دانشگاه آزاد اسلامی واحد تهران جنوب

دانشکده مهندسی پزشکی، گروه بیوالکتریک

رشته / گرایش: مهندسی پزشکی/بیوالکتریک

عنوان:

تشخیص بیماری صرع تمپورال راست از طریق تحلیل تصاویر دریافتی از دستگاه ام آر آی

استاد:

جناب آقای مهدی اسلامی

پژوهشگر:

نادیا جلالی فراهانی

شماره دانشجویی:

۴۰۰۱۴۱۴۰۱۱۱۰۳۷

بهار ۱۴۰۲


بسم الله الرحمن الرحيم

فهرست مطالب

فهرست شکل ها

فهرست جدول ها

چکیده پایان نامه :

مغز یکی از بزرگترین و پیچیده ترین ارگان های بدن انسان است و از بیش از ۱۰۰ بیلیون سلول عصبی و ترلیون ها ارتباط سیناپسی تشکلی شده است. تشنج یکی از مشکلاتی است که برای تعداد افراد زیادی سالانه در دنیا ایجاد میشود که میتواند ناشی از ضربه یا حمله مفزی باشد اما اکثرا دلیل مشخصی برای ان یافت نمیشود. تا به حال پژوهش های متعددی در خصوص تشخیص بیماری صرع با روشهای مختلفی انجام شده است اما در خصوص تشخیص بیماری صرع تمپورال راست از طریق تحلیل تصاویر دریافتی از دستگاه ام آر آی بررسی اندکی شده است.از آنجا که وجود اطلاعات صحیح و منسجم یکی از ملزومات موفقیت در علوم پزشکی مدرن محسوب می شود و اطلاعات شامل دو بخش داده و دانش است که چگونگی جمع آوری ذخیره و بازیابی داده ها در نظریه پایگاه داده مورد مطالعه قرار میگیرد در مهندسی دانش ،توجه اصلی به دانش روشهای فرمول بندی و بکارگیری شبکه عصبی آن در علوم مختلف است دانش استنباطی از زمینه ها و عرصه های مورد علاقه کارشناسان و متخصصان یا استنتاجی از مجموعه داده ها می باشد استنتاج دانش از مجموعه داده های معمولا ذخیره شده در پایگاه داده های بزرگ و داده کاوی است. در داده کاوی روشهای جدید در کنار روشهای آماری مورد استفاده قرار میگیرد خاستگاه روشهای جدید هوش مصنوعی است در واقع این روشها با اکتشاف داده ها از پایگاه داده روابط پیش بینی نشده و ناشناخته ی میان آنهارا مشخص می کند. در این پروژه بر آن شدیم تا با پیاده سازی الگوریتم های یادگیری ماشین KNN، SVM، MLP، LR و RF به مقایسه ی روش های بعد از عمل تشنج پرداختیم. سپس به مقایسه الگوریتم های پیاده سازی شده پرداخته شد و قابلیت های هر کدام در تست و آموزش انجام شد. در این کار ابتدا بیماران را به‌عنوان دیتاست هایی از شاخص های قبل عمل و بعد عمل در نظر گرفته شد تا شباهت‌ها/تفاوت‌های بین پروفایل‌های بیماری شناسایی شود. سپس به بررسی روش پیشنهادی برای تشخیص بیماری بعد از تشنج به روش یادگیری ماشین و استفاده از تئوری شبکه عصبی و رگرسیون ماشین بردار پشتیبان

و درخت تصمیم پرداخته شد. و هدف پیش بینی NAM بود. در این تحقیق ابتدا به بررسی الگوریتم های ارایه شده در شناسایی بیماری تشنج پرداخته شد. نهایتا اطلاعات آموزش یافته مدل های یادگیری ماشین محاسبه و در ادامه نتایج و دقت و همگرایی الگوریتم ها بررسی شد. مشخص شد که مدل جنگل تصادفی بهترین مدل پیشنهادی است. در نتیجه الگوریتم جنگل تصادفی به دلیل تجزیه پذیری داده و عدم برازش (overfitting) به نسبت مدلهای دیگر بهتر شد و همانطور که مشاهده شد دقت ۰,۹ درصد را بدست آورد.

کلید واژه ها: مغز، صرع، تصویر سازی و تصویر برداری تشدید مغناطیسی، آموزش ماشین

فصل اول: کلیات تحقیق

۱-۱- بیان مساله

تشنج یک بیماری نیست و تنها نشانه ای است که از انجام نشدن مناسب کار مغز می گوید. وقتی مغز به طور طبیعی کار کند.یک سری امواج الکتریکی ایجاد می کند که این امواج مانند الکتریسیته در مسیر اعصاب مغزی عبور می کنند. در حالت تشنج یک جرقه الکتریسیته مانند برق در آسمان ایجاد می شود که این جرقه و طوفان الکتریکی بسته به محل خود در مغز باعث علایم مختلف شده که نوع تشنج و صرع را تعیین می کنند .اپی لپسی یا صرع حالتی است که یک سری از این تشنج ها و طوفان های الکتریکی در مغز به طور مکرر و با فاصله زمانی به طور خود به خود ایجاد شده و خاموش شوند.اگر چه تشنج از عالیم بارز صرع است ولی همه تشنج ها منشا صرعی ندارند تشنج ممکن است به دالیل مختلف ایجاد شود .بنابراین یک بار تشنج هیچ گاه صرع محسوب نمی شود، بلکه دو و یا سه بار تشنج باید رخ دهد تا پزشک تشخیص بیماری صرع را مطرح کند [۱] . صرع به علت تخلیه الکتریکی غیر طبیعی در گروهی از نورون های مغزی است که باعث بروز تشنج های مکرر می شود [۲] و پس از سکته مغزی، دومین عامل بیماری های سیستم عصبی مرکزی می باشد و حدود ۰/۵ تا ۱ درصد مردم دنیا به این بیماری مبتلا هستند [۳] . صرع عبارت اند از مجموعه ای از اختلالات عصبی مزمن پزشکی بلند مدت که با حمله ی صرعی مشخص شده است این حالت ممکن است بسیار خفیف باشد و یا طوالنی مدت و با لرزش شدید اندام ها همراه است.در صرع حمالت به طور مکرر روی می دهد.

با توجه به سلول های آسیب دیده ی قشر مغز، حمالت صرع به دو نوع حمالت صرعی جزئی و عمومی تقسیم شده است. جهت تشخیص صرع تفاوت های سیگنال های مغزی در افراد نرمال، صرع خفیف و صرع شدید مورد بررسی قرار داده شده است. روش های مختلفی برای تشخیص صرع ارائه شده است که عبارت اند از تشخیص به وسیله دستگاه الکتروانسفالوگراف با قرار دادن الکترودهایی روی پوست سر و ثبت سیگنال های مغزی، انواع اسکن های مغزی مانند CT (ترموگرافی کامپیوتری)، PET (ترموگرافی از طریق انتشار پوزیترون)، MRI (تصویر

برداری به وسیله میدان مغناطیسی تشدید شده) و SPECT (توموگرافی کامپیوتری به وسیله انتشار یک نوترون) . [۴]

از آن جا که صرع بیماری شایعی است و بیشتر در سنین جوانی اتفاق می افتد، عدم توجه به آن می تواند بر سطح علمی و رفتار اجتماعی فرد و اقتصاد و سازندگی یک کشور تاثیر منفی داشته باشد. اپیدمیولوژی این بیماری با توجه به عوامل مختلفی از جمله هرم سنی، نوع و روش تغذیه، تفاوت های جغرافیایی و ... متفاوت است. بنابراین بررسی ان در هر منطقه ای به منظور تسهیل در تشخیص و درمان آن ضروری به نظر می رسد. میزان شیوع صرع در ایران یک تا سه درصد کل جمعیت براورد شده است [۵] .

MRI روشی است که می توان با کمک گرفتن از آن تصاویر بسیار دقیق و واضحی از اندامهای درون بدن بدست آورد. [۶] در این نوع تصویر برداری، تصاویر کامال تفکیک شده ای از بخش های مختلف مغز گرفته می شود. وضعیت بیماران مبتال به سردرد حمله ای و ناگهانی؛ ضعف و دور بینی، با MRI مغز قابل بررسی است MRI . مغز برای تکمیل تصویربرداری به وسیله سی تی اسکن و در مواردی نامشخص بودن تصاویر دریافتی به وسیله scan - CT نیز استفاده میشود. این مطاله با هدف تشخیص بیماری صرع تمپورال راست از طریق تحلیل تصاویر دریافتی با دستگاه MRI می باشد.

۱-۲- اهمیت و ضرورت تحقیق

تا به حال پژوهش های متعددی در خصوص تشخیص بیماری صرع با روشهای مختلفی انجام شده است اما در خصوص تشخیص بیماری صرع تمپورال راست از طریق تحلیل تصاویر دریافتی از دستگاه ام آر آی بررسی اندکی شده است.

در مطالعه ای که در سال ۲۰۰۹ توسط WD GALLARD و همکاران در امریکا به انجام رسید گایدالین تصویربرداری در کودکان و نوزادان با بروز تشنج اخیر مورد ارزیابی قرار گرفت. در این مطالعه که به صورت Review انجام شد، تمام مقاالتی که به صورت گذشته و با آینده نگر با تعداد نمونه بیش از ۳۰ مورد که در آنها scan - CTو یا MRI در بررسی کودکان با تشنج مورد

استفاده قرار گرفته بود مطالعه شدند. نتایج مطالعه نشان داد بهترین موارد استفاده از تصویربرداری در چنین کودکانی مواردی است که درگیری قسمتی از مغز مورد شک بوده و یا سندرم های مربوط به صرع مورد نظر می باشد. از میان این دو روش تصویربرداری، MRI با توجه به دقت باالتر و اشعه کمتر بیش از scan - CT مورد قبول می باشد [۷] .

در مطالعه ای که در سال ۲۰۰۵ توسط G GUSSARD و همکاران در فرانسه به انجام رسید تصویر برداری در صرع کودکان مورد بررسی قرار گرفت. در این مطالعه که به صورت مروری به انجام رسید نتایج مطالعه نشان داد انجام MRI نسبت به scan - CT در بررسی کودکان مبتال به صرع عالمت دار ارزشمندتر می باشد، انجام scan CT -در موارد اورژانسی و همچنین جهت دادههای تکمیلی MRI همچون بافتن کلسیفیکاسیون ها می تواند مفید باشد MRI .این امکان را به پزشک میدهد که در موارد تشنج تکرار شونده بیمار را پیگیری کرده و نیز در تعیین پروگنوز مفید می باشد [۸] .

در مطالعه ای که در سال ۲۰۰۲ توسط RC Scott و همکاران در انگلستان به انجام رسید نتایج انجام MRI در طی ۵ روز نخست بعد از بروز صرعStatus در کودکان مورد بررسی قرار گرفت. در این مطالعه ۳۵ بیمار وارد مطالعه شدند. نتایج مطالعه نشان

داد اندازه هیپوکامپ در بیماران با تشنج بدون تب طول کشیده بیش از جمعیت نرمال بوده است. در نتایج MRI بیماران با تشنج بدون تب طول کشیده نشان داد که در این بیماران تغییرات گسترده ای در MRI مشهود است مانند افزایش سیگنال درT2 [۹] در مطالعه ای که در سال ۲۰۱۲ توسط M young و همکاران در آمریکا به انجام رسید ، لزوم استفاده از MRI در پیگیری کودکان با SPILEPTICUS STATUS CONVULSIVE مورد بررسی قرار گرفت. در این مطالعه ۸۰ بیمار در رده سنی یک ماه تا ۱۶ سال وارد مطالعه شدند و در طی ۱۳ هفته از بروز تشنج از آنها MRI گرفته شد. نتایج مطالعه نشان داد انجام MRI مغز در چنین کودکانی به خصوص در مواردی که قبال MRI انجام نشده است، اختالالت نورولوژیک پایدار ایجاد شده است و در کسانی که حملات تکرار شده است می تواند مفید باشد [۱۰] .

در مطالعه ای که در سال ۲۰۰۴ توسط WoneladaromS و همکاران در تایلند به انجام رسید انجام MRI در کودکان مبتال به صرع مورد بررسی قرار گرفت. در این مطالعه که به صورت گذشته نگر به انجام رسید ۱۰۰ بیمار با سن کمتر از ۱۵ سال وارد مطالعه شدند. نتایج مطالعه نشان داد MRI با نشان دادن اختالالت موجود در مغز اتیولوژی صرع را حدی تعیین می کند [۱۱] . در مطالعه ای که در سال ۲۰۰۰ توسط J Maytal و همکاران در آمریکا به انجام رسید نقش scan - CT در بررسی کودکان با اولین حمله تشنج در اورژانس مورد بررسی قرار گرفت. در این مطالعه ۶۶ بیمار با میانگین سنی ۴/۹ سال و به صورت گذشته نگر مورد بررسی قرار گرفتند. نتایج مطالعه نشان داد تنها ۲۱/۲ ٪ بیماران مورد بررسی دارای نتایج scan - CT غیر طبیعی بودند که نشان داد انجام scan - CT در چنین کودکانی باید به اندیکاسیون های خاصی محدود شود [۱۲] . در مطالعه ای که در سال ۱۹۹۸

توسط MA Garvey و همکاران در آمریکا به انجام رسید انجام scan - CT در کودکان با تب مورد ارزیابی قرار گرفت. در این مطالعه ۱۰۷ بیمار به صورت گذشته نگر وارد مطالعه شدند. نتایج مطالعه نشان داد در یک کودک با تشنج با تب در CT معموال ضابعه غیر طبیعی دیده نمی شود [۱۳] .

در مطالعه ای که در سال ۱۳۸۷ توسط فالح و همکاران در یزد انجام شد نتایج scan - CT مغزی کودکان و ارتباط یافته های آن با علت درخواست scan - CT مورد بررسی قرار گرفت. در این مطالعه که به صورت مقطعی به انجام رسید ۱۰۰ کودک وارد مطالعه شدند. نتایج مطالعه نشان داد scan - CT در منتژیت بدون اختالل هوشیاری و اولین تشنج ژنرالیزه و با معاینه عصبی نرمال، بدون عالئم خطر چندان کمک کننده نبود و MRI در تشخیص تأخیر تکاملی و اختالل ساختمانی مغز مفیدتر می باشد [۱۴] .

در مطالعه ای که در سال ۱۳۸۷ توسط ناصحی و همکاران در مازندران به انجام رسید یافته های پاراکلینیک و نتایج درمانی در کودکان مبتال به صرع مقاوم مورد بررسی قرار گرفت. در این مطالعه که به صورت مقطعی انجام شد نتایج مطالعه نشان داد نتایج سی تی اسکن در ۳۹/۳ ٪و MRI در ۴۲/۷ ٪بیماران غیر طبیعی بوده است [۱۵] .

۱-۳- اهداف تحقیق

تعیین و مقایسه باند طیف و تغییرات غیرطبیعی بودن MRI برای مغز افراد به مبتال صرع تمپورال راست

تعیین باند تتا MRI مغز غیر طبیعی افراد مبتلا به صرع تمپورال راست و افراد سالم

تعیین باند دلتا MRI غیرطبیعی بودن بدون تزریق مغز افراد مبتال به صرع تمپورال راست و افراد سالم

تعیین باند گاما MRI غیرطبیعی بودن بدون تزریق مغز افراد مبتال به صرع تمپورال راست و افراد سالم

فصل دوم: مبانی نظری تحقیق و کارهای پیشین

۲-۱- آناتومی و فیزیولوژِی مغز

مغز یکی از بزرگترین و پیچیده ترین ارگان های بدن انسان است و از بیش از ۱۰۰ بیلیون سلول عصبی و ترلیون ها ارتباط سیناپسی تشکلی شده است

مغز از تعداد زسادی ناحیه ی اختصاص یافته که با هم کار میکنند، تشکیل شده است

۱- Cortex یا غشا که خارجی ترین لایه مغز است و تفکر از این ناحیه آغاز میشود

۲- ریشه یا stem که بین نخاع و باقی مغز واقع شده است و اعمال اساسی مانند تنفس و خواب در این بخش کنترل می شود

۳- گره های پایه ای (basal gangelion) دسته هایی ساختاری در مرکز مغز هستند که این بخش پیام های بین بخش های مختلف مغز را هماهنگ میکند

۴- مخچه (cerebellum) در پایه و پشت مغز واقع شده که وظیفه هماهنگی و تعادل بدن را دارد.

همچنین مغز به چند بخش (lobe) تقسیم میشود

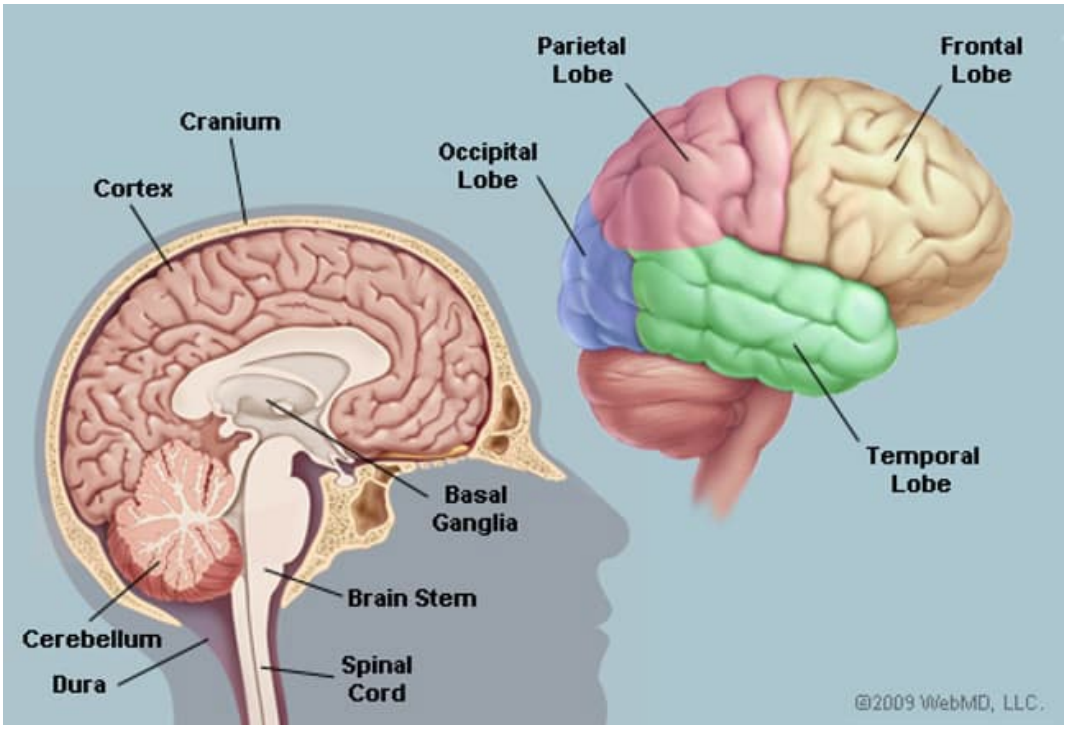
۱- Frontal که مسول حل مساله و قضاوت را دارد

۲- Parietal که دست خط، لامسه و پوزیشن بدن را مدیریت میکند

۳- Temporal که بخش حافظه و شنیداری را شامل میشود

۴- Occipital که حاوی سیستم پردازش بصری است [۱۶] .

در شکل زیر بخش های مختلف مغز را مشاهده کنید.



شکل ۱-۱- اناتومی مغز انسان

مشکلات مغز شامل موارد زیر است

۱-۴- Headache

۱-۵- Stroke

۱-۶- Brain aneurysm

۱-۷- Subdural hematoma

۱-۸- Epidural hematoma

۱-۹- Intracerebral hemorrhage

۱-۱۰- Concussion

۱-۱۱- Cerebral edema

۱-۱۲- Brain tumor

۱-۱۳- Glioblastoma

۱-۱۴- Hydrocephalus

۱-۱۵- Normal pressure hydrocephalus

۱-۱۶- Meningitis

۱-۱۷- Encephalitis

۱-۱۸- Traumatic brain injury

۱-۱۹- Parkinson's disease

۱-۲۰- Huntington's disease

۱-۲۱- Epilepsy

۱-۲۲- Dementia

۱-۲۳- Alzheimer’s disease

۱-۲۴- Brain abscess

به جهت سنجش و ارزیابی سلامتی مغز تست های مختلفی از جمله موارد زیر انجام میشود.

