



پردیس علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

تشخیص وجود سوفل در قلب بیماران با کمک مدل‌های مبتنی بر یادگیری با نظارت

نگارنده

فاطمه صبورروح منفرد

استاد راهنما: دکتر سید مرتضی امینی

پایان‌نامه برای دریافت درجه کارشناسی در رشته آمار

زمستان ۱۴۰۱

چکیده

تحلیل داده‌های صوتی با استفاده از مدل‌های مبتنی بر یادگیری ماشین امروزه به کمک علوم زیادی نظیر پزشکی آمده است. با استفاده تحلیل داده‌های صوتی می‌توان بیماری‌های زیادی را تشخیص داد و از تحمیل هزینه‌های درمان بالا پیشگیری کرد. در این پایان‌نامه سعی می‌کنیم با استفاده از تحلیل داده‌های صوتی با استفاده از چند مدل مبتنی بر آموزش با نظارت تشخیص دهیم که آیا افراد مشکل سوفل در قلب خود دارند و همچنین آیا قلب آن‌ها در وضعیت عادی یا غیرعادی به سر می‌برد. در انتها میزان دقت هر مدل را اندازه‌گیری کرده و نتیجه می‌گیریم که کدامیک عملکرد بهتری داشته است.

سپاسگزاری

از جناب آقای دکتر سید مرتضی امینی که در امر مشاوره این رساله مساعدت نمودند و در این امر نهایت مراقبت، توجه و دقت خود را مبذول فرموده اند کمال تشکر و امتنان را دارم .

پیشگفتار

امروزه تحلیل داده‌های صوتی با استفاده از مدل‌های یادگیری ماشین بیش از هر زمانی مورد استفاده قرار می‌گیرند. استخراج مفاهیم و اطلاعات می‌تواند به اشخاص، ارگان‌ها و شرکت‌ها در تصمیم‌گیری بهتر در مسائل و کسب موفقیت بیشتر کمک نماید.

علم پزشکی نیز از این قاعده مستثنی نمی‌باشد. تشخیص به موقع بیماری به ویژه در بیمارانی که در سن حساسی نظیر نوزادی قرار دارند، می‌تواند هزینه‌های درمان برای بیمار و نظام پزشکی یک کشور را به طرز قابل توجهی کاهش دهد.

یکی از عوامل شایع مرگ‌ومیر در میان کودکان و نوزادان مشکلات قلبی می‌باشد. به همین منظور می‌توان از تحلیل داده‌های صوتی استخراج شده از مکان‌های مختلف از قلب این افراد و اطلاعات جمعیت‌شناختی دیگر نظیر سن و جنسیت موارد مشکوک را شناسایی نمود.

در این پروژه قصد داریم با استفاده از مدل‌های مبتنی بر آموزش با نظارت نظیر جنگل تصادفی، رگرسیون لوجستیک و ماشین بردار پشتیبان به تحلیل داده‌های صوتی جمع‌آوری شده از افراد در سنین مختلف پردازیم و تشخیص بدهیم که آیا شخص سوپل دارد یا خیر و همچنین آیا تشخیص بالینی او عادی یا غیرعادی است.

در انتها پس از پیاده‌سازی مدل‌های مذکور و محاسبه دقت آن‌ها نتیجه‌گیری خواهیم کرد که آیا این مدل‌های می‌توانند به دقت مناسبی در چنین مسئله‌ای دست یابند یا خیر.

فهرست مطالب

چ	۱	بیان مسئله
ح	۱.۱	مجموعه داده
خ	۲.۱	ارائه مدل به منظور حل مسئله
۱	۲	مفاهیم مقدماتی
۲	۱.۲	یادگیری با نظارت
۲	۲.۲	یادگیری بی نظارت
۲	۳.۲	یادگیری نیمه نظارتی
۳	۴.۲	یادگیری تقویتی
۳	۵.۲	تبدیل فوریه
۴	۶.۲	ضرایب کپسترال مل ^۱
۵	۳	مدل‌های ابتدایی
۵	۱.۳	جنگل تصادفی
۵	۱.۱.۳	آشنایی با الگوریتم جنگل تصادفی
۶	۲.۱.۳	درخت تصمیم
۷	۳.۱.۳	برتری جنگل تصادفی بر درخت تصمیم
۸	۴.۱.۳	کیسه‌گذاری درختان ^۲
۸	۲.۳	رگرسیون لجستیک

^۱Mel Frequency Cepstral Coefficients

^۲Bagging

۱۰	برآورد پارامترهای بهینه	۱.۲.۳
۱۲	تنظیم مدل ^۳	۲.۲.۳
۱۳	ماشین بردار پشتیبان	۳.۳
۱۴	ماشین بردار پشتیبان خطی	۱.۳.۳
۱۵	حاشیه سخت ^۴	۲.۳.۳
۱۸	حاشیه نرم ^۵	۳.۳.۳
۱۹	ماشین بردار پشتیبان غیرخطی	۴.۳.۳
۲۰	طبقه‌بند بردار پشتیبان چندطبقه ^۶	۵.۳.۳
۲۲	شبکه‌های عصبی بازگشتی	۴
۲۲	شبکه‌های عصبی	۱.۴
۲۳	شبکه‌های عصبی بازگشتی (مکرر) ^۷	۲.۴
۲۴	حافظه طولانی کوتاه-مدت ^۸	۳.۴
۲۴	شبکه‌های عصبی بازگشتی دو طرفه ^۹	۴.۴
۲۶	بررسی مدل‌های پیاده‌سازی شده	۵
۲۶	استخراج ویژگی‌ها ^{۱۰}	۱.۵
۲۸	پیاده‌سازی مدل‌ها	۲.۵
۲۸	جنگل تصادفی	۱.۲.۵
۲۸	رگرسیون لوجستیک	۲.۲.۵
۲۸	بردار پشتیبان چندطبقه	۳.۲.۵
۲۹	بررسی یک مدل مبتنی بر یادگیری عمیق	۳.۵
۲۹	استخراج ویژگی‌ها	۱.۳.۵
۲۹	معماری شبکه عصبی	۲.۳.۵

