int J Med Inform. 2017 Dec;108:1-8. doi: 10.1016/j.ijmedinf.2017.09.013. Epub 2017 Sep 25. PMID: 29132615; PMCID: PMC5726571.
* Decision trees: from efficient prediction to responsible AI : Hendrik Blockeel & Laurens Devos
* https://betterdatascience.com/mml-random-forest