۱-۲۵- Computed tomography (CT scan)

۱-۲۶- Magnetic resonance imaging (MRI scan)

۱-۲۷- Angiography (brain angiogram)

۱-۲۸- Magnetic resonance angiography (MRA)

۱-۲۹- Lumbar puncture (spinal tap)

۱-۳۰- Electroencephalogram (EEG)

۱-۳۱- Neurocognitive testing

۱-۳۲- Brain biopsy

درمان هایی که بر روی مغز انجام میشود شامل موارد زیر است

۱- Antiplatelet agents Thrombolytics

۲- Cholinesterase inhibitors

۳- Antibiotics

۴- Levodopa

۵- Brain surgery

۶- Ventriculostomy

۷- Craniotomy

۸- Lumbar drain

۹- Radiation therapy

۲-۲- تاریخچه neuroimaging

اولین تصویربرداری مغز توسز Angelo Mosso در سال ۱۸۸۰ ابداع شد که بیش از یک قرن ناشناخته ماند ولی متغییر هایی مانند نسبت سیگنال به نویز که همچنان کاربرد دارند توسط او تعریف شد [۱۷] . در این روش اندازه گیری غیر تهاجمی وابسته به خون محیطی مغر هنگام احساسات و فعالیت های هوشی انجام شد [۱۸] . سپس در آغاز سال های ۱۹۰۰ (Walter Dandy جراح مغز و اعصاب) تکنیک با استفاده از مایع مغزی نخاعی اطراف مغز و جایگزینی آن با حباب هوا برای تصویربرداری اشعه ایکس بهتر، برنامه ریزی شد که روشی بسیار نا ایمن با خطر افزایش فشار مغز و عفونت بود پس MRI و CT در سال های ۱۹۷۰ گسترش پیدا کرد که در عین داشتن ایمینی بالاتر، جزییات بیشتری از مغز را آشکار میکرد [۱۹ و ۲۰] .

در سال ۱۹۲۷ انژیوگرافی عروق مغزی توسط Egas Moniz ارایه شد که برنده نوبل فیزولوژی شد

از سال ۱۹۶۰ روشهای مختلفی برای تحلیل تصاویر در موسسه ماساچوست دانشگاه مریلند به جهت بهبود تصاویر آغاز شد و هزینه های تحلیل در آن زمان بالا بود (William H. Oldendorf) [۲۱] .

اکنون روش های SPECT و PET Scan در حال اجرا و پیشترفت است پراکه تنها ماند ام آر آی و سی تی اسکن تنها تصویر استاتیک ایجاد نمیکند بلکه عملکرد مغز را نیز بررسی میکند و دانشمندان با استفاده ازMRI و CT و SPECT تصاویر FMRI را ایجاد کردند که دری به سوی مشاهده فعالیت مغزی باز کرده است.

۲-۳- اپیلپسی

بیماری صرع در زمان های خیلی قدیم وجود داشته و در زمان رونسانس به نظر میرسید که فقط درمان آن از طریق مذهب صورت میگرفته است .نزدیک اواخر قرن ۱۴ پزشکان وارد عرصه شده و به بررسی عوامل ایجاد اپیلپسی و ارتباط بیماری به تومورها و بیماری ها و جمع آوری اطلاعات پرداختند [۲۲] .

اواخر قرن ۱۹ تغریفی پاتوفیزولوژیکی توسط Jackson ارایه شد که در آن بیماری اپیلپسی دسته بندی شد و نتایج آن نیز با انسفالوگرام قابل تایید بود و در صورتی که هر دو بخش مغز هنگام تشنج درگیر بود نام دسته بندی اپیلیپسی جنرال بر روی آن گذاشته شد [۲۳ و ۲۴] .

صَرع یا اپی‌لپسی عبارت است از مجموعه‌ای از اختلالات عصبی مزمن پزشکی یا بلند مدت که با حمله صرعی مشخص می‌شود [۲۵] . این حملات ممکن است بسیار خفیف و تقریباً غیرقابل شناسایی بوده یا برعکس طولانی مدت و با لرزش شدید همراه باشد [۲۶] . در صرع حملات به‌طور مکرر روی می‌دهد و هیچ دلیل ثابت و مشخصی ندارد، در حالی که حملاتی را که به دلایل خاص روی می‌دهد، نباید به عنوان حمله صرعی تلقی کرد [۲۷] .

در بیشتر موارد دلیل نامشخص است، اما صرع در برخی افراد به دلیل آسیب مغزی، سرطان مغز، و سوء مصرف دارو و الکل، و دلایل دیگر ایجاد می‌شود. حملات صرعی نتیجه فعالیت سلولی بیش از حد و غیرعادی عصب کورتیکال یا غشایی در مغز است. فرایند تشخیص معمولاً شامل حذف تمام شرایطی است که ممکن است علائم مشابهی نظیر سنکوپ را ایجاد کند، و نیز بررسی اینکه آیا هیچ دلیل لحظه‌ای دیگری وجود داشته‌است یا خیر. صرع را می‌توان با گرفتن نوار مغزی یا الکتروانسفالوگرافی نیز تأیید کرد.

صرع را نمی‌توان درمان کرد، اما حملات را می‌توان با دارو تا حدود ۷۰ درصد موارد کنترل کرد [۲۸] . در افرادی که حملات به دارو پاسخ نمی‌دهند، جراحی، تحریک عصبی یا تغییر در رژیم غذایی را می‌توان در نظر گرفت. تمامی سندرم‌های صرع مادام العمر نیستند و اکثر افراد تا جایی بهبود می‌یابند که دیگر نیازی به دارو ندارند.

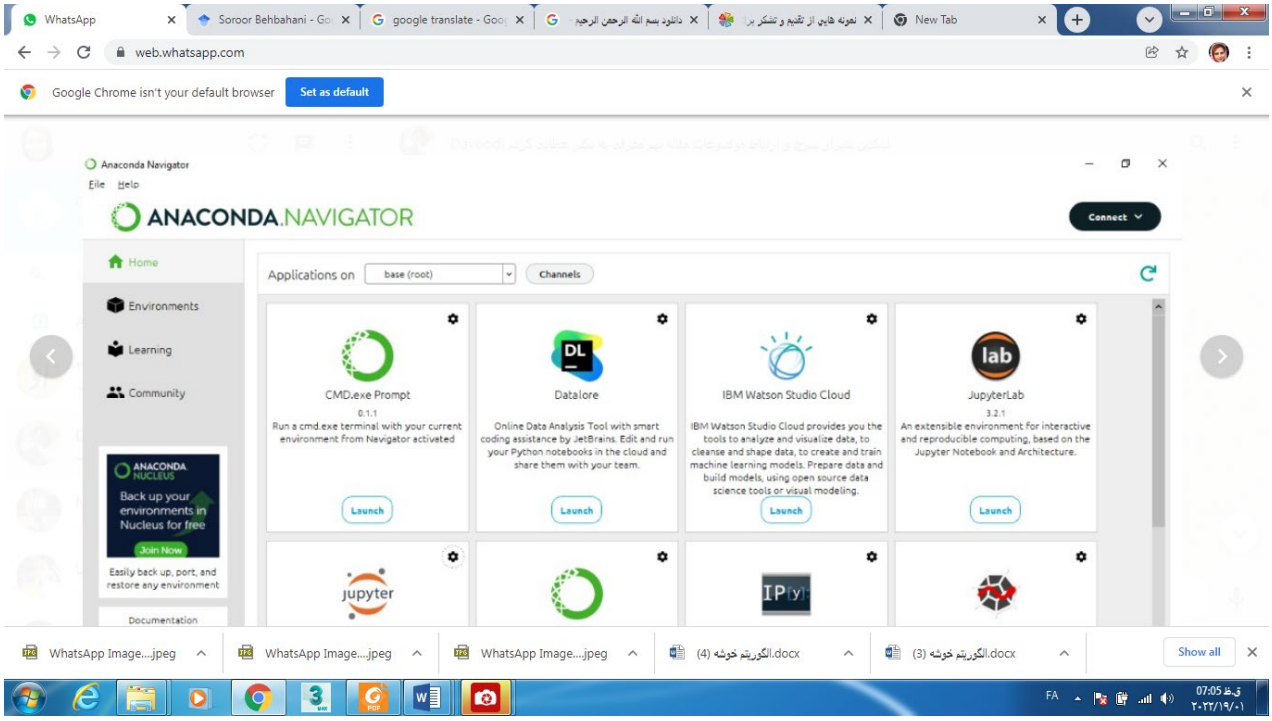
حدود ۱٪ درصد از جمعیت مردم جهان (۶۵ میلیون نفر) صرع دارند [۲۹] و تقریباً ۸۰٪ موارد در کشورهای در حال توسعه روی می‌دهند.صرع در افراد مسن‌تر رایج‌تر است [۳۰ و ۳۱] . در کشورهای توسعه یافته، شیوع موارد جدید بیشتر در نوزادان و سالمندان است [۳۲] . در کشورهای در حال توسعه این بیماری در کودکان بزرگتر و بزرگسالان جوان‌تر دیده می‌شود [۳۳] که دلیل آن تفاوت در فراوانی دلایل اصلی است. حدود ۵ تا ۱۰٪ از تمامی افراد یک حمله بی‌دلیل تا سن ۸۰ سالگی دارند [۳۴] و احتمال وقوع حمله دوم نیز بین ۴۰ و ۵۰٪ است [۳۵] . در بسیاری از نقاط دنیا افراد مبتلا به صرع حق رانندگی مشروط داشته یا کلا حق رانندگی ندارند [۳۶] اما بیشتر این افراد بعد از یک مدت بدون حمله می‌توانند رانندگی را مجدداً آغاز کنند. متخصصان معتقدند حدود ۷۰ تا ۸۰ درصد بیماری صرع پس از دو سال معالجه، به‌طور کامل درمان می‌شود و پزشک معالج می‌تواند داروی بیمار را قطع کند [۳۷] .

در پژوهشی که توسط فخوری و همکارانش [۳۸] انجام شد ۱۲۷ مورد حمله تشنجی در ۱۹ بیمار دارای صرع تمپورال به روش EEG مورد ارزیابی قرار گرفت و نتایج نشان داد که در صرع تمپورال چپ، حرکات تشنجی شدیدتر و همچنین تشنج های ثانویه بیشتری نسبت به بقیه رخ داد و در مقابل مشکلات گفتاری و برگشت سریعتر به خالت نرمال بیشتر در صرع تمپورال راست رخ میدهد.

فصل سوم: روش پیشنهادی

۳-۱- داده کاوی بر اساس پایتون

پروژه بر روی سیستم ویندوز اجرا میشود. با نصب برنامه آناکوندا پایتون و پکیج های آن نیز نصب میشود که در این صورن اشکالات فنی پیاده سازی در کمترین میزان خود قرار خواهد گرفت در شکل زیر محیط این نرم آفزار را مشاهده میکنید



شکل ۳-۱- محیط نرم افزار آناکندا

پس از نصب در بخش jupiterlab که محیطی live code برای پایتون است وارد شده و کدهای برنامه نوشته میشود.

دیتاست :

برای طبقه‌بندی نتایج بعد از عمل جراحی صرع، جروم انگل طرح زیر را پیشنهاد کرد [۴۰و۳۹] مقیاس نتیجه جراحی صرع انگل، که به استاندارد واقعی برای گزارش نتایج در ادبیات پزشکی تبدیل شده است:

کلاس I: بدون تشنج ناتوان کننده

کلاس II: تشنج های ناتوان کننده نادر (تقریباً بدون تشنج)

کلاس III: بهبود ارزشمند

کلاس IV: هیچ پیشرفت قابل توجهی وجود ندارد

ENG_b: Engel score = 1 (1 = ENG+); Engel score = 2, 3 or 4 (2 = ENG-)

- (ENG+) optimal postoperative clinical outcome: the Engel score of I, no seizures observed 2 years after surgery;

(ENG-) suboptimal postoperative clinical outcome: the Engel score of II-IV: persistence of more or less frequent and more or less severe seizures 2 years after surgery;

- (NAM+) optimal postoperative naming outcome: the naming score at 2 years after surgery did not decline or even improved compared to the presurgical baseline;

- (NAM-) suboptimal postoperative naming outcome: the naming score at 2 years after surgery has decreased compared to the presurgical baseline

Surgery information

AGE: Age at time of neurosurgery (in years)

DIST: distance of the post-op workup from the date of surgery (in years)

ENG: Engel score at two years post-surgery (Engel, 1993, 2013)

ENG_b: Engel score = 1 (1 = ENG+); Engel score = 2, 3 or 4 (2 = ENG-)

NAM: Naming DO80 at two years post-surgery (Metz-Lutz et al., 1991)

NAM_b: NAM Reliable Change Index > -1.28 SD (1 = NAM +); NAM RCI ≤ -1.28 SD (2 = NAM-)

Demographics

AGE: Age of patient (in years)

EDU: Educational level (in years)

GEN: Gender (Male/Female)

HDS: Handedness (Edinburgh laterality index)

HDS_b: Handedness (Left/Right)

Epilepsy related information

HEM: Hemispheric lateralization of the temporal lobe epilepsy (L = Left; R = right)

ASO: Age of seizures onset (in years)

AHS: Hippocampal asymmetry (absolute volume difference)

DUR: Duration of the epilepsy (in years)

DUR_b: Duration ≤ 15years (1); > 15years (2)

HS_b: Hippocampal sclerosis (yes/no)

FRQ: Frequency of seizures (by months)

AED: Antiepileptic drugs (daily taken)

SEV: Severity of the epilepsy (composite score deriving from DUR, FRQ and AED)

SEV_b: Severity composite score ≤ 6 (1); > 6 (2)

۳-۲- داده کاوی در علوم پزشکی

کاوش داده به زیر شـــاخه ای از علم آمار به نام « تحلیل اکتشـــافی داده» مرتبط اســـت که در حوزه ی ارقام آماری، اهداف و رویکرد های مشابهی با یکدیگر دارند. همچنین با بخش هایی از علم هوش مصنوعی بنام کشف دانش و یادگیری ما شینی ارتباط بـ سیار تنگاتنگی دارد. اما ویژگی مهم و متمایز داده کاوی آن ا ست که با حجم داده ی بسیار بالایی سر و کار دارد. با این وجود ایده ها و تکنیک‌های این حوزه های مطالعاتی مرتبط، در مسایل داده کاوی نیز قابل ا ستفاده هـستند. مقیاس پذیری به نـسبت سایز داده یک معیار مهم و جدید در مفاهیم کاوش داده است [۳۹].

یافتن روشها و الگو های مفید در پایگاه های داده یک تعریف از داده کاوی است. در واقع می توان تصور کرد که تمامی جستجو های پایگاه داده چنین هستند.

۳-۲-۱- تعریف تئوریک از داده کاوی

داده کاوی عبارت اســـت از فرآیند (نیمه)خودکار اســـتخراج دانش (در قالب الگوهای پنهان) از مجموعه اطلاعات ورودی. معمولا آگاهی اندکی در مورد دانش هدف وجود دارد و ورودی عمدتا بسیار حجیم و پردازش دستی آن ناممکن است.

نتایج حاصل از داده کاوی، با روشهای سنتی پردازش اطلاعات (گزارش گیری) قابل دستیابی نیست وخودکار یا نیمه خودکار بودن داده کاوی به معنای حداقل نیاز به دخالت کاربر ا ست. انواع اطلاعات (و نه صرفا اطلاعات عددی) در داده کاوی قابل پردازش می باشند.

۳-۲-۲- شبکه عصبی در داده کاوی

وجود اطلاعات صحیح و منسجم یکی از ملزومات موفقیت در علوم پزشکی مدرن محسوب می شود اطلاعات شامل دو بخش داده و دانش است که چگونگی جمع آوری ذخیره و بازیابی داده ها در نظریه پایگاه داده مورد مطالعه قرار میگیرد در مهند سی دانش توجه ا صلی به دانش رو شهای فرمول بندی و بکارگیری شبکه عصبی آن در علوم مختلف است دانش استنباطی از زمینه ها و عرصه های مورد علاقه کارشناسان و متخصصان یا ا ستنتاجی از مجموعه داده ها می با شد ا ستنتاج دانش از مجموعه داده های معمولا ذخیره شده در پایگاه داده های بزرگ داده کاوی نامیده می شود روشهای آماری همچون شبکه عصبی جزء روشهای کلاسیک استنتاج دانش به شمار میایند [۴۰] . در داده کاوی رو شهای جدید در کنار رو شهای آماری مورد ا ستفاده قرار میگیرد خا ستگاه روشهای جدید هوش مصنوعی است در واقع این روشها با اکتشاف داده ها از پایگاه داده روابط پیش بینی نشده و ناشناخته ی میان آنها را مشخص می کند.

۳-۳- کاربردهای داده کاوی

کاربردهای تجاری: تقریبا در تمام ســـازمانها و انواع تجارتها، به دلیل وجود اطلاعات، می توان داده کاوی را مورد استفاده قرار داد از جمله در :

- پیش بینی بیماری[۴۱]
- تحلیل سبد خرید[۴۲]
- شناسائی طبقات بیماری‌های مختلف [۴۳]
- تعیین میزان تاثیر عوامل مختلفی نظیر تبلیغات، تخفیف، ... بر میزان و الگوهای فروش[۴۴]

کاربردهای علمی: مثلا اطلاعات جمع آوری شده در حوزه های مختلف شامل اطلاعات جغرافیائی، اطلاعات اقلیمی، اطلاعات پزشـــکی، نیاز مبرم به تکنیکهای داده کاوی، حداقل جهت ایجاد امکان تصـــور اطلاعات برای متخصصان احساس می شود.

حوزه پزشکی: تشخیص بیماریها براساس انواع اطلاعات (تصاویر پزشکی، مشخصات بیمار احتمالی)تشخیص ناهنجاریهائی که توسط انسان به سختی قابل تشخیص خواهند بود (لکه ها و نقاط خاص داخل چشم که نشانه شروع کوری ناشی از دیابت می باشد)

۳-۴- شبکه عصبی

در حالت کلی، یک شبکه عصبی زیستی از مجموعه یا مجموعه‌ای از نورون‌های به صورت فیزیکی به هم متصل یا از لحاظ عملکردی به هم وابسته تشکیل شده‌است. هر نورون می‌تواند به تعداد بسیار زیادی از نورون‌ها وصل باشد و تعداد کل نورون‌ها و اتصالات بین آن‌ها می‌تواند بسیار زیاد باشد. اتصالات، که به آن‌ها سیناپس گفته می‌شود، معمولاً از آکسون‌ها و دندریت‌ها تشکلیل شده‌اند.هوش مصنوعی و مدل سازی شناختی سعی بر این دارند که بعضی از خصوصیات شبکه‌های عصبی را شبیه سازی کنند. این دو اگرچه در روش‌هایشان به هم شبیه هستند اما هدف هوش مصنوعی حل مسائل مشخصی است در حالی که هدف مدل سازی شناختی ساخت مدل‌های ریاضی سامانه‌های نورونی زیستی است.روشی برای محاسبه است که بر پایه اتصال به هم پیوسته چندین واحد پردازشی ساخته میشود.شبکه از تعداد دلخواهی نرون تشکیل میشود که مجموعه ورودی را به خروجی ربط میدهند.هر نورون می‌تواند به تعداد بسیار زیادی از نرون‌ها وصل باشد و تعداد کل نرون‌ها و اتصالات بین آن‌ها می‌تواند بسیار زیاد باشد[۴۵] .

یک نورون مصنوعی سامانه‌ای است با تعداد زیادی ورودی و تنها یک خروجی. نورون دارای دو حالت می‌باشد، حالت آموزش و حالت عملکرد. در حالت آموزش نورون یاد می‌گیرد که در مقابل الگوهای ورودی خاص برانگیخته شود و یا در اصطلاح آتش کند. در حالت عملکرد وقتی یک الگوی ورودی شناسایی شده وارد شود، خروجی متناظر با آن ارائه می‌شود. اگر ورودی جزء ورودی‌های از پیش شناسایی شده نباشد، قوانین آتش برای بر انگیختگی یا عدم آن تصمیم گیری می‌کند.با کنار گذاشتن برخی از خواص حیاتی نورون‌ها و ارتباطات درونی آنها می‌توان یک مدل ابتدایی از نورون را به وسیله کامپیوتر شبیه سازی کرد.