³Regularization

⁴Hard-Margin

⁵Soft-Margin

⁶Multi-Class SVC

⁷Recurrent Neural Network

⁸Long Short-Term Memory

⁹Bidirectional Recurrent Neural Networks

¹⁰Feature Extraction

۴.۵ ارزیابی مدل‌ها ۳۰

۶ جمع‌بندی ۳۲

فصل ۱

بیان مسئله

بیماری‌های مادرزادی قلب حدود یک درصد از نوزادان را تحت تاثیر قرار می‌دهد. این بیماری‌ها یک عامل مهم مرگ و میر و شامل بیماری‌هایی جدی از جمله نارسایی قلبی پیشرفته می‌شود [۱] در یک تحقیق در سال ۲۰۱۹، تخمین زده شد که بیماری‌های مادرزادی قلبی بیش از ۵۰۰۰۰۰ کودک در شرق آفریقا [۲] و حدود ۸ کودک از هر ۱۰۰۰ تولد را تحت تاثیر قرار می‌دهند [۳]. بیماری‌های قلبی اکتسابی شامل تب روماتیسمی و بیماری کاوازاکی است که اولی یک مشکل جدی بهداشت عمومی در مناطق در حال توسعه مانند روستایی برزیل است [۴].

مناطق زیادی در کشورهای در حال توسعه در تشخیص و درمان بیماری‌های قلبی مادرزادی و اکتسابی در کودکان مشکل دارند. این امر عمدتاً به دلیل کمبود زیرساخت‌ها و متخصصان قلب و عروق در مناطق وسیع جغرافیایی و مشکل در دسترسی به خدمات بهداشتی است. علاوه بر این، همه‌گیری کووید-۱۹ کنونی مشکلات جدیدی را در ارزیابی بالینی بیماران با به تأخیر انداختن تماس‌های مهم حضوری بیمار و پزشک ایجاد می‌کند که تأثیر منفی بر فعالیت‌های غربالگری و نظارت دارد.

ارزیابی غیرتهاجمی عملکرد مکانیکی قلب، که در مراکز مراقبتی انجام می‌شود، می‌تواند اطلاعات اولیه در مورد بیماری‌های قلبی مادرزادی و اکتسابی در کودکان ارائه دهد. فقدان تشخیص زودهنگام این شرایط نشان دهنده یک مشکل عمده بهداشت عمومی، به ویژه در کشورهای محروم با نرخ زاد و ولد بالا است [۵، ۶، ۷]. به طور خاص، سمع قلبی و تجزیه و تحلیل فونوکاردیوگرام^۱ PCG می‌تواند اطلاعات بالینی اساسی در مورد اختلال عملکرد قلب ناشی از بیماری قلبی مادرزادی

^۱Phonocardiogram

و اکتسابی در جمعیت کودکان را آشکار کند. این با تشخیص امواج صوتی غیر طبیعی یا سوفل قلبی در سیگنال PCG به دست می آید. سوفل امواج غیرطبیعی هستند که توسط جریان خون آشفته در ساختارهای قلبی و عروقی ایجاد می شوند. آنها ارتباط نزدیکی با بیماری های خاصی مانند نقص دیواره بینی، نارسایی بسته شدن مجرای شریانی در نوزادان و دریچه های قلب معیوب دارند.

۱.۱ مجموعه داده

داده های استفاده شده شامل یک یا چند صدای ضبط شده قلب برای ۱۵۶۸ بیمار و همچنین اطلاعات جمعیت شناختی معمول در مورد بیماران است. برجسب ها از دو نوع تشکیل شده است:

۱. برجسب های مربوط به سوفل، نشان می دهند که آیا متخصص وجود یا عدم وجود سوفل را در یک بیمار از صدای ضبط شده تشخیص داده است یا اینکه در مورد وجود یا عدم وجود سوفل مطمئن نبوده است.

۲. برجسب های مرتبط با نتیجه، نتیجه بالینی طبیعی یا غیرطبیعی را نشان می دهند که توسط متخصص پزشکی تشخیص داده شده است.

داده های چالش از اطفال طی دو کمپین غربالگری انبوه اجرا شده در شمال شرقی برزیل در ژوئیه-آگوست ۲۰۱۴ و ژوئن-ژوئیه ۲۰۱۵ جمع آوری شده است. جامعه هدف افراد ۲۱ ساله یا کمتر بودند که به طور داوطلبانه با رضایت نامه امضا شده والدین یا قیم قانونی برای غربالگری مراجعه کردند. همه شرکت کنندگان یک پرسشنامه جمعیت شناختی اجتماعی را تکمیل کردند و متعاقباً تحت معاینه بالینی، ارزیابی پرستاری و بررسی های قلبی قرار گرفتند. شرح مفصلی از مجموعه داده را می توان در [۸] یافت.

هر بیمار در داده های چالش، یک یا چند مورد ضبط شده از یک یا چند مکان برجسته سمع را دارد: دریچه ریوی^۲ (PV)، دریچه آئورت^۳ (AV)، دریچه میترال^۴ (MV)، دریچه سه لتی^۵ (TV)، و سایر (Phc). ضبط ها به صورت متوالی (نه به طور همزمان) از مکان های مختلف با

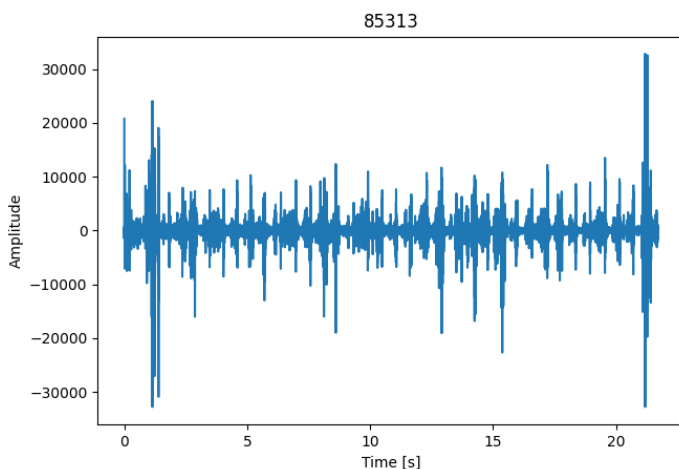
^۲Pulmonary Valve

^۳Aortic Valve

^۴Mitral Valve

^۵Tricuspid Valve

استفاده از گوشی پزشکی دیجیتال جمع آوری شدند. تعداد، مکان و مدت زمان ضبط بین بیماران متفاوت است. نمونه‌ای از این فایل‌های صوتی که به فرم یک نمودار سری زمانی درآمده، در شکل زیر مشخص است.



شکل ۱.۱: نمونه فایل صوتی بیمار با شناسه ۸۵۳۱۳

۲.۱ ارائه مدل به منظور حل مسئله

هدف از این پروژه ارائه مدلی مبتنی بر آموزش با نظارت به منظور شناسایی وجود، عدم وجود یا موارد نامشخص سופل و پیامدهای بالینی طبیعی در مقابل غیرطبیعی از صداهای ضبط شده از قلب است که از چندین مکان در بدن با استفاده از گوشی پزشکی دیجیتال جمع آوری شده است. برای هر بیمار (مستقل از تعداد مکان‌های ضبط)، مدل باید برچسب طبقه (وجود، عدم وجود، نامشخص) و همچنین یک امتیاز احتمال یا اطمینان برای هر طبقه مشخص کند. به عنوان مثال، فرض کنید که چهار صدا در چهار مکان روی بدنه دارید، مدل باید آن صداها را تجزیه و تحلیل کند، اما در پایان باید تنها یک برچسب (مثلاً وجود) برای هر شناسه موضوعی با امتیاز/احتمال برای همه طبقه‌ها ایجاد کند که اعدادی بین صفر و یک هستند.

خروجی نمونه برای بیمار شناسه ۱۲۳۴ در زیر مشخص است:

این خروجی نشان می‌دهد که مدل، وجود سופل را برای بیمار ۱۲۳۴ شناسایی کرده است

```
#1234
Present, Unknown, Absent, Abnormal, Normal
1, 0, 0 1, 0
0.75, 0.15, 0.1 0.6, 0.4
```

شکل ۲.۱: نمونه خروجی برای بیمار با شناس ۱۲۳۴

و احتمال وجود سوفل را ۷۵ درصد، احتمال عدم وجود سوفل را ۱۰ درصد و احتمال وضعیت ناشناخته سوفل را ۱۵ درصد نشان می‌دهد. همچنین مدل یک پیامد بالینی غیرطبیعی را شناسایی کرده است و احتمال یک نتیجه غیرطبیعی را ۶۰ درصد و احتمال یک نتیجه طبیعی را ۴۰ درصد نشان می‌دهد.

در این پروژه چندین مدل مانند: جنگل تصادفی،^۶ رگرسیون لجستیک^۷ و ماشین بردار پشتیبانی^۸ پیاده سازی شده و دقت نهایی آن در تشخیص برجسب‌های ذکر شده مقایسه می‌شود.

^۶Random Forest

^۷Logestic Regression

^۸Support Vector Machine

فصل ۲

مفاهیم مقدماتی

یادگیری ماشین^۱ یک زیرشاخه از هوش مصنوعی^۲ است که به ماشین‌ها توانایی یادگیری خودکار از داده‌ها و تجربیات گذشته را در حین شناسایی الگوهایی برای پیش‌بینی^۳ با حداقل مداخله انسانی می‌دهد. روش‌های یادگیری ماشین، رایانه‌ها را قادر می‌سازد تا به طور مستقل و بدون برنامه‌نویسی صریح کار کنند. برنامه‌های کاربردی یادگیری ماشینی با داده‌های جدید تغذیه می‌شوند و می‌توانند به طور مستقل یاد بگیرند، رشد کنند و توسعه و تطبیق یابند.

الگوریتم‌های یادگیری ماشین بر پایه یک مجموعه داده آموزشی^۴ برای ایجاد یک مدل^۵ هستند. داده‌های ورودی جدید به الگوریتم یادگیری ماشینی آموزش دیده معرفی می‌شود و می‌توان از آن برای پیش‌بینی در مورد داده‌های جدید استفاده کرد. الگوریتم‌های یادگیری ماشینی را می‌توان به روش‌های مختلفی آموزش داد که هر روشی مزایا و معایب خود را دارد. بر اساس این روش‌ها و روش‌های یادگیری، یادگیری ماشین به طور کلی به چهار نوع اصلی طبقه بندی می‌شود.

¹Machine Learning

²Artificial Intelligence

³Prediction

⁴Training Data

⁵Model

۱.۲ یادگیری با نظارت

یک مسئله یادگیری ماشین معمول، استفاده از یک مدل برای وظیفه پیش‌بینی است. برای انجام این کار، به یک مجموعه داده آموزشی نیاز است که برای آموزش یک مدل استفاده می‌شود. این مجموعه آموزشی باید برای متغیرهای ورودی مثل x ، برچسب‌هایی^۶ مانند y داشته باشد. سپس ورودی‌ها به مدل نشان داده می‌شوند. سپس با استفاده از خروجی مدل، مدل تغییر داده می‌شود به طوری که خروجی‌هایش شبیه تر به خروجی‌های درست مورد انتظار باشد. از مسائل یادگیری با نظارت می‌توان به مسئله طبقه‌بندی^۷ یا رگرسیون^۸ اشاره کرد.

۲.۲ یادگیری بی نظارت

آموزش بی نظارت یک روش دیگر آموزش مدل‌هاست که در آن فقط داده‌های ورودی x را داریم و هیچ برچسبی نداریم. چنین مدلی با استخراج و خلاصه کردن الگوهای موجود در داده‌های ورودی ساخته می‌شود. معمولاً اصلاحی در مدل وجود ندارد زیرا مدل چیزی را پیش‌بینی نمی‌کند. از معروف‌ترین مسائل آموزش بی نظارت می‌توان به مسئله خوشه بندی^۹ و استخراج قوانین انجمنی^{۱۰} اشاره کرد.