شبکه های عصبی مصنوعی (Artificial Neural Network) الگویی برای پردازش اطلاعات می باشند که با تقلید از شبکه های عصبی بیولوژیکی مثل مغز انسان ساخته شده اند.عنصر کلیدی این الگو ساختار جدید سیستم پردازش اطلاعات آن می باشد و از تعداد زیادی عناصر (نرون) با ارتباطات قوی داخلی که هماهنگ با هم برای حل مسائل مخصوص کار می کنند تشکیل شده اند. شبکه های عصبی مصنوعی با پردازش روی داده های تجربی، دانش یا قانون نهفته در ورای داده ها را به ساختار شبکه منتقل می کند که به این عمل یادگیری می گویند. اصولاً توانایی یادگیری مهمترین ویژگی یک سیستم هوشمند است. سیستمی که بتواند یاد بگیرد منعطف تر است وساده تر برنامه ریزی میشود، بنابراین بهتر میتواند در مورد مسایل و معادلات جدید پاسخگو باشد.انسانها از زمانهای بسیار دور سعی بر آن داشتند که بیوفیزیولوژی مغز را دریابند چون همواره مسئله هوشمندی انسان و قابلیت یادگیری ، تعمیم، خلاقیت ،انعطاف پذیری و پردازش موازی در مغز برای بشر جالب بوده و بکارگیری این قابلیت ها در ماشینها بسیار مطلوب می نمود.روشهای الگوریتمیک برای پیاده سازی این خصایص در ماشین ها مناسب نمی باشند در نتیجه می بایست روشها مبتنی بر همان مدل های بیولوژیکی باشد.درست مثل انسان ها با استفاده از مثال ها آموزش می بیند ، همانطور که یک بچه با دیدن انواع مختلف از یک حیوان قادر به تشخیص آن می باشد.به عبارت دیگر شبکه عصبی مصنوعی یک سامانه پردازشی دادهها است که از مغز انسان ایده گرفته و پردازش دادهها را به عهده پردازندههای کوچک و بسیار زیادی سپرده که به صورت شبکهای به هم پیوسته و موازی با یکدیگر رفتار میکنند تا یک مسئله را حل نمایند. در این شبکهها به کمک دانش برنامه نویسی، ساختار دادهای طراحی میشود که میتواند همانند نورون عمل کند. که به این ساختارداده گره گفته میشود. بعد با ایجاد شبکهای بین این گرهها و اعمال یک الگوریتم آموزشی به آن ، شبکه را آموزش میدهند. در این حافظه یا شبکه عصبی گرهها دارای دو حالت فعال (روشن یا ۱) و غیرفعال (خاموش یا ۰) اند و هر یال (سیناپس یا ارتباط بین گرهها) دارای یک وزن میباشد. یالهای با وزن مثبت، موجب تحریک یا فعال کردن گره غیر فعال بعدی میشوند و یالهای با وزن منفی، گره متصل بعدی را غیر فعال یا مهار (در صورتی که فعال بوده باشد) میکنند[۴۵] .

۳-۵- ساختار شبکه‌های عصبی:

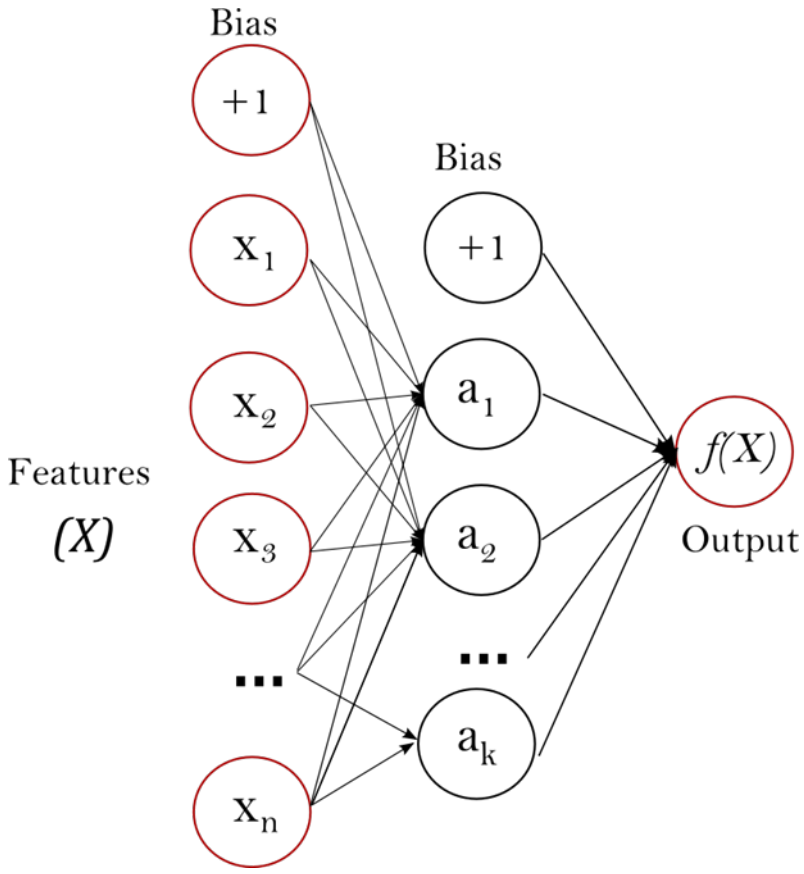
یک شبکه عصبی شامل اجزای سازنده لایه‌ها و وزن‌ها می‌باشد. رفتار شبکه نیز وابسته به ارتباط بین اعضا است.

در حالت کلی در شبکه‌های عصبی سه نوع لایه نورونی وجود دارد[۴۵] :

۱. لایه ورودی : دریافت اطلاعات خامی که به شبکه تغذیه شده‌است.

۲. لایه‌های پنهان : عملکرد این لایه‌ها به وسیله ورودی‌ها و وزن ارتباط بین آنها و لایه‌های پنهان تعیین می‌شود.

وزن‌های بین واحدهای ورودی و پنهان تعیین می‌کند که چه وقت یک واحد پنهان باید فعال شود.



شکل ۳-۲- ساختار شبکه عصبی مصنوعی

۳. لایه خروجی : عملکرد واحد خروجی بسته به فعالیت واحد پنهان و وزن ارتباط بین واحد پنهان و خروجی می‌باشد.

شبکه‌های تک لایه و چند لایه‌ای نیز وجود دارند که سازماندهی تک لایه که در آن تمام واحدها به یک لایه اتصال دارند بیشترین مورد استفاده را دارد و پتانسیل محاسباتی بیشتری نسبت به سازماندهی‌های چند لایه دارد. در شبکه‌های چند لایه واحدها به وسیله لایه‌ها شماره گذاری می‌شوند (به جای دنبال کردن شماره گذاری سراسری). هر دو لایه از یک شبکه به وسیله وزن‌ها و در واقع اتصالات با هم ارتباط می‌یابند. در شبکه‌های عصبی چند نوع اتصال و یا پیوند وزنی وجود دارد: پیشرو : بیشترین پیوندها از این نوع است که در آن سیگنال‌ها تنها در یک جهت حرکت می‌کنند. از ورودی به خروجی هیچ بازخوردی (حلقه) وجود ندارد. خروجی هر لایه بر همان لایه تاثیری ندارد. پسرو : داده‌ها از گره‌های لایه بالا به گره‌های لایه پایین بازخورانده می‌شوند. جانبی : خروجی گره‌های هر لایه به عنوان ورودی گره‌های همان لایه استفاده می‌شوند.

۳-۵-۱- شبکه عصبی پرسپترون:

این شبکه عصبی برمبنای یک واحد محاسباتی به نام پرسپترون ساخته میشود.یک پرسپترون برداری از ورودیها با مقادیر حقیقی را گرفته و یک ترکیب خطی از ورودی ها را محاسبه می کند.اگر حاصل از یک مقدار آستانه بیشتر بود خروجی پرسپترون برابر با 1 و در غیر اینصورت معادل 1- خواهد بود.شبکه های عصبی پرسپترون،به ویژه پرسپترون چندلایه در زمره کاربردی ترین شبکه های عصبی می باشند،این شبکه ها قادرند باانتخاب مناسب تعدادلایه ها و سلول های عصبی،که اغلب هم زیاد نیستند، یک نگاشت غیر خطی را با دقت دلخواه انجام دهند[۴۶]

.

۳-۶- تقسیم بندی شبکه‌های عصبی :

بر مبنای روش آموزش به چهار دسته تقسیم می‌شوند:

۱. وزن ثابت : آموزشی در کار نیست و مقادیر وزن‌ها به هنگام نمی‌شود.کاربرد آن در بهینه سازی اطلاعات ،کاهش حجم، تفکیک پذیری و فشرده سازی می باشد.

۲. آموزش بدون سرپرست : وزن‌ها فقط بر اساس ورودی‌ها اصلاح می‌شوند و خروجی مطلوب وجود ندارد تا با مقایسه خروجی شبکه با آن و تعیین مقدار خطا وزن‌ها اصلاح شود. وزن‌ها فقط بر اساس اطلاعات الگوهای ورودی به هنگام می‌شوند. هدف استخراج مشخصه‌های الگوهای ورودی بر اساس راهبرد خوشه یابی و یا دسته‌بندی و تشخیص شباهت‌ها (تشکیل گروه‌هایی با الگوی مشابه) می‌باشد، بدون اینکه خروجی یا کلاس‌های متناظر با الگوهای ورودی از قبل مشخص باشد. این یاد گیری معمولاً بر پایه شیوه برترین هم خوانی انجام می‌گیرد. شبکه بدون سرپرست وزن‌های خود را بر پایه خروجی حاصل شده از ورودی تغییر می‌دهد تا در برخورد بعدی پاسخ مناسبی را برای این ورودی داشته باشد. در نتیجه شبکه یاد می‌گیرد چگونه به ورودی پاسخ بدهد. اصولا هدف این است که با تکنیک نورون غالب نورونی که بیشترین تحریک آغازین را دارد برگزیده شود. بنابر این در شبکه‌های بدون سرپرست یافتن نورون غالب یکی از مهمترین کارها است [۴۷ و ۴۸] .

۳. آموزش با سرپرست : به ازای هر دسته از الگوهای ورودی خروجی‌های متناظر نیز به شبکه نشان داده می‌شود و تغییر وزن‌ها تا موقعی صورت می‌گیرد که اختلاف خروجی شبکه به ازای الگوهای آموزشی از خروجی‌های مطلوب در حد خطای قابل قبولی باشد. در این روش‌ها یا از خروجی‌ها به وزن‌ها ارتباط وجود دارد یا خلا به صورت پس انتشار از لایه خروجی به ورودی توزیع شده‌است و وزن‌ها اصلاح می‌شوند. هدف طرح شبکه‌ای است که ابتدا با استفاده از داده‌های آموزشی موجود، آموزش ببیند و سپس با ارائه بردار ورودی به شبکه که ممکن

است شبکه آن را قبلا فراگرفته یا نگرفته باشد کلاس آن را تشخیص دهد. چنین شبکه‌ای به طور گسترده برای کارهای تشخیص الگو به کار گرفته می‌شود.

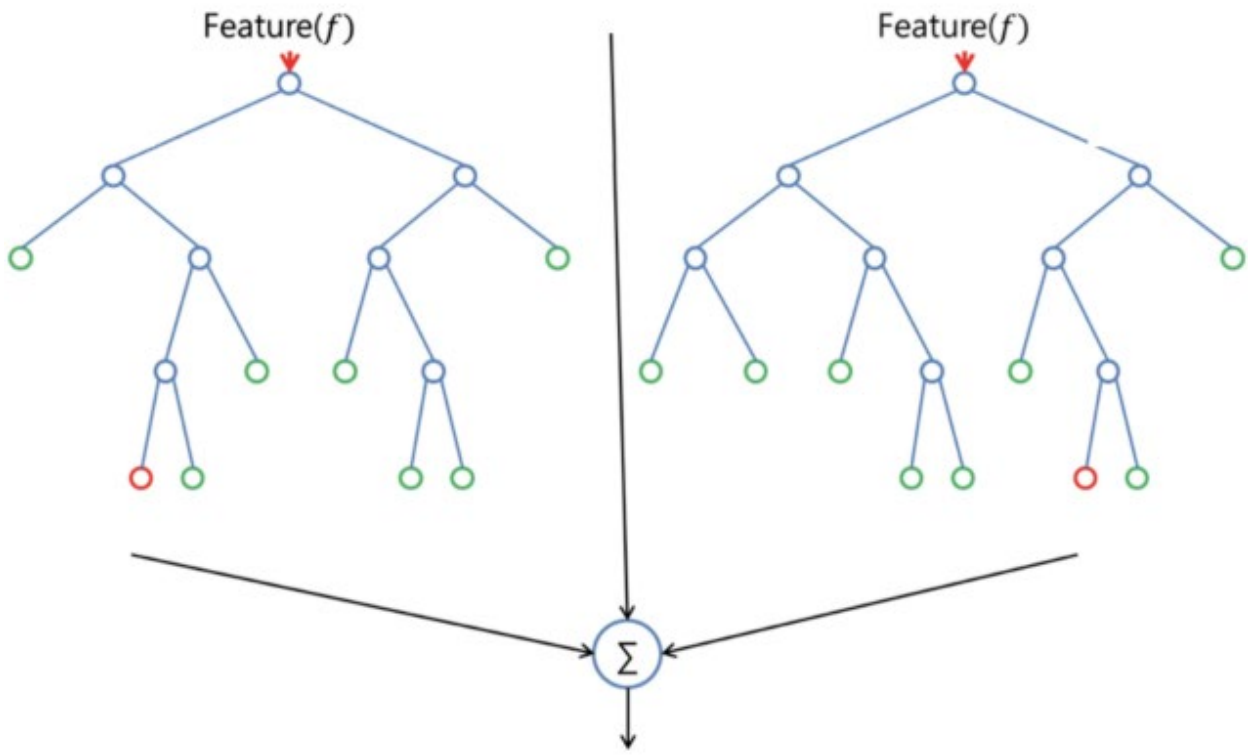
یکی دیگر از روش‌های آموزش شبکه عصبی استفاده از الگوریتم‌های تقریبی برای تخمین وزن های ورودی شبکه عصبی است که در این حالت می‌توان با استفاده از تقریب درست مناسب از الگوریتم شبکه عصبی و تعیین وزنی مناسب (که وزن مناسب را الگوریتم تعیین میکند) به پیش بینی دقیقتر شبکه عصبی کمک کرد.

۳-۷- الگوریتم جنگل تصادفی

جنگل تصادفی یک الگوریتم یادگیری نظارت شده محسوب می‌شود. همانطور که از نام آن مشهود است، این الگوریتم جنگلی را به طور تصادفی می‌سازد. «جنگل» ساخته شده، در واقع گروهی از درخت‌های تصمیم است. کار ساخت جنگل با استفاده از درخت‌ها اغلب اوقات به روش کیسه‌گذاری [۱]انجام می‌شود. ایده اصلی روش کیسه‌گذاری آن است که ترکیبی از مدل‌های یادگیری، نتایج کلی مدل را افزایش می‌دهد. به بیان ساده، جنگل تصادفی چندین درخت تصمیم ساخته و آن‌ها را با یکدیگر ادغام می‌کند تا پیش‌بینی‌های صحیح‌تر و پایدارتری حاصل شوند.

یکی از مزایای جنگل تصادفی قابل استفاده بودن آن، هم برای مسائل دسته‌بندی و هم رگرسیون است که غالب سیستم‌های یادگیری ماشین کنونی را تشکیل می‌دهند. در اینجا، عملکرد جنگل تصادفی برای انجام دسته‌بندی تشریح خواهد شد، زیرا گاهی دسته‌بندی را به عنوان بلوک سازنده یادگیری ماشین در نظر می‌گیرند. در تصویر زیر، می‌توان دو جنگل تصادفی ساخته شده از دو درخت را مشاهده کرد.

[۱] Bagging



شکل ۳-۳- الگوریتم جنگل تصادفی

جنگل تصادفی دارای فراپارامترهایی مشابه درخت تصمیم یا دسته‌بند کیسه‌گذاری [۱]است. خوشبختانه، نیازی به ترکیب یک درخت تصمیم با یک دسته‌بند کیسه‌گذاری نیست و می‌توان از کلاس دسته‌بندی[۲] جنگل تصادفی استفاده کرد. چنانکه پیش‌تر بیان شد، با جنگل تصادفی، و در واقع رگرسور جنگل تصادفی[۳] می‌توان به حل مسائل رگرسیون نیز پرداخت.

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[۴]، 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،

[۱] Bagging Classifier
[۲] Classifier-Class
[۳] Random Forest Regressor
[۴] Node

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).

۳-۸- الگوریتم K-نزدیکترین همسایه

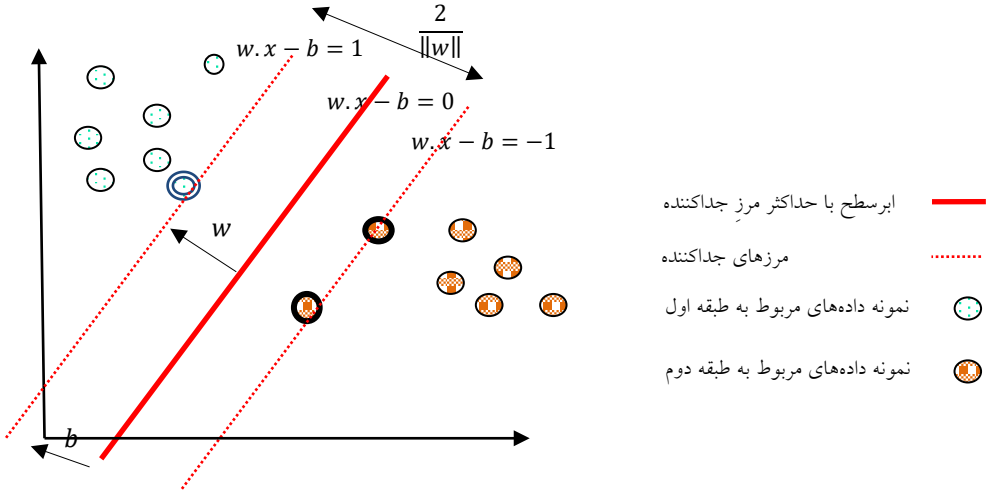
در پیش‌بینی الگو، الگوریتم K-نزدیکترین همسایه [۱]یک متد آمار ناپارامتری است که برای طبقه‌بندی آماری و رگرسیون استفاده می شود. در هر دو حالت کی شامل نزدیک ترین مثال آموزشی در فضای داده ای می باشد و خروجی آن بسته به نوع مورد استفاده در طبقه بندی و رگرسیون متغیر است. در حالت طبقه بندی با توجه به مقدار مشخص شده برای کی، به محاسبه فاصله نقطه ای که میخواهیم برچسب آن را مشخص کنیم با نزدیک ترین نقاط میپردازد و با توجه به تعداد رای حداکثری این نقاط همسایه، در رابطه با برچسب نقطه مورد نظر تصمیم گیری می‌کنیم. برای محاسبه این فاصله میتوان از روش های مختلفی استفاده کرد که یکی از مطرح ترین این روش ها، فاصله اقلیدسی است. در حالت رگرسیون نیز میانگین مقادیر بدست آمده از کی خروجی آن می باشد. از آنجا که محاسبات این الگوریتم بر اساس فاصله است نرمال‌سازی داده‌ها می‌تواند به بهبود عملکرد آن کمک کند.

[۱] k-nearest neighbors algorithm

۳-۹- الگوریتم ماشین بردار پشتیبان خطی

فرض کنیم مجموعه نقاط داده $\{(x_1.c_1).(x_2.c_2).\ldots.(x_n.c_n)\}$ را در اختیار داریم و می‌خواهیم آنها را به دو طبقه $c_i = \{-1.1\}$ تفکیک کنیم. هر x_i یک بردار p بعدی از اعداد حقیقی است که در واقع همان متغیرهای بیانگر رفتار کاربر در انتخاب فیلم می‌باشد

روشهای طبقه بندی خطی، سعی دارند که با ساختن یک ابرسطح (که عبارت است از یک معادله خطی)، داده‌ها را از هم تفکیک کنند. روش طبقه بندی ماشین بردار پشتیبان که یکی از روشهای طبقه‌بندی خطی است، بهترین ابرسطحی را پیدا می‌کند که با **حداکثر فاصله** (maximum margin)، داده‌های مربوط به دو طبقه را از هم تفکیک کند. به منظور درک بهتر مطلب، در شکل ۳ تصویری از یک مجموعه داده متعلق به دو کلاس نشان داده شده که روش ماشین بردار پشتیبان بهترین ابرسطح را برای جداسازی آنها انتخاب می کند..