۳.۲ یادگیری نیمه نظارتی

یادگیری نیمه نظارتی شامل ویژگی‌های یادگیری ماشین با نظارت و بی نظارت است. این روش از ترکیب مجموعه داده‌های برچسب دار و بدون برچسب برای آموزش الگوریتم‌های خود استفاده می‌کند. معمولاً تهیه حجم زیادی از داده‌های برچسب‌دار وقت‌گیر و هزینه‌بر است و از طرفی داده‌های بدون برچسب، زمانی که همراه با مقدار کمی از داده‌های برچسب‌دار استفاده می‌شوند، می‌توانند بهبود قابل توجهی در دقت یادگیری ایجاد کنند.

⁶Label

⁷Classification

⁸Regression

⁹Clustering

¹⁰Association Rule Mining

۴.۲ یادگیری تقویتی

یادگیری تقویتی یک فرآیند مبتنی بر بازخورد است. در اینجا، عامل هوش مصنوعی^{۱۱} به طور خودکار با روش آمارگیری و آزمایشی، محیط اطراف خود را بررسی می‌کند، اعمالی را اتخاذ می‌کند، از تجربیات می‌آموزد و عملکردش را بهبود می‌بخشد. این عامل هوشمند برای هر عمل خوب پاداش و برای هر حرکت اشتباه جریمه می‌شود. بنابراین، عامل یادگیری تقویتی با انجام اقدامات خوب به حداکثر رساندن پاداش می‌پردازد. یادگیری تقویتی در زمینه‌های مختلفی مانند نظریه بازی‌ها^{۱۲}، نظریه اطلاعات^{۱۳} و سیستم‌های چند عاملی^{۱۴} استفاده می‌شود.

۵.۲ تبدیل فوریه

تبدیل فوریه^{۱۵} یا FT یک تبدیل انتگرالی^{۱۶} است که توابعی^{۱۷} را که بر حسب زمان یا فضا هستند، به توابعی بر حسب فرکانس زمانی یا فضایی تجزیه می‌کند، مانند بیان یک آکورد^{۱۸} موسیقی بر حسب حجم‌ها و فرکانس‌های نت‌های تشکیل دهنده آن. اصطلاح تبدیل فوریه هم به نمایش دامنه فرکانس^{۱۹} و هم به عملیات ریاضی^{۲۰} مربوط به آن که نمایش دامنه فرکانس را به تابعی از مکان یا زمان مرتبط می‌کند، اطلاق می‌شود. همانطور که پیشتر گفته شد، تبدیل فوریه یک تبدیل انتگرالی است که هر تابع $f(t)$ را به یک تابع دیگر $F(\omega)$ منعکس می‌کند. در این صورت، به $F(\omega)$ تبدیل فوریه^{۲۱} تابع $f(t)$ می‌گویند. حالت خاص تبدیل فوریه، سری فوریه^{۲۱} نام دارد و آن زمانی کاربرد دارد که تابع $f(t)$ متناوب^{۲۲} باشد، یعنی: $f(t+T) = f(t)$. چنانچه تابع متناوب نباشد یا به عبارتی، تناوب آن برابر بی‌نهایت باشد ($T \rightarrow \infty$)، از سری فوریه عبارت زیر به دست

¹¹AI Agent

¹²Game Theory

¹³Information Theory

¹⁴Multi-Agent Systems

¹⁵Fourier Transform

¹⁶Integral Transform

¹⁷Function

¹⁸Chord

¹⁹Frequency Domain

²⁰Operation

²¹Fourier Series

²²Periodic Function

می‌آید:

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega)e^{i\omega t} d\omega$$

۶.۲ ضرایب کپسترال مل^{۲۳}

متداولترین ویژگی‌ها برای تشخیص گفتار^{۲۴}، ضرایب فرکانسی مل کپستروم (MFCC) هستند. ایده اصلی در استخراج ضرایب (MFCC) خاصیت گوش انسان در دریافت و فهم گفتار است و همین مسئله این ضرایب را به ابزاری قدرتمند در تمامی حوزه‌هایی پردازش و بازساخت گفتار تبدیل کرده است. تعداد ضرایب مورد استفاده در بازشناسی گفتار معمولاً بین ۹ تا ۱۳ ضریب تغییر می‌کند. ضریب صفرم نشان‌دهنده انرژی است که از آن بعنوان ویژگی شیمر^{۲۵} یاد می‌شود. برای استخراج این ضرایب در مرحله اول، تبدیل فوریه بر سیگنال اعمال می‌شود و سپس توان طیف^{۲۶} به دست آمده به مقیاس مل نگاشته می‌شود و از توان در هر فرکانس مل، لگاریتم گرفته می‌شود. در مرحله آخر طیف لگاریتمی مل به حوزه زمان برگردانده می‌شود. نتیجه این تبدیلات نمایش کپسترال طیف سیگنال گفتار است که ویژگی‌های طیفی یک فریم از سیگنال گفتار را به خوبی نشان می‌دهد.

²³Mel Frequency Cepstral Coefficients

²⁴Speech Recognition

²⁵Shimmer

²⁶Power Spectrum

فصل ۳

مدل‌های ابتدایی

۱.۳ جنگل تصادفی

جنگل تصادفی یا جنگل‌های تصمیم تصادفی یک روش یادگیری ترکیبی^۱ برای طبقه‌بندی و یا رگرسیون می‌باشد، که بر اساس ساختاری متشکل از شمار بسیاری درخت تصمیم^۲ روی زمان آموزش و خروجی طبقه‌ها (طبقه‌بندی) یا برای پیش‌بینی‌های هر درخت به شکل مجزا، کار می‌کنند. جنگل‌های تصادفی برای درختان تصمیم که در مجموعه آموزشی دچار بیش‌برازش می‌شوند، مناسب هستند. عملکرد جنگل تصادفی معمولاً بهتر از درخت تصمیم است، اما این بهبود عملکرد تا حدی به نوع داده هم بستگی دارد. [۹] [۱۰] [۱۱]

۱.۱.۳ آشنایی با الگوریتم جنگل تصادفی

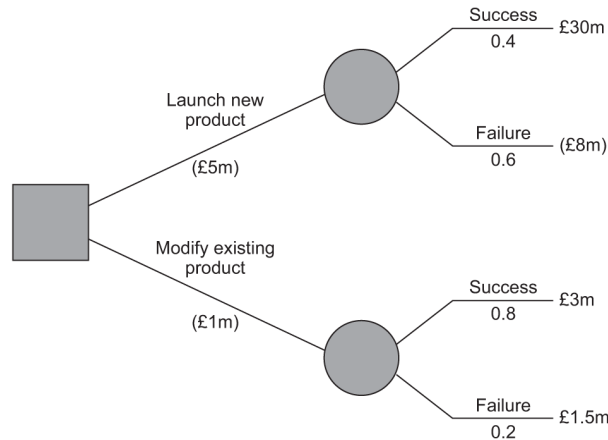
درک الگوریتم جنگل تصادفی بدون شناخت از درخت تصمیم ممکن نیست. به این منظور در ابتدای بحث به بررسی چگونگی کارکرد درخت تصمیم می‌پردازیم.

¹Ensemble Learning

²Decision Tree

۲.۱.۳ درخت تصمیم

درخت تصمیم روشی است که به طور معمول در داده کاوی^۳ از آن استفاده می‌شود. هدف این مدل این است که بتواند مقدار یک متغیر هدف را براساس مقادیر متغیرهای ورودی پیش‌بینی کند.



شکل ۱.۳: نمونه‌ای از یک درخت تصمیم [۱۲]

در اینجا فرض کنید که مقادیر هر ویژگی ورودی‌ها، دارای دامنه گسسته و محدود باشد. و یک ویژگی هدف به نام «طبقه‌بند»^۴ وجود دارد. به هر عضو در دامنه طبقه‌بند یک طبقه^۵ گفته می‌شود. درخت تصمیم یا درخت طبقه‌بندی، درختی است که به هر گره داخلی (گره‌ای که برگ نیست) یکی از ویژگی‌های ورودی‌ها برحسب زده شده‌است. یال‌های خارج شده از گره داخلی به یک گره برگ برای دسته‌بندی داده‌ها برحسب یک ویژگی دیگر هدایت می‌شوند. در نهایت، هر گره برگ برحسب یکی از مقادیر در دامنه طبقه‌بند برچسب زده می‌شود و همه داده‌هایی که در آن گره قرار بگیرند، به عنوان عضو آن طبقه پیش‌بینی خواهند شد.

روند ایجاد درخت تصمیم از تقسیم کردن مجموعه تمام داده‌های در دسترس (که در واقع ریشه درخت را تشکیل می‌دهند)، به زیر مجموعه‌هایی (که به هر گره فرزند تبدیل می‌شوند) است. شرایط تقسیم کردن داده‌های هر گره به گره‌های فرزند آن براساس یک شرط تقسیم‌بندی در هر گره داخلی خواهد بود. تقسیم کردن هر گره بدست آمده به گره‌های فرزند آن در یک روند بازگشتی انجام

³Data Mining

⁴Classifier

⁵Class

خواهد گرفت که به آن تقسیم‌بندی بازگشتی^۶ نیز گفته خواهد شد. این تقسیم‌بندی تا زمانی ادامه پیدا خواهد کرد که داده‌های در هر گره برگ متعلق به یک طبقه باشند یا زمانی که دیگر ویژگی‌ای برای تقسیم کردن داده‌های در یک گره وجود نداشته باشد. به این فرایند القای درخت تصمیم از بالا به پایین [۱۳] گفته می‌شود که یک مثال از الگوریتمی حریصانه^۷ است.

۳.۱.۳ برتری جنگل تصادفی بر درخت تصمیم

در کل، معمولاً درخت تصمیمی که بیش از حد عمیق باشد الگوی دقیقی نخواهد داشت. به این علت که دچار بیش برارزش شده^۸ و دارای سوگیری^۹ پایین و واریانس^{۱۰} بالا می‌باشد. جنگل تصادفی روشی است برای میانگین‌گیری با هدف کاهش واریانس با استفاده از درخت‌های تصمیم عمیقی که از قسمت‌های مختلف داده آموزشی ایجاد شده باشند. در این روش معمولاً افزایش جزئی سوگیری و از دست رفتن کمی از قابلیت تفسیر^{۱۱} اتفاق افتاده اما در کل عملکرد مدل را بسیار افزایش خواهد داد. [۹] [۱۰]

به علاوه به کمک انحراف معیار^{۱۲} پیش‌بینی‌ها از درخت‌های رگرسیون مستقل می‌توان تخمینی از عدم قطعیت پیش‌بینی داشت:

$$\sigma = \sqrt{\frac{\sum_{b=1}^B (f_b(x') - \hat{f})^2}{B - 1}}$$

تعداد نمونه‌ها در فرمول بالا (B) یک متغیر آزاد است که معمولاً به کمک روش‌هایی مانند اعتبارسنجی متقابل^{۱۳} مقدار بهینه را برای فرمول بالا می‌توان به دست آورد.

⁶Recursive Partitioning

⁷Greedy Algorithm

⁸Overfit

⁹Bias

¹⁰Variance

¹¹Interpretability

¹²Standard Deviation

¹³Cross Validation

۴.۱.۳ کیسه‌گذاری درختان^{۱۴}

مجموعه داده را با D نمایش می‌دهیم، $D = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ و B درخت تصادفی با ایجاد B داده جدید از D ایجاد می‌کنیم. مدل نهایی با میانگین گرفتن یا رأی‌گیری بین درختان کار می‌کند [۱۴]. جزئیات این الگوریتم در زیر آمده است:

برای B تا $b = 1$:

- نمونه n نمونه با جایگزینی از داده D انتخاب شود و این نمونه‌ها در مجموعه داده D_b قرار داده می‌شود. از آنجا که نمونه‌گیری با جایگزینی صورت می‌گیرد یک نمونه ممکن است چندین بار انتخاب شود.