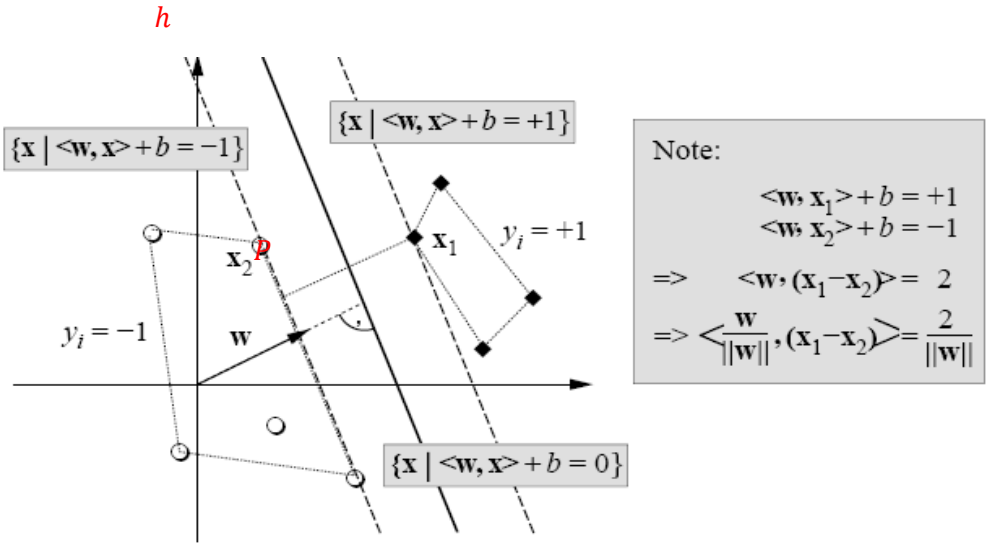


شکل ۳- ۴- ماشین بردار پشتیبان

در این شکل داده‌ها دو بعدی هستند یعنی هر داده تنها از دو متغیر تشکیل شده است.

۳-۱۰-نحوه تشکیل ابرسطح جداکننده توسط ماشین بردار پشتیبان

در این بخش می خواهیم نحوه ساخت ابرسطح جداکننده را بر روی یک مثال با جزئیات شرح دهیم. تصویر دقیقی از نحوه تشکیل ابرسطح جداکننده توسط ماشین بردار پشتیبان در شکل ۴ نشان داده شده است.



شکل ۳-۵-: نحوه ساخت ابرسطح جداکننده بین دو طبقه داده در فضای دو بعدی

ابتدا یک convex (پوسته محدب) در اطراف نقاط هر کدام از کلاسها در نظر بگیرید. در شکل ۲ در اطراف نقاط مربوط به کلاس -1 و نقاط مربوط به کلاس $+1$ پوسته محدب رسم شده است. خط P خطی است که نزدیکترین فاصله بین دو پوسته محدب را نشان می دهد. h که در واقع همان ابرسطح جداکننده است، خطی است که P را از وسط نصف کرده و بر آن عمود است.

b عرض از مبدا برای ابرسطح با حداکثر مرز جداکننده است. اگر b صرف‌نظر شود، پاسخ تنها ابرسطح‌هایی هستند که از مبدا می‌گذرند. فاصله عمودی ابرسطح تا مبدا با تقسیم قدرمطلق مقدار پارامترb برطول w بدست می‌آید.

ایده اصلی این است که یک جداکننده مناسب انتخاب شود. منظور، جداکننده‌ای است که بیشترین فاصله را با نقاط همسایه از هر دو طبقه دارد. این جواب درواقع بیشترین مرز را با نقاط مربوط به دو طبقه مختلف دارد و می‌تواند با دو ابرسطح موازی که حداقل از یکی از نقاط دو طبقه عبور می‌کنند، کران‌دار شود. این بردارها، ***بردارهای پشتیبان*** نام دارند. فرمول ریاضی این دو ابرسطح موازی که مرز جداکننده را تشکیل می‌دهند در عبارات(۱) و (۲) نشان داده شده است:

$$w.x - b = 1 \qquad (۱)$$

$$w.x - b = -1 \qquad (۲)$$

نکته قابل توجه این است که اگر داده‌های تعلیمی به صورت خطی تفکیک‌پذیر باشند، می‌توان دو ابرسطح مرزی را به گونه‌ای انتخاب کرد که هیچ داده‌ای بین آنها نباشد و سپس فاصله بین این دو ابرسطح موازی را به حداکثر رساند. با به کارگیری قضایای هندسی، فاصله این دو ابرسطح عبارت است از $2/|w|$ ، پس باید $|w|$ را به حداقل رساند. همچنین باید از قرار گرفتن نقاط داده در ناحیه درون مرز جلوگیری کرد، برای این کار یک محدودیت ریاضی به تعریف فرمال اضافه می‌شود. برای هر i ، با اعمال محدودیت‌های زیر اطمینان حاصل می‌شود که هیچ نقطه‌ای در مرز قرار نمی‌گیرد:

(۳) برای داده‌های مربوط به طبقه اول $w.x_i - b \geq 1$

(۴) برای داده‌های مربوط به طبقه دوم $w.x_i - b \leq -1$

می‌توان این محدودیت را به صورت رابطه زیر نشان داد:

$$c_i(w.x_i - b) \geq 1 \qquad , 1 \leq i \leq n \qquad (۵)$$

لذا مسأله بهینه‌سازی بدین شکل تعریف می‌شود:

به حداقل رساندن w، با درنظر گرفتن محدودیت زیر:

$$c_i(w.x_i - b) \geq 1 \qquad , 1 \leq i \leq n$$

۱۱-۳- جمع بندی

با توجه به مطالعات انجام شده در فصل دو و روش‌های ارایه شده، در این فصل ما با پیاده سازی الگوریتم‌های یادگیری ماشین KNN ، SVM، MLP، LR و RF به مقایسه‌ی روش‌های بعد از عمل تشنج پرداختیم. در نتیجه الگوریتم‌های پیاده سازی شده در فصل بعد به مقایسه خواهیم پرداخت و قابلیت‌های هر کدام در تست و آموزش انجام خواهد شد. در این کار ابتدا بیماران را به‌عنوان دیتا ست‌هایی از شاخص‌های قبل عمل و بعد عمل در نظر گرفتیم تا شباهت‌ها/تفاوت‌های بین پروفایل‌های بیمار آنها را شناسایی کنیم.

فصل چهارم: پیاده سازی و ارزیابی روش پیشنهادی

live code

```
!pip install numpy
!pip install pandas
!pip install seaborn
!pip install matplotlib
!pip install scikit-learn
Requirement already satisfied: numpy in c:\users\bagher\anaconda3\lib\site-packages (1.19.5)
Requirement already satisfied: pandas in c:\users\bagher\anaconda3\lib\site-packages (1.3.4)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\bagher\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\bagher\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in c:\users\bagher\anaconda3\lib\site-packages (from pandas) (1.19.5)
Requirement already satisfied: six>=1.5 in c:\users\bagher\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Requirement already satisfied: seaborn in c:\users\bagher\anaconda3\lib\site-packages (0.11.2)
Requirement already satisfied: matplotlib>=2.2 in c:\users\bagher\anaconda3\lib\site-packages (from seaborn) (3.4.3)
Requirement already satisfied: numpy>=1.15 in c:\users\bagher\anaconda3\lib\site-packages (from seaborn) (1.19.5)
Requirement already satisfied: pandas>=0.23 in c:\users\bagher\anaconda3\lib\site-packages (from seaborn) (1.3.4)
Requirement already satisfied: scipy>=1.0 in c:\users\bagher\anaconda3\lib\site-packages (from seaborn) (1.7.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (2.8.2)
```

Requirement already satisfied: pyparsing>=2.2.1 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (3.0.4)
Requirement already satisfied: pillow>=6.2.0 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (8.4.0)
Requirement already satisfied: six in c:\users\bagher\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib>=2.2->seaborn) (1.16.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\bagher\anaconda3\lib\site-packages (from pandas>=0.23->seaborn) (2021.3)
Requirement already satisfied: matplotlib in c:\users\bagher\anaconda3\lib\site-packages (3.4.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: numpy>=1.16 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib) (1.19.5)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\bagher\anaconda3\lib\site-packages (from matplotlib) (8.4.0)
Requirement already satisfied: six in c:\users\bagher\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib) (1.16.0)
Requirement already satisfied: scikit-learn in c:\users\bagher\anaconda3\lib\site-packages (0.24.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\bagher\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: numpy>=1.13.3 in c:\users\bagher\anaconda3\lib\site-packages (from scikit-learn) (1.19.5)
Requirement already satisfied: scipy>=0.19.1 in c:\users\bagher\anaconda3\lib\site-packages (from scikit-learn) (1.7.1)
Requirement already satisfied: joblib>=0.11 in c:\users\bagher\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
!pip install Pyppeteer

Collecting Pyppeteer
  Downloading pyppeteer-1.0.2-py3-none-any.whl (83 kB)
Requirement already satisfied: importlib-metadata>=1.4 in c:\users\bagher\anaconda3\lib\site-packages (from Pyppeteer) (4.8.1)
Requirement already satisfied: urllib3<2.0.0,>=1.25.8 in c:\users\bagher\anaconda3\lib\site-packages (from Pyppeteer) (1.26.7)
Requirement already satisfied: appdirs<2.0.0,>=1.4.3 in c:\users\bagher\anaconda3\lib\site-packages (from Pyppeteer) (1.4.4)
Requirement already satisfied: certifi>=2021 in c:\users\bagher\anaconda3\lib\site-packages (from Pyppeteer) (2021.10.8)
Collecting websockets<11.0,>=10.0
  Downloading websockets-10.1-cp39-cp39-win_amd64.whl (97 kB)
Collecting pyee<9.0.0,>=8.1.0
  Downloading pyee-8.2.2-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in c:\users\bagher\anaconda3\lib\site-packages (from Pyppeteer) (4.62.3)
Requirement already satisfied: zipp>=0.5 in c:\users\bagher\anaconda3\lib\site-packages (from importlib-metadata>=1.4->Pyppeteer) (3.6.0)
Requirement already satisfied: colorama in c:\users\bagher\anaconda3\lib\site-packages (from tqdm<5.0.0,>=4.42.1->Pyppeteer) (0.4.4)
Installing collected packages: websockets, pyee, Pyppeteer
Successfully installed Pyppeteer-1.0.2 pyee-8.2.2 websockets-10.1
!pip install tensorflow
Collecting tensorflow
  Using cached tensorflow-2.7.0-cp39-cp39-win_amd64.whl (430.8 MB)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\bagher\anaconda3\lib\site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\bagher\anaconda3\lib\site-packages (from tensorflow) (3.10.0.2)
Collecting keras-preprocessing>=1.1.1
  Using cached Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
Collecting libclang>=9.0.1
  Downloading libclang-13.0.0-py2.py3-none-win_amd64.whl (13.9 MB)

Requirement already satisfied: wheel<1.0,>=0.32.0 in c:\users\bagher\anaconda3\lib\site-packages (from tensorflow) (0.37.0)
Requirement already satisfied: protobuf>=3.9.2 in c:\users\bagher\anaconda3\lib\site-packages (from tensorflow) (3.19.1)
Requirement already satisfied: six>=1.12.0 in c:\users\bagher\anaconda3\lib\site-packages (from tensorflow) (1.16.0)
Collecting opt-einsum>=2.3.2
  Using cached opt_einsum-3.3.0-py3-none-any.whl (65 kB)
Collecting grpcio<2.0,>=1.24.3
  Downloading grpcio-1.43.0-cp39-cp39-win_amd64.whl (3.4 MB)
Requirement already satisfied: h5py>=2.9.0 in c:\users\bagher\anaconda3\lib\site-packages (from tensorflow) (3.2.1)
Collecting flatbuffers<3.0,>=1.12
  Using cached flatbuffers-2.0-py2.py3-none-any.whl (26 kB)
Collecting termcolor>=1.1.0
  Using cached termcolor-1.1.0-py3-none-any.whl
Collecting gast<0.5.0,>=0.2.1
  Using cached gast-0.4.0-py3-none-any.whl (9.8 kB)
Collecting tensorboard~=2.6
  Downloading tensorboard-2.8.0-py3-none-any.whl (5.8 MB)
Collecting tensorflow-estimator<2.8,~=2.7.0rc0
  Using cached tensorflow_estimator-2.7.0-py2.py3-none-any.whl (463 kB)
Requirement already satisfied: keras<2.8,>=2.7.0rc0 in c:\users\bagher\anaconda3\lib\site-packages (from tensorflow) (2.7.0)
Collecting absl-py>=0.4.0
  Using cached absl_py-1.0.0-py3-none-any.whl (126 kB)
Requirement already satisfied: numpy>=1.14.5 in c:\users\bagher\anaconda3\lib\site-packages (from tensorflow) (1.19.5)
Collecting google-pasta>=0.1.1
  Using cached google_pasta-0.2.0-py3-none-any.whl (57 kB)
Collecting tensorflow-io-gcs-filesystem>=0.21.0
  Using cached tensorflow_io_gcs_filesystem-0.23.1-cp39-cp39-win_amd64.whl (1.5 MB)
Collecting astunparse>=1.6.0

```
  Using cached astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\bagher\anaconda3\lib\site-
packages (from tensorboard~=2.6->tensorflow) (2.26.0)
Collecting google-auth-oauthlib<0.5,>=0.4.1
  Using cached google_auth_oauthlib-0.4.6-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: werkzeug>=0.11.15 in c:\users\bagher\anaconda3\lib\site-packages
(from tensorboard~=2.6->tensorflow) (2.0.2)
Requirement already satisfied: google-auth<3,>=1.6.3 in c:\users\bagher\anaconda3\lib\site-
packages (from tensorboard~=2.6->tensorflow) (2.3.3)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in
c:\users\bagher\anaconda3\lib\site-packages (from tensorboard~=2.6->tensorflow) (0.6.1)
Requirement already satisfied: setuptools>=41.0.0 in c:\users\bagher\anaconda3\lib\site-packages
(from tensorboard~=2.6->tensorflow) (58.0.4)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
c:\users\bagher\anaconda3\lib\site-packages (from tensorboard~=2.6->tensorflow) (1.8.0)
Collecting markdown>=2.6.8
  Using cached Markdown-3.3.6-py3-none-any.whl (97 kB)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\users\bagher\anaconda3\lib\site-
packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (4.2.4)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\users\bagher\anaconda3\lib\site-packages
(from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (4.8)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\bagher\anaconda3\lib\site-
packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\bagher\anaconda3\lib\site-
packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensorflow) (1.3.0)
Requirement already satisfied: importlib-metadata>=4.4 in c:\users\bagher\anaconda3\lib\site-
packages (from markdown>=2.6.8->tensorboard~=2.6->tensorflow) (4.8.1)
Requirement already satisfied: zipp>=0.5 in c:\users\bagher\anaconda3\lib\site-packages (from
importlib-metadata>=4.4->markdown>=2.6.8->tensorboard~=2.6->tensorflow) (3.6.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\bagher\anaconda3\lib\site-
packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard~=2.6-
>tensorflow) (0.4.8)
```

Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\bagher\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\bagher\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in c:\users\bagher\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (3.2)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\bagher\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (1.26.7)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\bagher\anaconda3\lib\site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensorflow) (3.1.1)
Installing collected packages: markdown, grpcio, google-auth-oauthlib, absl-py, termcolor, tensorflow-io-gcs-filesystem, tensorflow-estimator, tensorboard, opt-einsum, libclang, keras-preprocessing, google-pasta, gast, flatbuffers, astunparse, tensorflow
Successfully installed absl-py-1.0.0 astunparse-1.6.3 flatbuffers-2.0 gast-0.4.0 google-auth-oauthlib-0.4.6 google-pasta-0.2.0 grpcio-1.43.0 keras-preprocessing-1.1.2 libclang-13.0.0 markdown-3.3.6 opt-einsum-3.3.0 tensorboard-2.8.0 tensorflow-2.7.0 tensorflow-estimator-2.7.0 tensorflow-io-gcs-filesystem-0.23.1 termcolor-1.1.0

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
df = pd.read_csv("data.csv")

df.head()
```

AGE	GEN	EDU	HDS_b	EDH	HEM	HS_b	AHS	ASO	DUR	...	FRQ
	AED	SEV	SEV_b	AGE.1	DIST	ENG	ENG_b		NAM	NAM_b	

0	46	1	2	2	80	1	1	2.70	34	10	...	3
	4	12	2	33	1.78	1	1	-0.58	2			
1	48	1	3	2	100	1	1	1.21	27	21	...	2
	2	8	2	28	1.79	2	2	0.72	1			
2	42	1	3	2	100	1	1	1.34	6	31	...	2
	2	8	2	25	2.31	1	1	1.01	1			
3	23	1	3	2	90	1	2	5.98	3	20	...	2
	4	16	2	19	1.78	2	2	-2.05	1			
4	37	1	2	2	100	1	1	0.02	6	31	...	2
	3	12	2	30	2.24	2	2	-0.15	1			

5 rows × 21 columns

df.tail()

AGE	GEN AED	EDU SEV	HDS_b SEV_b	AGE.1	EDH DIST	HEM ENG	HS_b ENG_b	AHS	ASO NAM	DUR NAM_b	...	FRQ
42	41	2	2	2	100	2	2	22.01	24	17	...	1
	2	4	1	23	2.19	1	1	0.76	1			
43	30	2	2	2	90	2	2	18.98	19	17	...	2
	4	16	2	23	2.38	1	1	1.04	1			
44	45	2	3	2	100	2	2	17.89	39	13	...	2
	2	4	1	39	1.78	1	1	0.32	1			
45	35	2	3	2	100	2	2	14.61	3	28	...	2
	3	12	2	21	2.19	1	1	0.00	2			
46	30	2	3	2	90	2	2	32.55	19	20	...	4
	3	24	2	26	2.24	4	2	-0.70	2			

5 rows × 21 columns