- یک درخت تصادفی به اسم T_b و D_b به روش پایین ساخته می‌شود:

هر دفعه برای پیدا کردن بهترین متغیر ابتدا یک تعداد مشخصی از متغیرها را کاملاً به صورت تصادفی انتخاب شود (مثلاً m تا m از قبل به مسئله داده شده است، و معمولاً با جذر تعداد متغیرها برابر است) و از میان آن‌ها بهترین متغیر انتخاب می‌شود.

در مسئله رگرسیون، مدل نهایی میانگین تمامی درخت‌ها است [۱۴] یعنی $F(x) = \frac{1}{B} \sum_{b=1}^B T_b(x)$. از طرفی دیگر در مسئله طبقه‌بندی، با استفاده از رأی‌گیری بین درختان به جواب نهایی می‌رسیم [۱۴]. این نوع ترکیب مدل‌ها جواب بهتری به ما می‌دهد، زیرا گوناگونی و تنوع مدل‌ها را افزایش می‌دهد بدون این که سوگیری را افزایش دهد؛ این بدین معناست که زمانی که پیش‌بینی تکی از یک درخت دارای نویز^{۱۵} بالایی باشد، در میانگین بسیاری از درخت‌ها این نویز وجود نخواهد داشت. به شکل ساده آموزش درختان به صورت تکی می‌تواند درخت‌هایی با ارتباط قوی تر ارائه دهد.

۲.۳ رگرسیون لجستیک

رگرسیون لجستیک یک مدل آماری رگرسیون برای متغیرهای وابسته دوسویی مانند بیماری یا سلامت، مرگ یا زندگی است. این مدل را می‌توان به عنوان مدل خطی تعمیم‌یافته‌ای که از تابع لجوجیت^{۱۶} به عنوان تابع پیوند^{۱۷} استفاده می‌کند و خطایش از توزیع چندجمله‌ای پیروی می‌کند،

¹⁴Bagging

¹⁵Noise

¹⁶Logit

¹⁷Link Function

به حساب آورد. رگرسیون لجستیک می‌تواند یک مورد خاص از مدل خطی عمومی و رگرسیون خطی^{۱۸} دیده شود. مدل رگرسیون لجستیک، بر اساس فرض‌های کاملاً متفاوتی (درباره رابطه متغیرهای وابسته و مستقل) از رگرسیون خطی است. تفاوت مهم این دو مدل در دو ویژگی رگرسیون لجستیک می‌تواند دیده شود. اول توزیع شرطی $y|\vec{x}$ یک توزیع برنولی^{۱۹} به جای یک توزیع گوسی^{۲۰} است چونکه متغیر وابسته دودویی^{۲۱} است. دوم مقادیر پیش‌بینی احتمالاتی است و محدود بین بازه صفر و یک و به کمک تابع توزیع لجستیک بدست می‌آید. رگرسیون لجستیک احتمال خروجی را پیش‌بینی می‌کند. این مدل به صورت

$$\text{logit}(p) = \ln \left(\frac{p}{1-p} \right) = \beta_0 + \beta_1 x_{1,i} + \dots + \beta_k x_{k,i}$$

$$i = 1, \dots, n,$$

است که

$$p = \Pr(y_i = 1)$$

$$p = \Pr(y_i = 1 | \vec{x}_i; \vec{\beta}) = \frac{e^{\beta_0 + \beta_1 x_{1,i} + \dots + \beta_k x_{k,i}}}{1 + e^{\beta_0 + \beta_1 x_{1,i} + \dots + \beta_k x_{k,i}}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{1,i} + \dots + \beta_k x_{k,i})}}$$

رگرسیون لجستیک را می‌توان توسط تابع لجستیک تعریف کرد. دامنه این تابع اعداد حقیقی هستند و برد این تابع بین صفر و یک می‌باشد. [۱۵] این تابع با $\sigma : \mathbb{R} \rightarrow (0, 1)$ نمایش داده می‌شود و به نحو پایین محاسبه می‌شود: [۱۵]

$$\sigma = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

¹⁸Linear Regression

¹⁹Bernoulli Distribution

²⁰Gaussian Distribution

²¹Binary

با احتساب تابع لجستیک، رگرسیون لجستیک را می‌توان به شکل پایین بازنویسی کرد: [۱۵]

$$\Pr(y_i = 1 | \vec{x}_i; \vec{\beta}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{1,i} + \dots + \beta_k x_{k,i})}} = \sigma(\beta_0 + \beta_1 x_{1,i} + \dots + \beta_k x_{k,i})$$

۱.۲.۳ برآورد پارامترهای بهینه

برای بدست آوردن پارامترهای بهینه یعنی $\vec{\beta} = [\beta_0, \beta_1, \dots, \beta_k]$ می‌توان از روش برآورد درست نمایی بیشینه^{۲۲} استفاده کرد. اگر فرض کنیم که تعداد مثال‌هایی که قرار است برای تخمین پارامترها استفاده کنیم n است و این مثال‌ها را به این شکل نمایش دهیم $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$ ، $D = (\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$ پارامتر بهینه پارامتری است که برآورد درست نمایی را بیشینه کند، البته برای سادگی کار برآورد لگاریتم درست نمایی را بیشینه می‌کنیم. لگاریتم درست نمایی داده D برای پارامتر $\vec{\beta}$ را با $L(D, \vec{\beta})$ نمایش می‌دهیم:

$$\begin{aligned} L(D, \vec{\beta}) &= \log \left(\prod_{i=1}^n \Pr(y_i = 1 | \vec{x}_i; \vec{\beta})^{y_i} \times \Pr(y_i = 0 | \vec{x}_i; \vec{\beta})^{1-y_i} \right) \\ &= \sum_{i=1}^n y_i \times \log \Pr(y_i = 1 | \vec{x}_i; \vec{\beta}) + (1 - y_i) \log \Pr(y_i = 0 | \vec{x}_i; \vec{\beta}) \end{aligned}$$

اگر برای داده i ام، $y_i = 1$ باشد، هدف افزایش $\Pr(y_i = 1 | \vec{x}_i; \vec{\beta})$ است و اگر y_i صفر باشد هدف افزایش مقدار $\Pr(y_i = 0 | \vec{x}_i; \vec{\beta})$ است. به همین علت از فرمول زیر استفاده می‌کنیم:

$$\Pr(y_i = 1 | \vec{x}_i; \vec{\beta})^{y_i} \times \Pr(y_i = 0 | \vec{x}_i; \vec{\beta})^{1-y_i}$$

اگر $y_i = 1$ فرمول به ما $\Pr(y_i = 1 | \vec{x}_i; \vec{\beta})$ را بدهد و اگر $y_i = 0$ بود به ما $\Pr(y_i = 0 | \vec{x}_i; \vec{\beta})$ را بدهد.