```
print('Rows :',df.shape[0])
print('Columns :',df.shape[1])
```

Rows : 47

Columns : 21

df.describe()

AGE	GEN	EDU	HDS_b	EDH	HEM	HS_b	AHS	ASO	DUR	...	FRQ
	AED	SEV	SEV_b	AGE.1	DIST	ENG	ENG_b	NAM	NAM_b		
count	47.000000	47.000000	47.000000	47.000000	47.000000	47.000000	47.000000	47.000000	47.000000		
	47.000000	47.000000	47.000000	47.000000	...	47.000000					
	47.000000	47.000000	47.000000	47.000000	47.00000	47.000000					
	47.000000	47.000000	47.000000								
mean	39.297872	1.425532	2.234043	1.851064	64.680851	1.489362					
	1.553191	9.341702	15.829787	22.893617	...	2.319149					
	2.659574	10.638298	1.723404	27.319149	2.05766	1.510638					
	1.361702	-0.437021	1.404255								
std	9.853084	0.499769	0.666358	0.359875	55.983346	0.505291					
	0.502538	10.287381	10.076488	11.417443	...	0.694900					
	0.891423	5.798896	0.452151	8.663031	0.21922	0.804129					
	0.485688	1.104264	0.496053								
min	21.000000	1.000000	1.000000	1.000000	-100.000000	1.000000					
	1.000000	0.010000	3.000000	5.000000	...	1.000000					
	1.000000	2.000000	1.000000	18.000000	1.73000	1.000000					
	1.000000	-2.870000	1.000000								
25%	31.000000	1.000000	2.000000	2.000000	70.000000	1.000000					
	1.000000	0.805000	6.500000	12.500000	...	2.000000					
	2.000000	6.000000	1.000000	20.500000	1.87500	1.000000					
	1.000000	-1.225000	1.000000								
50%	39.000000	1.000000	2.000000	2.000000	80.000000	1.000000					
	2.000000	5.480000	13.000000	22.000000	...	2.000000					
	3.000000	9.000000	2.000000	25.000000	1.99000	1.000000					
	1.000000	-0.500000	1.000000								
75%	48.000000	2.000000	3.000000	2.000000	100.000000	2.000000					
	2.000000	17.390000	21.500000	32.000000	...	3.000000					
	3.000000	14.000000	2.000000	31.500000	2.22500	2.000000					
	2.000000	0.625000	2.000000								
max	63.000000	2.000000	3.000000	2.000000	100.000000	2.000000					
	2.000000	33.230000	46.000000	48.000000	...	4.000000					

4.000000	24.000000	2.000000	46.000000	2.48000	4.000000
2.000000	1.070000	2.000000			

8 rows × 21 columns

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47 entries, 0 to 46
Data columns (total 21 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   AGE     47 non-null     int64
 1   GEN     47 non-null     int64
 2   EDU     47 non-null     int64
 3   HDS_b   47 non-null     int64
 4   EDH     47 non-null     int64
 5   HEM     47 non-null     int64
 6   HS_b    47 non-null     int64
 7   AHS     47 non-null     float64
 8   ASO     47 non-null     int64
 9   DUR     47 non-null     int64
 10  DUR_b   47 non-null     int64
 11  FRQ     47 non-null     int64
 12  AED     47 non-null     int64
 13  SEV     47 non-null     int64
 14  SEV_b   47 non-null     int64
 15  AGE.1   47 non-null     int64
 16  DIST    47 non-null     float64
 17  ENG     47 non-null     int64
 18  ENG_b   47 non-null     int64
 19  NAM     47 non-null     float64
 20  NAM_b   47 non-null     int64
dtypes: float64(3), int64(18)
memory usage: 7.8 KB
```

```
df['AGE'].value_counts()
48    4
30    4
42    3
49    3
52    3
32    3
46    2
31    2
35    2
39    2
54    2
23    2
63    1
36    1
41    1
28    1
43    1
27    1
40    1
47    1
21    1
34    1
38    1
29    1
25    1
37    1
45    1
Name: AGE, dtype: int64
df['ENG'].value_counts()
1    30
2    12
3     3
```

```
4    2
Name: ENG, dtype: int64
df['NAM_b'].value_counts()
1    28
2    19
Name: NAM_b, dtype: int64
sns.pairplot(df,hue='ENG',size=2)
C:\Users\bagher\anaconda3\lib\site-packages\seaborn\axisgrid.py:2076: UserWarning: The `size`
parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
<seaborn.axisgrid.PairGrid at 0x2d1c43e43a0>

sns.heatmap(df.corr(), annot = True)
plt.show()

plt.subplot(1, 2, 1)
plt.scatter(x = df['NAM_b'], y = df['AGE'], color = 'red', marker = 'o',)
plt.title('NAM_b vs AGE')



plt.subplot(1, 2, 2)
plt.scatter(x = df['NAM'], y = df['AGE'], color = 'blue', marker = 'x',)
plt.title('NAM vs AGE')

plt.show()

features = list(df.columns)

print(features)
['AGE', 'GEN', 'EDU', 'HDS_b', 'EDH', 'HEM', 'HS_b', 'AHS', 'ASO', 'DUR', 'DUR_b', 'FRQ',
'AED', 'SEV', 'SEV_b', 'AGE.1', 'DIST', 'ENG', 'ENG_b', 'NAM', 'NAM_b']
features.remove('ENG')
features.remove('NAM_b')
```

```
features.remove('ENG_b')
features.remove('NAM')

print(features)
['AGE', 'GEN', 'EDU', 'HDS_b', 'EDH', 'HEM', 'HS_b', 'AHS', 'ASO', 'DUR', 'DUR_b', 'FRQ',
'AED', 'SEV', 'SEV_b', 'AGE.1', 'DIST']
Y = df.NAM_b
X = df[features].values.astype(np.float32)
X
array([[ 4.600e+01,  1.000e+00,  2.000e+00,  2.000e+00,  8.000e+01,
         1.000e+00,  1.000e+00,  2.700e+00,  3.400e+01,  1.000e+01,
         1.000e+00,  3.000e+00,  4.000e+00,  1.200e+01,  2.000e+00,
         3.300e+01,  1.780e+00],
       [ 4.800e+01,  1.000e+00,  3.000e+00,  2.000e+00,  1.000e+02,
         1.000e+00,  1.000e+00,  1.210e+00,  2.700e+01,  2.100e+01,
         2.000e+00,  2.000e+00,  2.000e+00,  8.000e+00,  2.000e+00,
         2.800e+01,  1.790e+00],
       [ 4.200e+01,  1.000e+00,  3.000e+00,  2.000e+00,  1.000e+02,
         1.000e+00,  1.000e+00,  1.340e+00,  6.000e+00,  3.100e+01,
         2.000e+00,  2.000e+00,  2.000e+00,  8.000e+00,  2.000e+00,
         2.500e+01,  2.310e+00],
       [ 2.300e+01,  1.000e+00,  3.000e+00,  2.000e+00,  9.000e+01,
         1.000e+00,  2.000e+00,  5.980e+00,  3.000e+00,  2.000e+01,
         2.000e+00,  2.000e+00,  4.000e+00,  1.600e+01,  2.000e+00,
         1.900e+01,  1.780e+00],
       [ 3.700e+01,  1.000e+00,  2.000e+00,  2.000e+00,  1.000e+02,
         1.000e+00,  1.000e+00,  2.000e-02,  6.000e+00,  3.100e+01,
         2.000e+00,  2.000e+00,  3.000e+00,  1.200e+01,  2.000e+00,
         3.000e+01,  2.240e+00],
       [ 4.900e+01,  1.000e+00,  2.000e+00,  2.000e+00,  9.000e+01,
         1.000e+00,  1.000e+00,  6.000e-02,  1.100e+01,  1.200e+01,
         1.000e+00,  2.000e+00,  3.000e+00,  6.000e+00,  1.000e+00,
         2.900e+01,  1.990e+00],
```

[5.200e+01, 1.000e+00, 2.000e+00, 2.000e+00, 8.000e+01,
  1.000e+00, 1.000e+00, 4.500e-01, 6.000e+00, 3.200e+01,
  2.000e+00, 2.000e+00, 2.000e+00, 8.000e+00, 2.000e+00,
  4.500e+01, 2.430e+00],
[2.300e+01, 2.000e+00, 2.000e+00, 2.000e+00, 7.000e+01,
  1.000e+00, 1.000e+00, 6.000e-01, 1.100e+01, 1.800e+01,
  2.000e+00, 2.000e+00, 3.000e+00, 1.200e+01, 2.000e+00,
  1.800e+01, 2.210e+00],
[3.200e+01, 2.000e+00, 3.000e+00, 2.000e+00, 1.000e+02,
  1.000e+00, 1.000e+00, 4.000e-01, 2.700e+01, 5.000e+00,
  1.000e+00, 3.000e+00, 2.000e+00, 6.000e+00, 1.000e+00,
  1.900e+01, 2.440e+00],
[3.000e+01, 2.000e+00, 1.000e+00, 2.000e+00, 8.000e+01,
  1.000e+00, 1.000e+00, 1.010e+00, 8.000e+00, 1.700e+01,
  2.000e+00, 3.000e+00, 3.000e+00, 1.800e+01, 2.000e+00,
  2.400e+01, 1.960e+00],
[2.500e+01, 1.000e+00, 3.000e+00, 1.000e+00, -4.000e+01,
  1.000e+00, 2.000e+00, 5.670e+00, 1.500e+01, 3.700e+01,
  2.000e+00, 2.000e+00, 2.000e+00, 8.000e+00, 2.000e+00,
  2.000e+01, 2.350e+00],
[2.900e+01, 1.000e+00, 1.000e+00, 1.000e+00, -6.000e+01,
  1.000e+00, 2.000e+00, 3.221e+01, 2.300e+01, 9.000e+00,
  1.000e+00, 3.000e+00, 4.000e+00, 1.200e+01, 2.000e+00,
  1.800e+01, 1.850e+00],
[3.100e+01, 2.000e+00, 2.000e+00, 1.000e+00, -6.000e+01,
  1.000e+00, 2.000e+00, 1.712e+01, 2.300e+01, 8.000e+00,
  1.000e+00, 2.000e+00, 3.000e+00, 6.000e+00, 1.000e+00,
  2.000e+01, 2.270e+00],
[3.800e+01, 2.000e+00, 2.000e+00, 1.000e+00, -6.000e+01,
  1.000e+00, 2.000e+00, 1.748e+01, 1.000e+01, 2.500e+01,
  2.000e+00, 2.000e+00, 2.000e+00, 8.000e+00, 2.000e+00,
  3.200e+01, 2.180e+00],
[4.800e+01, 1.000e+00, 2.000e+00, 2.000e+00, 7.000e+01,

1.000e+00, 2.000e+00, 1.944e+01, 1.300e+01, 3.500e+01,
2.000e+00, 2.000e+00, 2.000e+00, 8.000e+00, 2.000e+00,
4.400e+01, 1.820e+00],
[4.000e+01, 1.000e+00, 1.000e+00, 2.000e+00, 8.000e+01,
1.000e+00, 2.000e+00, 3.323e+01, 1.800e+01, 2.200e+01,
2.000e+00, 3.000e+00, 4.000e+00, 2.400e+01, 2.000e+00,
3.500e+01, 2.060e+00],
[3.400e+01, 1.000e+00, 3.000e+00, 2.000e+00, 1.000e+02,
1.000e+00, 2.000e+00, 6.440e+00, 6.000e+00, 2.400e+01,
2.000e+00, 2.000e+00, 2.000e+00, 8.000e+00, 2.000e+00,
2.700e+01, 1.970e+00],
[2.100e+01, 1.000e+00, 3.000e+00, 2.000e+00, 6.000e+01,
1.000e+00, 2.000e+00, 4.980e+00, 1.200e+01, 1.000e+01,
1.000e+00, 3.000e+00, 3.000e+00, 9.000e+00, 2.000e+00,
1.800e+01, 1.730e+00],
[4.800e+01, 1.000e+00, 2.000e+00, 2.000e+00, 7.000e+01,
1.000e+00, 2.000e+00, 1.242e+01, 1.300e+01, 3.600e+01,
2.000e+00, 2.000e+00, 3.000e+00, 1.200e+01, 2.000e+00,
2.000e+01, 2.480e+00],
[3.200e+01, 1.000e+00, 2.000e+00, 2.000e+00, 8.000e+01,
1.000e+00, 2.000e+00, 1.730e+01, 2.000e+01, 1.200e+01,
1.000e+00, 2.000e+00, 4.000e+00, 8.000e+00, 2.000e+00,
2.700e+01, 1.870e+00],
[4.700e+01, 2.000e+00, 2.000e+00, 2.000e+00, 6.000e+01,
1.000e+00, 2.000e+00, 1.562e+01, 1.200e+01, 3.500e+01,
2.000e+00, 3.000e+00, 3.000e+00, 1.800e+01, 2.000e+00,
4.300e+01, 1.730e+00],
[5.400e+01, 2.000e+00, 2.000e+00, 2.000e+00, 9.000e+01,
1.000e+00, 2.000e+00, 1.278e+01, 1.800e+01, 3.600e+01,
2.000e+00, 4.000e+00, 1.000e+00, 8.000e+00, 2.000e+00,
4.400e+01, 1.890e+00],
[4.900e+01, 2.000e+00, 3.000e+00, 2.000e+00, 7.000e+01,
1.000e+00, 2.000e+00, 5.790e+00, 5.000e+00, 4.400e+01,

2.000e+00, 2.000e+00, 2.000e+00, 8.000e+00, 2.000e+00,
2.100e+01, 1.890e+00],
[5.400e+01, 2.000e+00, 3.000e+00, 2.000e+00, 7.000e+01,
1.000e+00, 1.000e+00, 1.200e+00, 6.000e+00, 4.800e+01,
2.000e+00, 2.000e+00, 4.000e+00, 1.600e+01, 2.000e+00,
3.300e+01, 1.980e+00],
[3.600e+01, 1.000e+00, 3.000e+00, 1.000e+00, -1.000e+02,
2.000e+00, 1.000e+00, 6.300e-01, 3.200e+01, 1.300e+01,
1.000e+00, 3.000e+00, 2.000e+00, 6.000e+00, 1.000e+00,
2.100e+01, 1.900e+00],
[4.300e+01, 1.000e+00, 2.000e+00, 1.000e+00, -4.000e+01,
2.000e+00, 1.000e+00, 1.900e-01, 6.000e+00, 3.600e+01,
2.000e+00, 2.000e+00, 3.000e+00, 1.200e+01, 2.000e+00,
2.800e+01, 1.820e+00],
[4.900e+01, 2.000e+00, 2.000e+00, 1.000e+00, -8.000e+01,
2.000e+00, 1.000e+00, 1.400e-01, 1.300e+01, 3.600e+01,
2.000e+00, 2.000e+00, 1.000e+00, 4.000e+00, 1.000e+00,
4.100e+01, 1.880e+00],
[5.200e+01, 1.000e+00, 2.000e+00, 2.000e+00, 1.000e+02,
2.000e+00, 1.000e+00, 2.450e+00, 4.600e+01, 5.000e+00,
1.000e+00, 2.000e+00, 2.000e+00, 4.000e+00, 1.000e+00,
4.600e+01, 2.350e+00],
[2.800e+01, 1.000e+00, 2.000e+00, 2.000e+00, 5.000e+01,
2.000e+00, 1.000e+00, 1.800e+00, 1.800e+01, 1.000e+01,
1.000e+00, 3.000e+00, 1.000e+00, 3.000e+00, 1.000e+00,
2.300e+01, 1.840e+00],
[3.100e+01, 1.000e+00, 1.000e+00, 2.000e+00, 8.000e+01,
2.000e+00, 1.000e+00, 3.000e-02, 1.000e+01, 2.000e+01,
2.000e+00, 3.000e+00, 2.000e+00, 1.200e+01, 2.000e+00,
2.500e+01, 2.350e+00],
[4.200e+01, 1.000e+00, 2.000e+00, 2.000e+00, 8.000e+01,
2.000e+00, 1.000e+00, 7.000e-02, 2.400e+01, 6.000e+00,
1.000e+00, 1.000e+00, 2.000e+00, 2.000e+00, 1.000e+00,

1.800e+01, 2.180e+00],
[3.500e+01, 2.000e+00, 2.000e+00, 2.000e+00, 1.000e+02,
2.000e+00, 1.000e+00, 1.000e-02, 6.000e+00, 2.500e+01,
2.000e+00, 3.000e+00, 3.000e+00, 1.800e+01, 2.000e+00,
1.900e+01, 2.080e+00],
[4.200e+01, 2.000e+00, 1.000e+00, 2.000e+00, 9.000e+01,
2.000e+00, 1.000e+00, 7.000e-02, 5.000e+00, 3.600e+01,
2.000e+00, 3.000e+00, 3.000e+00, 1.800e+01, 2.000e+00,
3.100e+01, 2.210e+00],
[5.200e+01, 2.000e+00, 2.000e+00, 2.000e+00, 1.000e+02,
2.000e+00, 1.000e+00, 1.200e+00, 1.700e+01, 2.600e+01,
2.000e+00, 3.000e+00, 2.000e+00, 1.200e+01, 2.000e+00,
2.600e+01, 2.400e+00],
[3.900e+01, 2.000e+00, 3.000e+00, 2.000e+00, 1.000e+02,
2.000e+00, 1.000e+00, 9.800e-01, 1.400e+01, 2.100e+01,
2.000e+00, 1.000e+00, 2.000e+00, 4.000e+00, 1.000e+00,
2.200e+01, 2.060e+00],
[6.300e+01, 1.000e+00, 3.000e+00, 2.000e+00, 8.000e+01,
2.000e+00, 2.000e+00, 6.790e+00, 3.300e+01, 3.000e+01,
2.000e+00, 2.000e+00, 4.000e+00, 1.600e+01, 2.000e+00,
4.500e+01, 1.850e+00],
[4.800e+01, 1.000e+00, 3.000e+00, 2.000e+00, 1.000e+02,
2.000e+00, 2.000e+00, 5.480e+00, 2.000e+01, 2.800e+01,
2.000e+00, 2.000e+00, 4.000e+00, 1.600e+01, 2.000e+00,
2.200e+01, 2.060e+00],
[2.700e+01, 1.000e+00, 3.000e+00, 2.000e+00, 1.000e+02,
2.000e+00, 2.000e+00, 4.100e+00, 1.800e+01, 9.000e+00,
1.000e+00, 1.000e+00, 2.000e+00, 2.000e+00, 1.000e+00,
1.800e+01, 1.990e+00],
[3.200e+01, 1.000e+00, 2.000e+00, 2.000e+00, 1.000e+02,
2.000e+00, 2.000e+00, 1.902e+01, 6.000e+00, 2.600e+01,
2.000e+00, 2.000e+00, 3.000e+00, 1.200e+01, 2.000e+00,
1.900e+01, 1.970e+00],

```
       [ 3.900e+01,  1.000e+00,  1.000e+00,  2.000e+00,  1.000e+02,
         2.000e+00,  2.000e+00,  2.954e+01,  8.000e+00,  3.800e+01,
         2.000e+00,  3.000e+00,  4.000e+00,  2.400e+01,  2.000e+00,
         2.500e+01,  1.930e+00],
       [ 4.600e+01,  1.000e+00,  2.000e+00,  2.000e+00,  1.000e+02,
         2.000e+00,  2.000e+00,  2.540e+01,  7.000e+00,  3.200e+01,
         2.000e+00,  3.000e+00,  2.000e+00,  1.200e+01,  2.000e+00,
         3.000e+01,  1.950e+00],
       [ 3.000e+01,  2.000e+00,  2.000e+00,  2.000e+00,  8.000e+01,
         2.000e+00,  2.000e+00,  1.967e+01,  2.400e+01,  6.000e+00,
         1.000e+00,  2.000e+00,  2.000e+00,  4.000e+00,  1.000e+00,
         2.100e+01,  2.110e+00],
       [ 4.100e+01,  2.000e+00,  2.000e+00,  2.000e+00,  1.000e+02,
         2.000e+00,  2.000e+00,  2.201e+01,  2.400e+01,  1.700e+01,
         2.000e+00,  1.000e+00,  2.000e+00,  4.000e+00,  1.000e+00,
         2.300e+01,  2.190e+00],
       [ 3.000e+01,  2.000e+00,  2.000e+00,  2.000e+00,  9.000e+01,
         2.000e+00,  2.000e+00,  1.898e+01,  1.900e+01,  1.700e+01,
         2.000e+00,  2.000e+00,  4.000e+00,  1.600e+01,  2.000e+00,
         2.300e+01,  2.380e+00],
       [ 4.500e+01,  2.000e+00,  3.000e+00,  2.000e+00,  1.000e+02,
         2.000e+00,  2.000e+00,  1.789e+01,  3.900e+01,  1.300e+01,
         1.000e+00,  2.000e+00,  2.000e+00,  4.000e+00,  1.000e+00,
         3.900e+01,  1.780e+00],
       [ 3.500e+01,  2.000e+00,  3.000e+00,  2.000e+00,  1.000e+02,
         2.000e+00,  2.000e+00,  1.461e+01,  3.000e+00,  2.800e+01,
         2.000e+00,  2.000e+00,  3.000e+00,  1.200e+01,  2.000e+00,
         2.100e+01,  2.190e+00],
       [ 3.000e+01,  2.000e+00,  3.000e+00,  2.000e+00,  9.000e+01,
         2.000e+00,  2.000e+00,  3.255e+01,  1.900e+01,  2.000e+01,
         2.000e+00,  4.000e+00,  3.000e+00,  2.400e+01,  2.000e+00,
         2.600e+01,  2.240e+00]], dtype=float32)
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 4)

print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
(37, 17)
(10, 17)
(37,)
(10,)
y_test
32    1
45    2
4     1
15    2
38    1
20    1
16    1
7     2
18    2
11    1
Name: NAM_b, dtype: int64
# standardization

#from sklearn.preprocessing import MinMaxScaler

#mm = MinMaxScaler()

# feeding the data to the scaler
#x_train = mm.fit_transform(x_train)
#x_test = mm.transform(x_test)
```

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

# feeding the into the scaler

x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
x_train
array([[ 0.88482505,  1.145644  ,  1.0942203 ,  0.43994135,  0.1220886 ,
        -1.0846523 ,  0.97332853, -0.2591923 , -1.1982504 ,  1.8714821 ,
         0.7359801 , -0.40712827, -0.5887448 , -0.30231085,  0.7359801 ,
        -0.76299185, -0.7261282 ],
       [-1.2780806 , -0.8728716 ,  1.0942203 ,  0.43994135,  0.643313  ,
         0.92195445,  0.97332853, -0.43810052,  0.06569355, -1.0949429 ,
        -1.3587325 , -1.7765596 , -0.5887448 , -1.3509516 , -1.3587325 ,
        -1.1086739 , -0.26843235],
       [-0.88482505,  1.145644  , -0.5252257 , -2.2730303 , -2.1365507 ,
        -1.0846523 ,  0.97332853,  0.94023377,  0.5518258 , -1.179698  ,
        -1.3587325 , -0.40712827,  0.5004331 , -0.65185773, -1.3587325 ,
        -0.8782192 ,  1.0131158 ],
       [ 0.09831389,  1.145644  , -0.5252257 ,  0.43994135,  0.643313  ,
         0.92195445,  0.97332853,  1.457903  ,  0.6490523 , -0.416903  ,
         0.7359801 , -1.7765596 , -0.5887448 , -1.0014046 , -1.3587325 ,
        -0.53253716,  0.6469595 ],
       [ 0.88482505,  1.145644  , -0.5252257 , -2.2730303 , -2.4840336 ,
         0.92195445, -1.0274023 , -0.8573173 , -0.4204387 ,  1.1934421 ,
         0.7359801 , -0.40712827, -1.6779227 , -1.0014046 , -1.3587325 ,
         1.5415549 , -0.7718978 ],
       [ 0.2949417 , -0.8728716 , -0.5252257 , -2.2730303 , -1.7890676 ,
         0.92195445, -1.0274023 , -0.85202414, -1.1010239 ,  1.1934421 ,
         0.7359801 , -0.40712827,  0.5004331 ,  0.396783  ,  0.7359801 ,
         0.04359953, -1.046515  ],
```

[0.49156946, 1.145644 , 1.0942203 , 0.43994135, 0.643313 ,
  0.92195445, 0.97332853, 1.021748 , 2.107449 , -0.755923 ,
 -1.3587325 , -0.40712827, -0.5887448 , -1.0014046 , -1.3587325 ,
  1.3111002 , -1.2295938],
[-0.19662778, 1.145644 , -0.5252257 , -2.2730303 , -2.1365507 ,
 -1.0846523 , 0.97332853, 0.97834426, -0.7121181 , 0.26113704,
  0.7359801 , -0.40712827, -0.5887448 , -0.30231085, 0.7359801 ,
  0.5045089 , 0.60119   ],
[-1.4747084 , -0.8728716 , 1.0942203 , -2.2730303 , -1.7890676 ,
 -1.0846523 , 0.97332853, -0.2718958 , -0.22598581, 1.278197 ,
  0.7359801 , -0.40712827, -0.5887448 , -0.30231085, 0.7359801 ,
 -0.8782192 , 1.379272],
[-0.7865111 , -0.8728716 , -0.5252257 , 0.43994135, 0.29583007,
 -1.0846523 , 0.97332853, 0.9592889 , 0.26014647, -0.840678 ,
 -1.3587325 , -0.40712827, 1.5896109 , -0.30231085, 0.7359801 ,
 -0.0716278 , -0.8176673],
[-1.1797668 , -0.8728716 , -0.5252257 , 0.43994135, -0.22539434,
  0.92195445, -1.0274023 , -0.681585 , 0.06569355, -1.010188 ,
 -1.3587325 , 0.96230316, -1.6779227 , -1.1761781 , -1.3587325 ,
 -0.53253716, -0.9549759],
[2.2612195 , -0.8728716 , 1.0942203 , 0.43994135, 0.29583007,
  0.92195445, 0.97332853, -0.15332945, 1.5240903 , 0.6849121 ,
  0.7359801 , -0.40712827, 1.5896109 , 1.0958768 , 0.7359801 ,
  2.0024643 , -0.9092064],
[1.1797668 , -0.8728716 , -0.5252257 , 0.43994135, 0.643313 ,
  0.92195445, -1.0274023 , -0.61277413, 2.7880344 , -1.433963 ,
 -1.3587325 , -0.40712827, -0.5887448 , -1.0014046 , -1.3587325 ,
  2.1176918 , 1.379272],
[-1.867964 , -0.8728716 , 1.0942203 , 0.43994135, -0.05165287,
 -1.0846523 , 0.97332853, -0.34494117, -0.51766515, -1.010188 ,
 -1.3587325 , 0.96230316, 0.5004331 , -0.12753738, 0.7359801 ,
 -1.1086739 , -1.4584415],
[-0.9831389 , 1.145644 , -0.5252257 , 0.43994135, 0.29583007,

0.92195445, 0.97332853, 1.2101839 , 0.6490523 , -1.349208 ,
-1.3587325 , -0.40712827, -0.5887448 , -1.0014046 , -1.3587325 ,
-0.76299185, 0.28080207],
[1.1797668 , 1.145644 , -0.5252257 , 0.43994135, 0.643313 ,
 0.92195445, -1.0274023 , -0.7451027 , -0.0315329 , 0.345892 ,
 0.7359801 , 0.96230316, -0.5887448 , 0.396783 , 0.7359801 ,
-0.18685515, 1.6081208],
[0.7865111 , -0.8728716 , -0.5252257 , 0.43994135, 0.1220886 ,
-1.0846523 , 0.97332853, 1.1858355 , -0.4204387 , 1.108687 ,
 0.7359801 , -0.40712827, -0.5887448 , -0.30231085, 0.7359801 ,
 1.887237 , -1.046515],
[0.19662778, -0.8728716 , 1.0942203 , 0.43994135, 0.643313 ,
-1.0846523 , -1.0274023 , -0.7302819 , -1.1010239 , 0.7696671 ,
 0.7359801 , -0.40712827, -0.5887448 , -0.30231085, 0.7359801 ,
-0.30208248, 1.1961939],
[-0.39325556, -0.8728716 , 1.0942203 , -2.2730303 , -2.8315165 ,
 0.92195445, -1.0274023 , -0.80544454, 1.4268639 , -0.755923 ,
-1.3587325 , 0.96230316, -0.5887448 , -0.65185773, -1.3587325 ,
-0.76299185, -0.6803587],
[-0.49156946, 1.145644 , -0.5252257 , 0.43994135, 0.643313 ,
 0.92195445, -1.0274023 , -0.8710795 , -1.1010239 , 0.26113704,
 0.7359801 , 0.96230316, 0.5004331 , 1.4454237 , 0.7359801 ,
-0.9934466 , 0.14349347],
[-0.88482505, -0.8728716 , -2.1446717 , 0.43994135, 0.29583007,
 0.92195445, -1.0274023 , -0.8689623 , -0.7121181 , -0.16263798,
 0.7359801 , 0.96230316, -0.5887448 , 0.396783 , 0.7359801 ,
-0.30208248, 1.379272],
[1.1797668 , -0.8728716 , -0.5252257 , 0.43994135, 0.29583007,
-1.0846523 , -1.0274023 , -0.82449985, -1.1010239 , 0.8544221 ,
 0.7359801 , -0.40712827, -0.5887448 , -0.30231085, 0.7359801 ,
 2.0024643 , 1.7454294],
[-0.9831389 , 1.145644 , -0.5252257 , 0.43994135, 0.46957156,
 0.92195445, 0.97332853, 1.1371385 , 0.16292   , -0.416903 ,

0.7359801 , -0.40712827, 1.5896109 , 1.0958768 , 0.7359801 ,
-0.53253716, 1.5165818],
[-0.09831389, 1.145644 , 1.0942203 , 0.43994135, 0.643313 ,
0.92195445, -1.0274023 , -0.76839256, -0.32321227, -0.07788298,
0.7359801 , -1.7765596 , -0.5887448 , -1.0014046 , -1.3587325 ,
-0.6477645 , 0.0519544],
[1.3763945 , 1.145644 , -0.5252257 , 0.43994135, 0.46957156,
-1.0846523 , 0.97332853, 0.48078892, 0.06569355, 1.1934421 ,
0.7359801 , 2.3317347 , -1.6779227 , -0.30231085, 0.7359801 ,
1.887237 , -0.7261282],
[0.5898834 , -0.8728716 , -0.5252257 , 0.43994135, 0.29583007,
-1.0846523 , -1.0274023 , -0.5863084 , 1.6213169 , -1.010188 ,
-1.3587325 , 0.96230316, 1.5896109 , 0.396783 , 0.7359801 ,
0.61973625, -1.2295938],
[-1.6713362 , -0.8728716 , 1.0942203 , 0.43994135, 0.46957156,
-1.0846523 , 0.97332853, -0.23907833, -1.3927033 , -0.16263798,
0.7359801 , -0.40712827, 1.5896109 , 1.0958768 , 0.7359801 ,
-0.9934466 , -1.2295938],
[0.7865111 , -0.8728716 , 1.0942203 , 0.43994135, 0.643313 ,
0.92195445, 0.97332853, -0.29200974, 0.26014647, 0.515402 ,
0.7359801 , -0.40712827, 1.5896109 , 1.0958768 , 0.7359801 ,
-0.6477645 , 0.0519544],
[0.19662778, -0.8728716 , -0.5252257 , 0.43994135, 0.29583007,
0.92195445, -1.0274023 , -0.8647277 , 0.6490523 , -1.349208 ,
-1.3587325 , -1.7765596 , -0.5887448 , -1.3509516 , -1.3587325 ,
-1.1086739 , 0.60119],
[-0.09831389, -0.8728716 , -2.1446717 , 0.43994135, 0.643313 ,
0.92195445, 0.97332853, 2.2550502 , -0.90657103, 1.3629521 ,
0.7359801 , 0.96230316, 1.5896109 , 2.4940646 , 0.7359801 ,
-0.30208248, -0.5430501],
[-0.9831389 , 1.145644 , -2.1446717 , 0.43994135, 0.29583007,
-1.0846523 , -1.0274023 , -0.76521665, -0.90657103, -0.416903 ,
0.7359801 , 0.96230316, 0.5004331 , 1.4454237 , 0.7359801 ,

```
 -0.41730982, -0.40574098],
[-0.7865111 ,  1.145644  ,  1.0942203 ,  0.43994135,  0.643313  ,
 -1.0846523 , -1.0274023 , -0.829793  ,  0.9407316 , -1.433963  ,
 -1.3587325 ,  0.96230316, -0.5887448 , -0.65185773, -1.3587325 ,
 -0.9934466 ,  1.791199  ],
[ 1.3763945 ,  1.145644  ,  1.0942203 ,  0.43994135,  0.1220886 ,
 -1.0846523 , -1.0274023 , -0.7451027 , -1.1010239 ,  2.2105021 ,
  0.7359801 , -0.40712827,  1.5896109 ,  1.0958768 ,  0.7359801 ,
  0.61973625, -0.3142019 ],
[ 0.5898834 , -0.8728716 , -0.5252257 ,  0.43994135,  0.643313  ,
  0.92195445,  0.97332853,  1.816778  , -1.0037975 ,  0.8544221 ,
  0.7359801 ,  0.96230316, -0.5887448 ,  0.396783  ,  0.7359801 ,
  0.27405423, -0.45151052],
[ 0.7865111 , -0.8728716 ,  1.0942203 ,  0.43994135,  0.643313  ,
 -1.0846523 , -1.0274023 , -0.7440441 ,  0.9407316 , -0.07788298,
  0.7359801 , -0.40712827, -0.5887448 , -0.30231085,  0.7359801 ,
  0.04359953, -1.1838242 ],
[ 0.88482505, -0.8728716 , -0.5252257 ,  0.43994135,  0.46957156,
 -1.0846523 , -1.0274023 , -0.8657863 , -0.6148916 , -0.840678  ,
 -1.3587325 , -0.40712827,  0.5004331 , -0.65185773, -1.3587325 ,
  0.15882687, -0.26843235],
[-0.9831389 ,  1.145644  ,  1.0942203 ,  0.43994135,  0.46957156,
  0.92195445,  0.97332853,  2.573697  ,  0.16292   , -0.16263798,
  0.7359801 ,  2.3317347 ,  0.5004331 ,  2.4940646 ,  0.7359801 ,
 -0.18685515,  0.87580717]], dtype=float32)
```

Using Machine Learning to solve the Epilspy Dataset : Support Vector Classifier

```
model = SVC()

model.fit(x_train,y_train)

y_pred = model.predict(x_test)

print("training accuracy :", model.score(x_train, y_train))
```

```
print("testing accuracy :", model.score(x_test, y_test))

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
print(cm)
training accuracy : 0.8918918918918919
testing accuracy : 0.5
[[5 1]
 [4 0]]
Decision Tree
model = DecisionTreeClassifier()

model.fit(x_train, y_train)

y_pred = model.predict(x_test)

print("training accuracy :", model.score(x_train, y_train))
print("testing accuracy :", model.score(x_test ,y_test))

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
print(cm)
training accuracy : 1.0
testing accuracy : 0.8
[[5 1]
 [1 3]]
Random Forest
model = RandomForestClassifier()

model.fit(x_train, y_train)
```

```
y_pred = model.predict(x_test)

print("training accuracy :", model.score(x_train, y_train))
print("testing accuracy :", model.score(x_test, y_test))

cm = confusion_matrix(y_test, y_pred)
print(cm)
training accuracy : 1.0
testing accuracy : 0.7
[[5 1]
 [2 2]]
from sklearn.neural_network import MLPClassifier
import tensorflow as tf
import seaborn as sns
Multi Layer Perceptron
model = MLPClassifier(hidden_layer_sizes = (5,2), max_iter = 150)

model.fit(x_train, y_train)
y_pred = model.predict(x_test)

print("training accuracy :", model.score(x_train, y_train))
print("testing accuracy :", model.score(x_test, y_test))

cm = confusion_matrix(y_test, y_pred)
print(cm)
training accuracy : 1.0
testing accuracy : 0.5
[[5 1]
 [4 0]]
from sklearn.linear_model import Perceptron

model = Perceptron()
model.fit(x_train, y_train)
```

```
y_pred = model.predict(x_test)

print("Misclassified Samples : %d" %(y_test != y_pred).sum())

print("Training Accuracy :", model.score(x_train, y_train))
print("Testing Accuracy :", model.score(x_test, y_test))

cm = confusion_matrix(y_pred, y_test)
print("Confusion Matrix :",cm)
Misclassified Samples : 5
Training Accuracy : 0.7837837837837838
Testing Accuracy : 0.5
Confusion Matrix : [[3 2]
 [3 2]]
#create object of StandardScaler class
X_train =x_train
X_test=x_test

ss=StandardScaler()
X_train=ss.fit_transform(X_train)
X_test=ss.transform(X_test)
#Create a neural network
#create the object of Sequencial model
model=tf.keras.Sequential([
        tf.keras.layers.Dense(2,activation='relu',input_shape=(X.shape[1],)),
        tf.keras.layers.Dense(3,activation='relu'),
        tf.keras.layers.Dense(3,activation='softmax') #output layer unit of neuron=3
])
#compile the model
#optimizer='adam' if use multiclass in classification algo. and loss='sparse_crossentropy
model.compile(optimizer="adam", loss="sparse_categorical_crossentropy")
Y_train=y_train
```

```
#train the model , use fit() method
trained_model=model.fit(X_train,Y_train,epochs=30, batch_size=10)
Epoch 1/30
4/4 [==============================] - 0s 1ms/step - loss: 1.1595
Epoch 2/30
4/4 [==============================] - 0s 2ms/step - loss: 1.1512
Epoch 3/30
4/4 [==============================] - 0s 2ms/step - loss: 1.1424
Epoch 4/30
4/4 [==============================] - 0s 2ms/step - loss: 1.1350
Epoch 5/30
4/4 [==============================] - 0s 2ms/step - loss: 1.1283
Epoch 6/30
4/4 [==============================] - 0s 2ms/step - loss: 1.1205
Epoch 7/30
4/4 [==============================] - 0s 2ms/step - loss: 1.1145
Epoch 8/30
4/4 [==============================] - 0s 2ms/step - loss: 1.1067
Epoch 9/30
4/4 [==============================] - 0s 2ms/step - loss: 1.1011
Epoch 10/30
4/4 [==============================] - 0s 2ms/step - loss: 1.0938
Epoch 11/30
4/4 [==============================] - 0s 2ms/step - loss: 1.0887
Epoch 12/30
4/4 [==============================] - 0s 2ms/step - loss: 1.0821
Epoch 13/30
4/4 [==============================] - 0s 2ms/step - loss: 1.0765
Epoch 14/30
4/4 [==============================] - 0s 1ms/step - loss: 1.0709
Epoch 15/30
4/4 [==============================] - 0s 2ms/step - loss: 1.0660
Epoch 16/30
```

4/4 [==============================] - 0s 2ms/step - loss: 1.0605

Epoch 17/30

4/4 [==============================] - 0s 2ms/step - loss: 1.0557

Epoch 18/30

4/4 [==============================] - 0s 1ms/step - loss: 1.0504

Epoch 19/30

4/4 [==============================] - 0s 2ms/step - loss: 1.0456

Epoch 20/30

4/4 [==============================] - 0s 1ms/step - loss: 1.0420

Epoch 21/30

4/4 [==============================] - 0s 2ms/step - loss: 1.0361

Epoch 22/30

4/4 [==============================] - 0s 2ms/step - loss: 1.0322

Epoch 23/30

4/4 [==============================] - 0s 2ms/step - loss: 1.0275

Epoch 24/30

4/4 [==============================] - 0s 2ms/step - loss: 1.0236

Epoch 25/30

4/4 [==============================] - 0s 2ms/step - loss: 1.0194

Epoch 26/30

4/4 [==============================] - 0s 1ms/step - loss: 1.0150

Epoch 27/30

4/4 [==============================] - 0s 2ms/step - loss: 1.0111

Epoch 28/30

4/4 [==============================] - 0s 1ms/step - loss: 1.0070

Epoch 29/30

4/4 [==============================] - 0s 2ms/step - loss: 1.0033

Epoch 30/30

4/4 [==============================] - 0s 2ms/step - loss: 0.9992

#testing the model

Y_pred = model.predict(X_test)

Y_pred

array([[0.2615192 , 0.31009597, 0.4283849],

```
       [0.28147605, 0.34318054, 0.37534347],
       [0.29321373, 0.36389884, 0.34288743],
       [0.22554258, 0.2554135 , 0.5190439 ],
       [0.2990215 , 0.3745828 , 0.32639563],
       [0.24667062, 0.2868602 , 0.4664692 ],
       [0.2990215 , 0.3745828 , 0.32639563],
       [0.2990215 , 0.3745828 , 0.32639563],
       [0.26364824, 0.31351623, 0.4228356 ],
       [0.1752586 , 0.1865203 , 0.6382211 ]], dtype=float32)
Y_pred = Y_pred.argmax(axis=1) #select highest probability of multiple class column wise
#and hold in Y_pred
#argmax : inbuilt method ,select highest prob. of multiple class
Y_pred
array([2, 2, 1, 2, 1, 2, 1, 1, 2, 2], dtype=int64)
list(y_test)
[1, 2, 1, 2, 1, 1, 1, 2, 2, 1]
y_test-Y_pred
32   -1
45    0
4     0
15    0
38    0
20   -1
16    0
7     1
18    0
11   -1
Name: NAM_b, dtype: int64
plt.subplot(211)
plt.scatter(Y_pred,[1,2,3,4,5,6,7,8,9,10], color='red', lw=5)
plt.scatter(y_test,[1,2,3,4,5,6,7,8,9,10], color='orange', lw=7)

plt.show()
```

```
import seaborn as sns

cm = confusion_matrix(y_test, y_pred)
ax = sns.heatmap(cm, annot=True, cmap='Blues')

ax.set_title('Seaborn Confusion Matrix with labels\n\n');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values ');

## Ticket labels - List must be in alphabetical order
ax.xaxis.set_ticklabels([1,2])
ax.yaxis.set_ticklabels([2,1])

## Display the visualization of the Confusion Matrix.
plt.show()

from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
              precision    recall  f1-score   support

           1       0.70      1.00      0.82         7
           2       0.00      0.00      0.00         3

    accuracy                           0.70        10
   macro avg       0.35      0.50      0.41        10
weighted avg       0.49      0.70      0.58        10

C:\Users\bagher\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
C:\Users\bagher\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\bagher\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
#
# DO NOT EDIT THIS FUNCTION; IF YOU WANT TO PLAY AROUND WITH DATA GENERATION,
# MAKE A COPY OF THIS FUNCTION AND THEN EDIT
#
import numpy as np
from sklearn.datasets import make_moons
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap

def generate_data(n_samples, tst_frac=0.2, val_frac=0.2):
  # Generate a non-linear data set
  X, y = make_moons(n_samples=n_samples, noise=0.25, random_state=42)

  # Take a small subset of the data and make it VERY noisy; that is, generate outliers
  m = 30
  np.random.seed(30)  # Deliberately use a different seed
  ind = np.random.permutation(n_samples)[:m]
  X[ind, :] += np.random.multivariate_normal([0, 0], np.eye(2), (m, ))
  y[ind] = 1 - y[ind]
```

```
  # Plot this data
  cmap = ListedColormap(['#b30065', '#178000'])
  plt.scatter(X[:, 0], X[:, 1], c=y, cmap=cmap, edgecolors='k')

  # First, we use train_test_split to partition (X, y) into training and test sets
  X_trn, X_tst, y_trn, y_tst = train_test_split(X, y, test_size=tst_frac,
                                        random_state=42)