حال برای بدست آوردن پارامتر بهینه باید $\vec{\beta}$ ای پیدا کنیم که مقدار $L(D, \vec{\beta})$ را بیشینه کند. از آنجا که این تابع نسبت به $\vec{\beta}$ مقعر است حتماً یک بیشینه مطلق دارد. برای پیدا کردن جواب می‌توان

²²Maximum Likelihood Estimation

از روش گرادیان افزایشی تصادفی^{۲۳} استفاده کرد. در این روش هر بار یک مثال را به صورت اتفاقی از نمونه‌های داده انتخاب کرده، گرادیان درست‌نمایی را حساب می‌کنیم و کمی در جهت گرادیان پارامتر را حرکت می‌دهیم تا به یک پارامتر جدید برسیم. گرادیان جهت موضعی بیشترین افزایش را در تابع به ما نشان می‌دهد، برای همین در آن جهت کمی حرکت می‌کنیم تا به بیشترین افزایش موضعی تابع برسیم. اینکار را آنقدر ادامه می‌دهیم که گرادیان به اندازه کافی به صفر نزدیک شود. به جای اینکه داده‌ها را به صورت تصادفی انتخاب کنیم می‌توانیم به ترتیب داده شماره 1 تا داده شماره n را انتخاب کنیم و بعد دوباره به داده اولی برگردیم و این کار را به صورت متناوب چندین بار انجام دهیم تا به اندازه کافی گرادیان به صفر نزدیک شود. از لحاظ ریاضی این کار را می‌توان به شکل پایین انجام داد:

پارامتر $\vec{\beta}$ را در ابتدا به صورت تصادفی مقداردهی می‌کنیم و بعد برای داده i ام و تمامی j ها، یعنی از $j = 1$ تا $j = m + 1$ تغییر پایین را اعمال می‌کنیم، در اینجا α همان مقداریست که در جهت گرادیان هر بار حرکت می‌کنیم و $\vec{x}_{i,j}$ مشتق جزئی داده i ام در بعد j ام است:

$$\left\{ \begin{array}{l} \text{Initialize } \vec{\beta}^{\text{old}} \text{ randomly} \\ \text{loop until convergence :} \\ \text{for } i = 0 \text{ to } n : \\ \quad \text{for } j = 0 \text{ to } m : \\ \qquad \vec{\beta}_j^{\text{new}} = \vec{\beta}_j^{\text{old}} + \alpha \left(y_i - \frac{1}{1 + e^{-(\beta_0^{\text{old}} + \beta_1^{\text{old}} x_{1,i} + \dots + \beta_k^{\text{old}} x_{k,i})}} \right) \vec{x}_{i,j} \\ \qquad \beta^{\text{old}} = \beta^{\text{new}} \end{array} \right.$$

²³Stochastic Gradient Ascent

۲.۲.۳ تنظیم مدل^{۲۴}

پیچیدگی مدل‌های پارامتری با تعداد پارامترهای مدل و مقادیر آن‌ها سنجیده می‌شود. هرچه این پیچیدگی بیشتر باشد خطر بیش‌برازش^{۲۵} برای مدل بیشتر است [۱۶]. پدیده بیش‌برازش زمانی رخ می‌دهد که مدل به‌جای یادگیری الگوهای داده، داده را حفظ کند و در عمل، فرایند یادگیری به خوبی انجام نمی‌شود. برای جلوگیری از بیش‌برازش در مدل‌های خطی مانند رگرسیون خطی یا رگرسیون لجستیک جریمه‌ای به تابع هزینه^{۲۶} اضافه می‌شود تا از افزایش زیاد پارامترها جلوگیری شود. تابع هزینه را در رگرسیون لجستیک با منفی لگاریتم درست‌نمایی تعریف می‌کنیم تا کمینه کردن آن به بیشینه کردن تابع درست‌نمایی بیانجامد. به این کار تنظیم مدل گفته می‌شود. دو راه متداول تنظیم مدل‌های خطی روش‌های L_1 و L_2 هستند. [۱۷] در روش L_1 ضریبی از نرم L_1 ^{۲۷} به تابع هزینه اضافه می‌شود و در روش L_2 ضریبی از نرم L_2 که همان نرم اقلیدسی^{۲۸} است به تابع هزینه اضافه می‌شود. در تنظیم مدل به روش L_1 تابع هزینه را به این شکل تغییر می‌دهیم: [۱۸]

$$L_r(D, \vec{\beta}) = -L(D, \vec{\beta}) + \lambda \|\vec{\beta}\|_1$$

$$= - \sum_{i=1}^n y_i \times \log \Pr(y_i = 1 | \vec{x}_i; \vec{\beta}) + (1 - y_i) \log \Pr(y_i = 0 | \vec{x}_i; \vec{\beta}) + \lambda \sum_{k=0}^m |\beta_k|$$