  # Next, we use train_test_split to further partition (X_trn, y_trn) into training and validation sets
  X_trn, X_val, y_trn, y_val = train_test_split(X_trn, y_trn, test_size=val_frac,
                                        random_state=42)

  return (X_trn, y_trn), (X_val, y_val), (X_tst, y_tst)
#
#  DO NOT EDIT THIS FUNCTION; IF YOU WANT TO PLAY AROUND WITH
VISUALIZATION,
#  MAKE A COPY OF THIS FUNCTION AND THEN EDIT
#

def visualize(models, param, X, y):
  # Initialize plotting
  if len(models) % 3 == 0:
    nrows = len(models) // 3
  else:
    nrows = len(models) // 3 + 1

  fig, axes = plt.subplots(nrows=nrows, ncols=3, figsize=(15, 5.0 * nrows))
  cmap = ListedColormap(['#b30065', '#178000'])

  # Create a mesh
  xMin, xMax = X[:, 0].min() - 1, X[:, 0].max() + 1
  yMin, yMax = X[:, 1].min() - 1, X[:, 1].max() + 1
  xMesh, yMesh = np.meshgrid(np.arange(xMin, xMax, 0.01),
```

```
                    np.arange(yMin, yMax, 0.01))

  for i, (p, clf) in enumerate(models.items()):
    # if i > 0:
    #   break
    r, c = np.divmod(i, 3)
    ax = axes[r, c]

    # Plot contours
    zMesh = clf.decision_function(np.c_[xMesh.ravel(), yMesh.ravel()])
    zMesh = zMesh.reshape(xMesh.shape)
    ax.contourf(xMesh, yMesh, zMesh, cmap=plt.cm.PiYG, alpha=0.6)

    if (param == 'C' and p > 0.0) or (param == 'gamma'):
      ax.contour(xMesh, yMesh, zMesh, colors='k', levels=[-1, 0, 1],
              alpha=0.5, linestyles=['--', '-', '--'])

    # Plot data
    ax.scatter(X[:, 0], X[:, 1], c=y, cmap=cmap, edgecolors='k')
    ax.set_title('{0} = {1}'.format(param, p))
# Generate the data
# n_samples = 300    # Total size of data set
# (X_trn, y_trn), (X_val, y_val), (X_tst, y_tst) = generate_data(n_samples)
X_trn= x_train
X_tst =x_test
y_trn=y_train
y_tst= y_test
# Learn support vector classifiers with a radial-basis function kernel with
# fixed gamma = 1 / (n_features * X.std()) and different values of C
from sklearn.svm import SVC
import matplotlib.pyplot as plt
import numpy as np
C_range = np.arange(-3.0, 6.0, 1.0)
```

```
C_values = np.power(10.0, C_range)

models = dict()
trnErr = dict()
valErr = dict()

for C in C_values:
    models[C]=SVC(C=C, gamma='scale')
    models[C].fit(X_trn,y_trn)
    err1=models[C].score(X_trn,y_trn)
#     err2=models[C].score(X_val,y_val)
    trnErr[C]=1-err1
#     valErr[C]=1-err2

print(trnErr)
# print(valErr)

lists1=trnErr.items()
x1,y1=zip(*lists1)
# lists2=valErr.items()
# x2,y2=zip(*lists2)



# visualize(models, 'C', X_trn, y_trn)

plt.figure()
plt.plot(x1, y1, marker='o', linewidth=4, markersize=14)
# # plt.plot(x2, y2, marker='s', linewidth=4, markersize=14)
# plt.xscale('log')
# plt.xlabel('C', fontsize=16)
# plt.ylabel('Validation/Training error', fontsize=16)
# # plt.xticks(list(C_values), fontsize=12)
# # plt.legend(['Validation Error', 'Training Error'], fontsize=16)
```

```
# plt.axis()
# #
# #
idealvalue=SVC(C=1, gamma='scale')
idealvalue.fit(X_trn,y_trn)
Accuracy=idealvalue.score(X_tst,y_tst)
print(Accuracy)
#Accuracy = 0.833333333334

#DISCUSSION:
#When the value of C is the lowest the validation and training errors are the highest.
#As C increases the errors slowly start decreasing.
#It decreases till a certain point and then starts increasing again.
#After a certain point the validation error almost remains constant but the Training error starts
increasing
#The increase in the error is due to OVERFITTING.
#When the value of C is too small the model is undefitted whereas when the value of C is too high
the model is overfitted.



# The reason for the overfitting is because the model overshadows the regularization term of the
objective function and is
#dominated by the loss function. The model gives a higher weightage for C in the objective
function w′w+CΣni=1ℓ(w|xi,yi)
#hence leaving out the other term and overshadowing it.

#The optimum value of C for this data in 1.

#FINAL MODEL SELECTION:
#Accuracy = 0.83333333334
{0.001: 0.4054054054054054, 0.01: 0.4054054054054054, 0.1: 0.4054054054054054, 1.0:
0.10810810810810811, 10.0: 0.0, 100.0: 0.0, 1000.0: 0.0, 10000.0: 0.0, 100000.0: 0.0}
0.5
```

```
# Learn support vector classifiers with a radial-basis function kernel with
# fixed C = 10.0 and different values of gamma
gamma_range = np.arange(-2.0, 4.0, 1.0)
gamma_values = np.power(10.0, gamma_range)

models = dict()
trnErr = dict()
# valErr = dict()

for G in gamma_values:
    models[G]=SVC(C=10, gamma=G)
    models[G].fit(X_trn,y_trn)
    trnErr[G]=1-models[G].score(X_trn,y_trn)
#     valErr[G]=1-models[G].score(X_val,y_val)

print(trnErr)
# print(valErr)

lists1=trnErr.items()
x1,y1=zip(*lists1)



# visualize(models, 'gamma', X_trn, y_trn)

# plt.figure()
# plt.plot(x2, y2, marker='o', linewidth=4, markersize=14)
# plt.plot(x1, y1, marker='s', linewidth=4, markersize=14)
# plt.xscale('log')
# #plt.yscale('log')
# plt.xlabel('Gamma', fontsize=16)
# plt.ylabel('Validation/Training error', fontsize=16)
# plt.xticks(list(gamma_values), fontsize=12)
```

```
# plt.legend(['Validation Error', 'Training Error'], fontsize=16)
# plt.axis()

# idealvalue=SVC(C=10, gamma=1)
# idealvalue.fit(X_trn,y_trn)
# Accuracy=idealvalue.score(X_tst,y_tst)
print(Accuracy)
#Accuracy = 0.83333333333334
{0.01: 0.1351351351351351, 0.1: 0.0, 1.0: 0.0, 10.0: 0.0, 100.0: 0.0, 1000.0: 0.0}
0.5
from sklearn.svm import SVC
import matplotlib.pyplot as plt
C_range = np.arange(-3.0, 6.0, 1.0)
C_values = np.power(10.0, C_range)
gamma_range = np.arange(-2.0, 4.0, 1.0)
gamma_values = np.power(10.0, gamma_range)

trnErr = dict()
valErr = dict()
print(C_values)
print(gamma_values)
xtrain=x_train
ytrain=y_train
xtest=x_test
ytest=y_test
for i in range(len(C_values)):
    for j in range(len(gamma_values)):
        Z = SVC(C=C_values[i], gamma=gamma_values[j])
        Z.fit(xtrain,ytrain)
#         err1=Z.score(xvalidation,yvalidation)
        err2=Z.score(xtrain,ytrain)
        valErr=1-err1
        trnErr=1-err2
```

```
        print(C_values[i],'  ', gamma_values[j],'  ',trnErr,' ',valErr)

idealvalue=SVC(C=100, gamma=0.01)
idealvalue.fit(xtrain,ytrain)
Accuracy=idealvalue.score(xtest,ytest)
print(Accuracy)
[1.e-03 1.e-02 1.e-01 1.e+00 1.e+01 1.e+02 1.e+03 1.e+04 1.e+05]
[1.e-02 1.e-01 1.e+00 1.e+01 1.e+02 1.e+03]
0.001    0.01    0.4054054054054054   0.0
0.001    0.1    0.4054054054054054   0.0
0.001    1.0    0.4054054054054054   0.0
0.001    10.0    0.4054054054054054   0.0
0.001    100.0    0.4054054054054054   0.0
0.001    1000.0    0.4054054054054054   0.0
0.01    0.01    0.4054054054054054   0.0
0.01    0.1    0.4054054054054054   0.0
0.01    1.0    0.4054054054054054   0.0
0.01    10.0    0.4054054054054054   0.0
0.01    100.0    0.4054054054054054   0.0
0.01    1000.0    0.4054054054054054   0.0
0.1    0.01    0.4054054054054054   0.0
0.1    0.1    0.4054054054054054   0.0
0.1    1.0    0.4054054054054054   0.0
0.1    10.0    0.4054054054054054   0.0
0.1    100.0    0.4054054054054054   0.0
0.1    1000.0    0.4054054054054054   0.0
1.0    0.01    0.4054054054054054   0.0
1.0    0.1    0.0   0.0
1.0    1.0    0.0   0.0
1.0    10.0    0.0   0.0
1.0    100.0    0.0   0.0
1.0    1000.0    0.0   0.0
10.0    0.01    0.1351351351351351   0.0
```

```
10.0    0.1    0.0   0.0
10.0    1.0    0.0   0.0
10.0    10.0    0.0   0.0
10.0    100.0    0.0   0.0
10.0    1000.0    0.0   0.0
100.0    0.01    0.0   0.0
100.0    0.1    0.0   0.0
100.0    1.0    0.0   0.0
100.0    10.0    0.0   0.0
100.0    100.0    0.0   0.0
100.0    1000.0    0.0   0.0
1000.0    0.01    0.0   0.0
1000.0    0.1    0.0   0.0
1000.0    1.0    0.0   0.0
1000.0    10.0    0.0   0.0
1000.0    100.0    0.0   0.0
1000.0    1000.0    0.0   0.0
10000.0    0.01    0.0   0.0
10000.0    0.1    0.0   0.0
10000.0    1.0    0.0   0.0
10000.0    10.0    0.0   0.0
10000.0    100.0    0.0   0.0
10000.0    1000.0    0.0   0.0
100000.0    0.01    0.0   0.0
100000.0    0.1    0.0   0.0
100000.0    1.0    0.0   0.0
100000.0    10.0    0.0   0.0
100000.0    100.0    0.0   0.0
100000.0    1000.0    0.0   0.0
0.5

  File "C:\Users\bagher\AppData\Local\Temp/ipykernel_11660/1010759116.py", line 1
    jupyter-nbconvert --to PDFviaHTML
```

```
    ^
SyntaxError: invalid syntax
```

فصل پنجم: نتیجه گیری و پیشنهادات

۵-۱-مقدمه

در این فصل به بررسی روش پیشنهادی برای تشخیص بیماری بعد از عمل تشنج به روش یادگیری ماشین و استفاده از تئوری شبکه عصبی و رگرسیون ماشین بردار پشتیبان و درخت تصمیم پرداخته خواهد شد. و هدف ما پیش بینی NAM خواهد بود. در این فصل ابتدا به بررسی الگوریتم های ارایه شده در فصل قبل در شناسایی بیماری تشنج خواهیم. نهایتا اطلاعات آموزش یافته مدل‌های یادگیری ماشین پرداخته شده است. در ادامه نتایج و دقت و همگرایی الگوریتم ها بررسی خواهد شد.

۵-۲-مشخصه‌های موثر بر تشخیص بیماری‌های بعد از عمل تشنج

در فصل قبل به مشخصه‌های تاثیر گذار در شناسایی بیماری‌های تشنج اشاره شد و به طور کامل مورد تشریح قرار گرفت که بطور خلاصه در زیر دسته‌بندی شده است:

برای طبقه‌بندی نتایج بعد از عمل جراحی تشنج، جروم اِنگِل طرح زیر را پیشنهاد کرد، [۱] مقیاس نتیجه جراحی تشنج اِنگِل، که به استاندارد واقعی برای گزارش نتایج در ادبیات پزشکی تبدیل شده است:

کلاس I: بدون تشنج ناتوان کننده

کلاس II: تشنج های ناتوان کننده نادر (تقریباً بدون تشنج)

کلاس III: بهبود ارزشمند

کلاس IV: هیچ پیشرفت قابل توجهی وجود ندارد

ENG_b: Engel score = 1 (1 = ENG+); Engel score = 2, 3 or 4 (2 = ENG-)

- (ENG+) optimal postoperative clinical outcome: the Engel score of I, no seizures observed 2 years after surgery;

(ENG-) suboptimal postoperative clinical outcome: the Engel score of II-IV: persistence of more or less frequent and more or less severe seizures 2 years after surgery;

- (NAM+) optimal postoperative naming outcome: the naming score at 2 years after surgery did not decline or even improved compared to the presurgical baseline;

- (NAM-) suboptimal postoperative naming outcome: the naming score at 2 years after surgery has decreased compared to the presurgical baseline

مشخصه های بعد از عمل:

AGE: Age at time of neurosurgery (in years)

DIST: distance of the post-op workup from the date of surgery (in years)

ENG: Engel score at two years post-surgery (Engel, 1993, 2013)

ENG_b: Engel score = 1 (1 = ENG+); Engel score = 2, 3 or 4 (2 = ENG-)

NAM: Naming DO80 at two years post-surgery (Metz-Lutz et al., 1991)

NAM_b: NAM Reliable Change Index > -1.28 SD (1 = NAM +); NAM RCI ≤ -1.28 SD (2 = NAM-)

مشخصه‌ی عمومی بیماران:

AGE: Age of patient (in years)

EDU: Educational level (in years)

GEN: Gender (Male/Female)

HDS: Handedness (Edinburgh laterality index)

HDS_b: Handedness (Left/Right)

مشخصه‌ی بیماران تشنج:

HEM: Hemispheric lateralization of the temporal lobe epilepsy (L = Left; R = right)

ASO: Age of seizures onset (in years)

AHS: Hippocampal asymmetry (absolute volume difference)

DUR: Duration of the epilepsy (in years)

DUR_b: Duration ≤ 15years (1); > 15years (2)

HS_b: Hippocampal sclerosis (yes/no)

FRQ: Frequency of seizures (by months)

AED: Antiepileptic drugs (daily taken)

SEV: Severity of the epilepsy (composite score deriving from DUR, FRQ and AED)

SEV_b: Severity composite score ≤ 6 (1); > 6 (2)

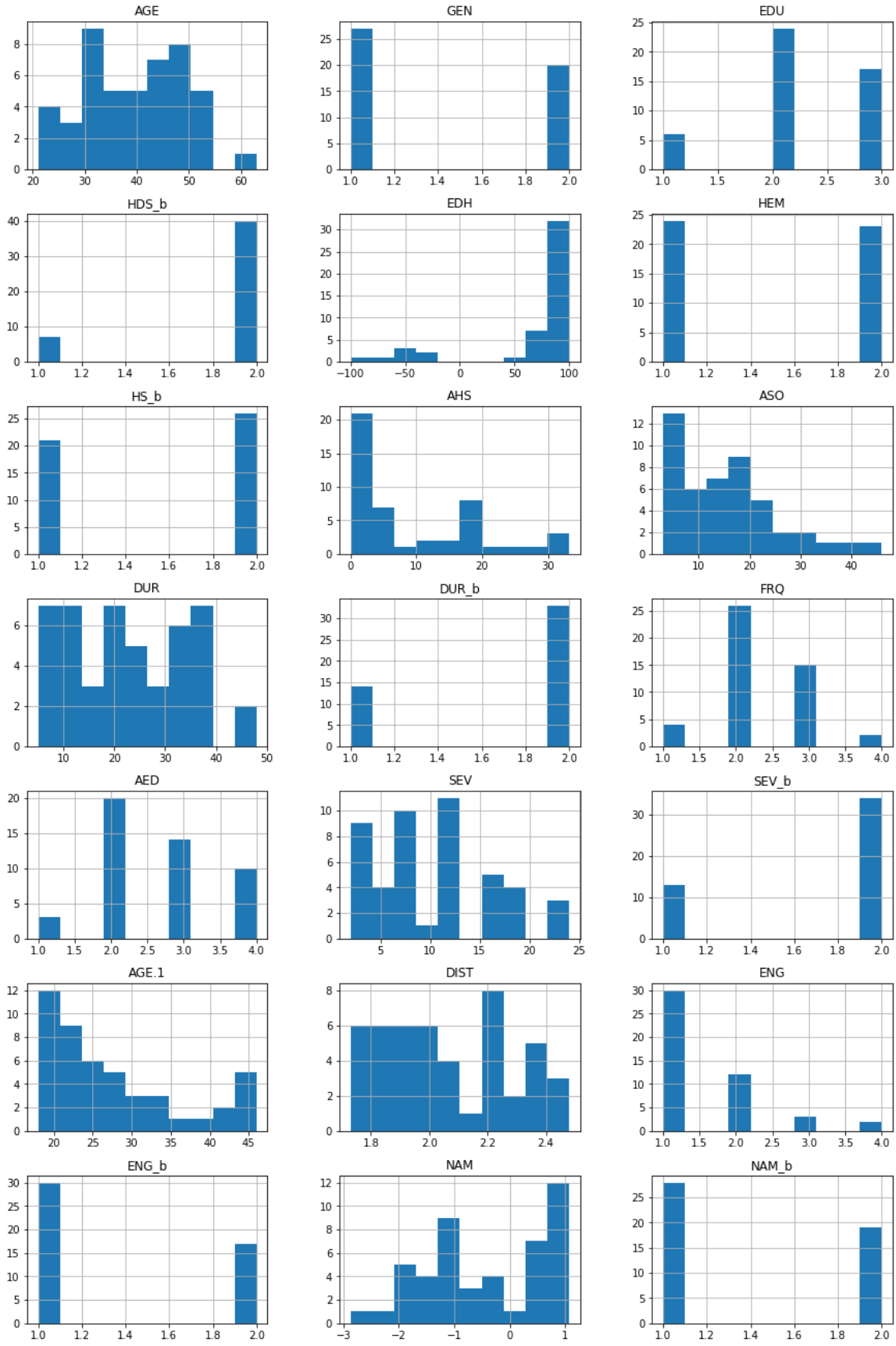
در روش پیشنهادی ما از یک دیتاست استاندارد برگرفته شده از مقاله[۵۴] استفاده شده است که تنها دیتاست استاندارد مربوط به موضوع تحقیق که دسترس عموم است می باشد. این دیتاست دارای ۲ کلاس و ۷۶ رکورد و ۲۱ستون ویژگی می باشد.

۵-۳-نتایج خروجی الگوریتم ها

ما داده های به دو دسته آموزش و تست ۲۰/۸۰ تقسیم کرده ایم و هدف اصلی ما پیش بینی HEM بیمار بعد از عمل خواهد بود که مشخص شود آیا عمل انجام شده منجر به بهبود صرع راست یا چپ بیمار شده است یا خیر. و ما دو کلاس ۱و۲ را که معنی صرع راست و چپ است را در آخر پیش بینی میکنیم

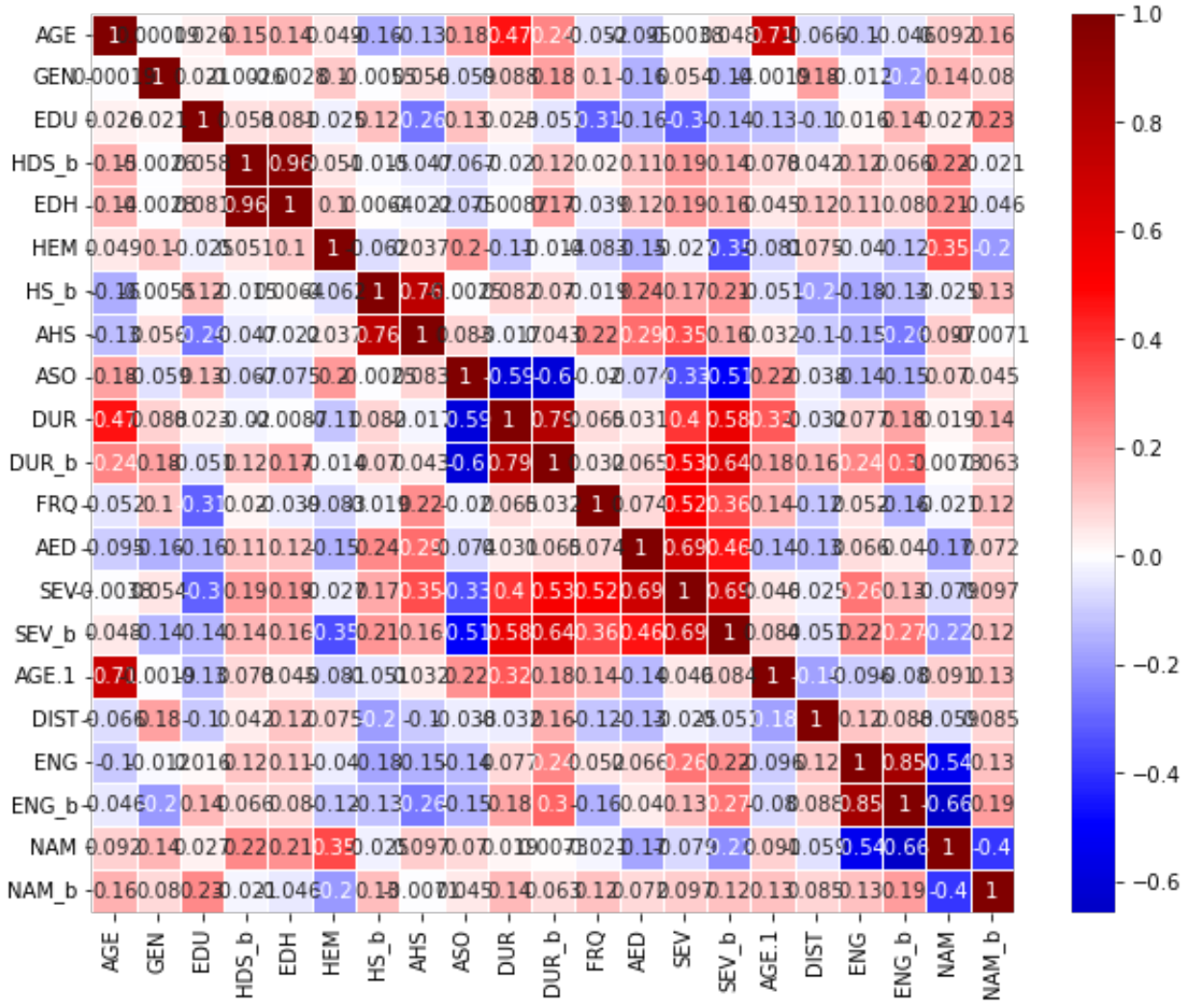
.۵-۴-تحلیل و بررسی دیتاست

به منظور بررسی و تحلیل داده‌های موجود ما یک نمای کلی از توزیع داده ها ترسیم خواهیم کرد همانطور که مشخص است توزیع AGE و DIST بیشتر از داده های دیگر است.



شکل ۵-۱- توزیع داده های مختلف ۲۱ ستون

در شکل زیر توزیع همبستگی داده ها براساس فرمول correlation بدست آمده است. مشاهده میشود که همبستگی بین NAM و DUR و ENG و SEV_b و HEM بیشتر شده است. ولی ما در پیش بینی ENG را حذف کردیم زیرا ENG مشخصه‌ی همان NAM است.



شکل۲-۵- همبستگی بین داده ها

۵-۵-ارزیابی F1 score

با در نظر گرفتن برنامه های کاربردی برای پیش بینی بیمار یا سالم، مدل پیشنهادی خود را بر روی داده برای تأیید اعمال گشته است. ما نتایج خود را با طبقه‌بندی کننده‌های پایه RF,SVMوMLP مقایسه کردیم. تمام نتایج به دست آمده با استفاده از نمره F1 تأیید می شود تا از عدالت[۱] و سازگاری[۲] نتایج مقایسه زیر اطمینان حاصل شود. عملکرد خوب روش RF، که بیماری را در سطح نمونه مدل می کند، پتانسیل کاربردهای جدید در مدل پیشنهادی دارد، زیرا این مدل نیاز به اعمال کمترین تقسیم‌بندی داده‌ها را دارد و کاربردی‌تر شده است و در نتیجه هر مقدار در دسته‌ی درست خودش قرار گرفته است یا به نوعی پیش‌بینی دقیقتری انجام داده است.

جدول ۵-۱ محاسبه ی F1 scoreها

TARGET/ MODEL	MLP	SVM	RF
YES	0.92921	0.94496	**0.95317**
NO	0.89397	0.91954	**0.89888**

در نتایج جدول ۵-۲ همانطور که مشخص است میزان صحت بالاتر در مدل پیشنهادی بیشتر صادق است به خصوص در یافتن اهداف YESوNO. مدل RF در واقع با بهتر در نظر گرفتن ویژگیها نتایج در کنار یکدیگر و آموزش دقیقتر خطای شبکه را کاهش دهد، زیرا خطای باقی‌مانده در ریشه را به برگ‌ها با کمترین خطا انتقال میدهد و این مهم باعث می‌شود خطای مدل جنگل تصادفی بیش از هد مورد نیاز نشود مثل یک تابع همانی که هر چه در ورودی به آن بدهید خروجی همان را مشاهده خواهید کرد، و این مزیت مدل پیشنهادی ما را می‌رساند.

در مدل SVM به کل چون شبکه توانایی انتقال ورودی به خروجی را ندارد از هر نقطه به نقطه دیگر نمی‌تواند تجربیات را منتقل کند و این مهم باعث می‌شود قدم به قدم خطا زیاد شود و در هر دور از اجرای الگوریتم فرصت نشود این موارد اصلاح شود.

[۱] fairness

[۲] consistency

در مدل MLP شبکه دارای چندین لایه هست و این مهم باعث شده که در هر عمقی که وارد می‌شود به صورت یک نگاشت نتایج در هر نقطه منتقل کند ولی این نتایج آموزش دیده نیستند. بنابراین نسبت به مدل پیشنهادی دقت بیشتری را تجربه نکرده ایم.

اما در بحث پوشش یا خطای بدست آمده همانطور که مشخص است باتوجه به شواهد ذکر شده، در مدل MLP ما شاهد کمی overfit نسبت به مدل RF هستیم که این نشان می‌دهد خطای شبکه را زیاد کرده است و به علت آموزش بیش از حد این اتفاق افتاده است. و در واقع زمانی آنهایی که صحیح هستند را شبکه به غلط تشخیص داده است. در مدل SVM نیز این مشاهدات غلط بیشتر نیز ملموس است زیرا مدل قدرت تشخیص لازم را نیز نداشته است.

جدول ۵-۲ محاسبه‌ی معیار پوشش و صحت[۱۲]

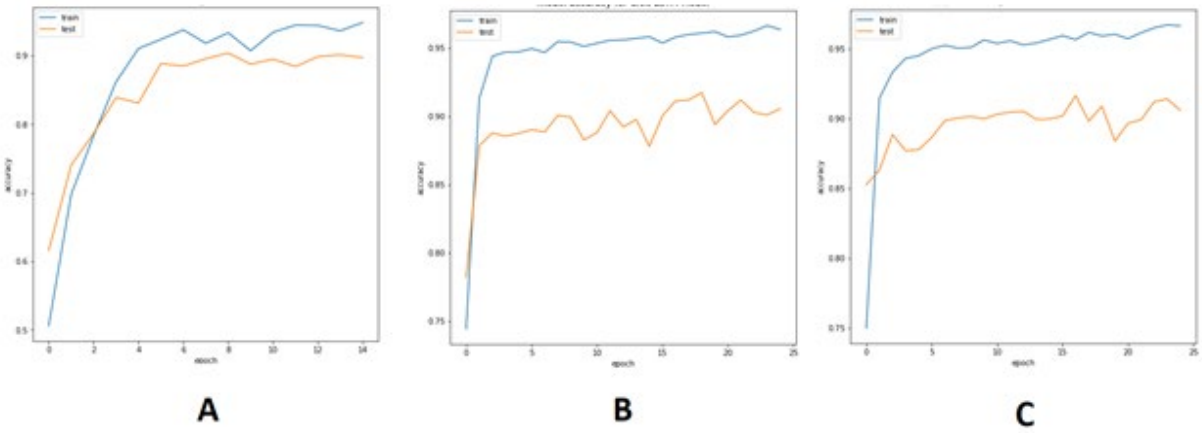
Target/Model	MLP		SVM		RF	
	precision	recall	precision	recall	precision	Recall
YES	0.91913	0.93952	0.9743	0.91734	0.98495	0.92339
NO	0.87576	0.91295	0.90535	0.93418	0.86614	0.93418

در شکل ۵-۳ بعد از هر epoch مقدار خطای روی داده تست و داده آموزش بدست آمده است، در مدل MLP نسبت این دو خطا کم نشان داده شده است ولی به جواب بهتری نیز همگرا نمی‌شود اما در مدل پیشنهادی (جنگل تصادفی) این نسبت از هم فاصله گرفتن در واقع آموزش و تست از هم نسبت دورتری دارند که ممکن است جواب‌ها overfit کنند.

همچنین این نتایج بعد از چندین epoch (بستگی به پیچیدگی مدل و حجم داده آموزش و ...) قابل مشاهده است که نمودارها برای هر دو (دقت داده آموزش و داده تست) همچنان در حال افزایش است. تا جایی که مشاهده می شود که نمودار خطای داده آموزش نسبت بالاتری به تست دارد، اما نقاطی که گاهی مینمم می‌شود این نقاط شروع یک overfitting متعددی در مساله است، بنابراین باید یادگیری مدل را متوقف کرد زیرا خطای تست را به مرور زمان افزایش خواهد داد. در مدل RF میزان مینمم‌های خطای تست کمتر از مدل SVM است زیرا در هر فاز

[۱] Recall
[۲] Precision

آموزش دقت افزایش می‌یابد اما در مدل MLP با اینکه دقت کمتر است ولی مقدار مینمم‌های خطایی در تست نیز کمتر مشاهده می‌شود و این مهم به علت نداشتن آموزش در هر دور از اجرای MLP است.



شکل ۵-۳- نتایج دقت آموزش و تست در سه مدل A- مدل MLP و B- مدل SVM و C- مدل RF

۵-۶- نتیجه گیری

این تحقیق مبتنی بر داده عکس‌های بیماران صورت گرفته است، که مشخص عکس‌های بیماران تشنج می تواند به عنوان یک پیش بینی کننده مرتبط برای نتایج نامگذاری بالینی و طولانی مدت پس از لوبکتومی گیجگاهی عمل کند. در بیماران مبتلا به تشنج در این زمینه، روش‌های یادگیری ماشین استفاده شده و مشخص شده است که مدل جنگل تصادفی بهترین مدل پیشنهادی بوده است. در نتیجه الگوریتم جنگل تصادفی به دلیل تجزیه پذیری داده به نسبت مدلهای دیگر کاراتر خواهد بود و همانطور که مشاهده شده است دقت ۰,۹ درصد را بدست

منابع

[۱]- غفرانی محمد، اشرفی محمودرضا، بیماری های مغز و اعصاب کودکان ویژه خانواده .چاپ اول . نهضت پویا، ۱۳

[2] Kaplan H, Sadock Bj. Synopsis of psychiatry. 9th ed. Baltimore: Wiliams and Wilkins; ۱۰۳۹-۲۰۰۳:۱۰۲۹.

[3] Katzung BG. Basic & clinical pharmacology. 8thed. Fast Norwalk: Apleton and langer; ۶۷۸-۲۰۰۱:۶۶۴.

[4] U. R. Acharya, H. Fujita, V. K Sudarshan, S. Bhat, J. E.W. Koh,"Application of entropies for automated diagnosis of epilepsy using EEG signals: A review," Knowledge-Based Systems, vol. 88, pp. 88- 99,November 2038,.

، [۵] کیهانی دوست زهرا، ورزش و بیماری صرع، فصلنامه بیماری های کودکان، ۴۱۳)۱۱(۲۳- ۵

[6] Klug JM, Shinnar S. MRI in brain. American radiology 2005; 27(3): 43-48

[7] Gaillard WD, Chiron C, Cross JH, Harvey AS, Kuzniecky R, Hertz-Pannier L, Vezina LG; ILAE, Committee for Neuroimaging, Subcommittee for Pediatric. Guidelines for imaging infants and children with recent-onset epilepsy. Epilepsia. 2009 Sep;50(9):2147- 53.

[8] Guissard G, Damry N, Dan B, David P, Sékhara T, Ziereisen F, Christophe C. Imaging in paediatric epilepsy. Arch Pediatr. 2005 Mar;12(3):337-46

[9] Scott R.C, Gadian D.G, King M.d and etal. Magnetic resonance imaging findings within 5 days of status epilepticus in childhood. Brain (2002) 125 (9): 1951-1959

[10] Yoong M, Madari R, Martinos M, Clark C, Chong K, Neville B, Chin R, Scott R. The role of magnetic resonance imaging in the follow-up of children with convulsive status epilepticus.

Dev Med Child Neurol. 2012 Apr;54(4):328-33

[11] Wongladarom S, Laothamatas J, Visudtibhan A, Sawatsut P. Magnetic resonance imaging of the brain in epileptic pediatric patients: review of the experience in Ramathibodi Hospital. J Med Assoc Thai. 2004 Sep;87(9):1092-9.

[12] Maytal J, Krauss JM, Novak G, Nagelberg J, Patel M. The role of brain computed tomography in evaluating children with new onset of seizures in the emergency department. Epilepsia. 2000 Aug;41(8):950-4

[13] Garvey MA, Gaillard WD, Rusin JA, Ochsenschlager D, Weinstein S, Conry JA, Winkfield DR, Vezina LG. Emergency brain computed tomography in children with seizures: who is most likely to benefit? J Pediatr. 1998 Nov;133(5):664-9.

[14] Fallah R, nafisi moghadam R, fallah tafti m. Result of non contrast brain computed tomography scan of 1-18 year old epileptic children. Iran Child neurol 2012; 6(3): 33-38.

[۱۵]-نتایج و پاراکلینیک های یافته بررسی .محمد غفرانی محمد، محوالتی مهدی، محمد ناصحی

درمانی در کودکان مبتال به صرع مقاوم مراجعه کننده به بیمارستان مفید در سال۸

[16] https://www.webmd.com/brain/picture-of-the-brain

[17] Sandrone; et al. (2012). "Angelo Mosso". Journal of Neurology. 259 (11): 2513–2514

[18] Sandrone; et al. (2014). "Weighing brain activity with the balance: Angelo Mosso's original manuscripts come to light". Brain. 137 (Pt 2): 621–633.

[19] iller AG (2009). "The history, development, and impact of computed imaging in neurological diagnosis and neurosurgery: CT, MRI, DTI". Nature Precedings.

[20] Per E. Roland and Lars Friberg (1985). "Localization of cortical areas activated by thinking". Journal of Neurophysiology. Vol. 53, no. 5. pp. 1219–1243.

[21] Gonzalez, Rafael C. (2008). Digital image processing. Woods, Richard E. (Richard Eugene), 1954- (3rd ed.). Upper Saddle River, N.J.: Prentice Hall. pp. 23–28. ISBN 9780131687288. OCLC 137312858.

https://en.m.wikipedia.org/wiki/Digital_image_processing

[22] Epilepsy during the Middle Ages, the Renaissance and the Enlightenment .Aristidis Diamantis 1, Kalliopi Sidiropoulou, Emmanouil Magiorkinis ,PMID: 20037763 DOI: 10.1007/s00415-009-5433-7

[23] Schweiz Rundsch Med Prax. 1989 Jul 18;78(29-30):816-24.[The mind and epilepsy: opinions and viewpoints over the course of time]C VaneyPMID: 2678362

[24] Epileptic Disord. 2014 Sep;16(3):261-9. doi: 10.1684/epd.2014.0676.History of epilepsy: nosological concepts and classification.Peter Wolf 1.PMID: 25256654 DOI: 10.1684/epd.2014.0676

[25] Chang BS, Lowenstein DH (2003). "Epilepsy". N. Engl. J. Med. **349** (13): 1257–66. doi:10.1056/NEJMra022308. PMID 14507951.

[26] "Epilepsy". Fact Sheets. World Health Organization. October 2012. Retrieved January 24, 2013.

[27] Fisher R, van Emde Boas W, Blume W, Elger C, Genton P, Lee P, Engel J (2005). "Epileptic seizures and epilepsy: definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE)". Epilepsia. **46** (4): 470–2. doi:10.1111/j.0013-9580.2005.66104.x. PMID 15816939

[28] Eadie, MJ (December 2012). "Shortcomings in the current treatment of epilepsy". Expert review of neurotherapeutics. **12** (12): 1419–27. doi:10.1586/ern.12.129. PMID 23237349.

[29] Thurman, DJ (September 2011). "Standards for epidemiologic studies and surveillance of epilepsy". Epilepsia. 52 Suppl 7: 2–26. doi:10.1111/j.1528-1167.2011.03121.x. PMID 21899536.

[30] Brodie, MJ (November 2009). "Epilepsy in later life". Lancet neurology. **8** (11): 1019–30. doi:10.1016/S1474-4422(09)70240-6. PMID 19800848.

[31] Holmes, Thomas R. Browne, Gregory L. (2008). Handbook of epilepsy (4th ed.). Philadelphia: Lippincott Williams & Wilkins. p. 7. ISBN 978-0-7817-7397-3.

[32] Wyllie's treatment of epilepsy: principles and practice (5th ed.). Philadelphia: Wolters Kluwer/Lippincott Williams & Wilkins. 2010. ISBN 978-1-58255-937-7.

[33] Newton, CR (29 September 2012). "Epilepsy in poor regions of the world". Lancet. **380** (9848): 1193–201. doi:10.1016/S0140-6736(12)61381-6. PMID 23021288.

[34] Wilden, JA (15 August 2012). "Evaluation of first nonfebrile seizures". American family physician. **86** (4): 334–40. PMID 22963022. [35] Berg, AT (2008). "Risk of recurrence after a first unprovoked seizure". Epilepsia. 49 Suppl 1: 13–8. doi:10.1111/j.1528-1167.2008.01444.x. PMID 18184149.

[36] L Devlin, A (December 2012). "Epilepsy and driving: current status of research". Epilepsy research. **102** (3): 135–52. doi:10.1016/j.eplepsyres.2012.08.003. PMID 22981339. [37] حدود ۷۰ تا ۸۰ درصد بیماری صرع به‌طور کامل درمان می‌شود . [خبرگزاری جمهوری اسلامی(ایرنا) http://www.irna.ir]

[38] T Fakhoury 1, B Abou-Khalil, E Peguero, Differentiating clinical features of right and left temporal lobe seizures. https://pubmed.ncbi.nlm.nih.gov/7925149/ Comparative Study Epilepsia . Sep-Oct 1994;35(5):1038-44.

[39]. Shouman, M., Turner, T., & Stocker, R. (2012). Using data mining techniques in heart disease diagnosis and treatment. Paper presented at the Electronics, Communications and Computers (JEC-ECC), 2012 Japan-Egypt Conference on.

[40] Craven, M. W., & Shavlik, J. W. (1997). Using neural networks for data mining. Future generation computer systems, 13(2), 211-229.

[41] Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. Expert Systems with applications, 29(4), 927-940.

[42] Hui, S. C., & Jha, G. (2000). Data mining for customer service support.Information & Management, 38(1), 1-13.

[43] Ngai, E. W., Xiu, L., & Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. Expert systems with applications, 36(2), 2592-2602.

[44] Linoff, G. S., & Berry, M. J. (2011). Data mining techniques: for marketing, sales, and customer relationship management. John Wiley & Sons.

[45] Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesús, O. (1996). Neural network design (Vol. 20). Boston: PWS publishing company.

[46] Widrow, B., & Lehr, M. A. (1990). 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proceedings of the IEEE, 78(9), 1415-1442.

[47]Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis(Vol. 3). New York: Wiley.

[48] Lippmann, R. P. (1989). Pattern classification using neural networks. IEEE communications magazine, 27(11), 47-50.

[49] Pal, M. (2005). Random forest classifier for remote sensing classification. International journal of remote sensing, 26(1), 217-222.

[50] Peterson, L. E. (2009). K-nearest neighbor. Scholarpedia, 4(2), 1883.

[51] Fung, G., Mangasarian, O., & Shavlik, J. (2002). Knowledge-based support vector machine classifiers. Advances in neural information processing systems, 15.

[52] Cheng, W., & Hüllermeier, E. (2009). Combining instance-based learning and logistic regression for multilabel classification. Machine Learning, 76(2), 211-225.

[53] Engel, Jerome (1993). Surgical Treatment of the Epilepsies. Lippincott Williams & Wilkins. ISBN 0-88167-988-7.

[54] Roger, E., Torlay, L., Banjac, S., Mosca, C., Minotti, L., Kahane, P., & Baciu, M. (2021). Prediction of the clinical and naming status after anterior temporal lobe resection in patients with epilepsy. Epilepsy & Behavior, 124, 108357.