این روش تنظیم مدل که به روش لاسو^{۲۹} نیز شهرت دارد، باعث می‌شود که بسیاری از پارامترهای مدل نهایی صفر شوند و مدل به اصطلاح خلوت^{۳۰} شود. [۱۹] در تنظیم مدل به روش L_2 تابع هزینه را به این شکل تغییر می‌دهیم:

$$L_r(D, \vec{\beta}) = L(D, \vec{\beta}) + \lambda \|\vec{\beta}\|_2^2$$

$$= - \sum_{i=1}^n y_i \times \log \Pr(y_i = 1 | \vec{x}_i; \vec{\beta}) + (1 - y_i) \log \Pr(y_i = 0 | \vec{x}_i; \vec{\beta}) + \lambda \sum_{k=0}^m \beta_k^2$$

²⁴Regularization

²⁵Overfitting

²⁶Loss Function

²⁷Norm

²⁸Euclidean Norm

²⁹Lasso

³⁰Sparse

در روش تنظیم از طریق L_2 سعی می‌شود فاصله اقلیدسی^{۳۱} بردار $\vec{\beta}$ کوتاه نگه داشته شود. λ در روش‌های L_1 و L_2 یک عدد مثبت است که میزان تنظیم مدل را معین می‌کند. هرچه قدر λ کوچکتر باشد جریمه کمتری برای بزرگی نرم بردار پارامترها یعنی $\vec{\beta}$ پرداخت می‌کنیم. مقدار ایده‌آل^{۳۲} λ از طریق آزمایش بر روی داده اعتبار^{۳۳} پیدا می‌شود.

۳.۳ ماشین بردار پشتیبان

ماشین بردار پشتیبان (SVM) یکی از روش‌های یادگیری بانظارت [۲۰] است که از آن برای طبقه‌بندی [۲۱] و رگرسیون [۲۲] استفاده می‌کنند. مبنای کاری طبقه‌بند، طبقه‌بندی خطی داده‌ها^{۳۴} است. در تقسیم خطی داده‌ها سعی می‌کنیم ابرصفحه‌ای^{۳۵} را انتخاب کنیم که حاشیه اطمینان^{۳۶} بیشتری داشته باشد. حل معادله پیدا کردن خط بهینه^{۳۷} برای داده‌ها به وسیله روش‌های برنامه‌نویسی غیرخطی^{۳۸} که روش‌های شناخته شده‌ای در حل مسائل محدودیت‌دار، هستند صورت می‌گیرد. قبل از تقسیم خطی برای اینکه ماشین بتواند داده‌های با پیچیدگی بالا را دسته‌بندی کند داده‌ها را به وسیله تابع ϕ به فضای با ابعاد خیلی بالاتر^{۳۹} [۲۳] می‌بریم. برای اینکه بتوانیم مسئله ابعاد خیلی بالا را با استفاده از این روش‌ها حل کنیم از قضیه دوگانی لاگرانژ^{۴۰} [۲۴] برای تبدیل مسئله بهینه‌سازی^{۴۱} مورد نظر به فرم دوگانی^{۴۲} آن که در آن به جای تابع پیچیده ϕ که ما را به فضایی با ابعاد بالا می‌برد، تابع ساده‌تری به نام تابع هسته^{۴۳} که ضرب برداری تابع ϕ است ظاهر می‌شود، استفاده می‌کنیم. از توابع هسته مختلفی از جمله هسته‌های نمایی، چندجمله‌ای^{۴۴} و سیگموئید^{۴۵}

³¹ Euclidean distance

³² Ideal

³³ Validation Data

³⁴ Linear Data Classification

³⁵ Hyperplane

³⁶ Safety Margin

³⁷ Optimal Line

³⁸ Nonlinear Programming

³⁹ Higher Dimensional Space

⁴⁰ Lagrangian Duality Theory

⁴¹ Optimization

⁴² Duality

⁴³ Kernel

⁴⁴ Polynomial

⁴⁵ Sigmoid

می توان استفاده نمود.

۱.۳.۳ ماشین بردار پشتیبان خطی

ما مجموعه داده‌های آزمایش D ^{۴۶} شامل n عضو را در اختیار داریم که به صورت زیر تعریف می‌شود:

$$D = \{(x_i, y_i) \mid x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n$$

جایی که مقدار y برابر 1 یا -1 و هر x_i یک بردار حقیقی p -بعدی است. هدف پیدا کردن ابرصفحه جداکننده با بیشترین فاصله از نقاط حاشیه‌ای است که نقاط با $y_i = 1$ را از نقاط با $y_i = -1$ جدا کند. هر ابرصفحه می‌تواند به صورت مجموعه‌ای از نقاط x که شرط زیر را ارضا می‌کنند نوشته شود:

$$w \cdot x - b = 0$$

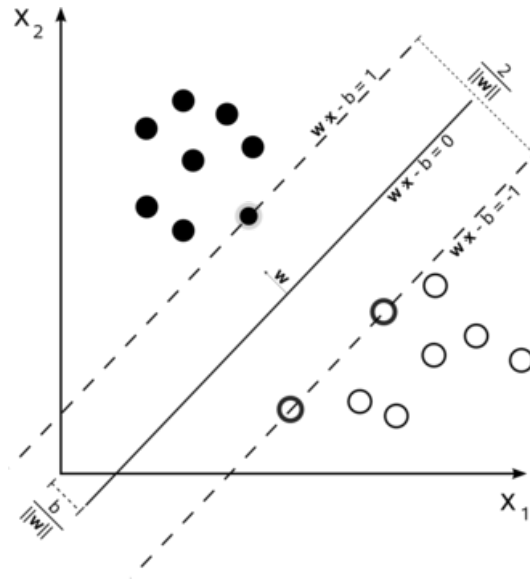
جایی که \cdot علامت ضرب و w بردار نرمال است، که به ابرصفحه عمود است. ما می‌خواهیم w و b را طوری انتخاب کنیم که بیشترین فاصله بین ابرصفحه‌های موازی که داده‌ها را از هم جدا می‌کنند، ایجاد شود. این ابرصفحه‌ها با استفاده از رابطه زیر توصیف می‌شوند. هر داده‌ای بالای ابرصفحه جداکننده با برچسب 1 مشخص می‌شود:

$$w \cdot x - b = 1$$

و هر داده‌ای پایین ابرصفحه جداکننده با برچسب -1 مشخص می‌شود:

$$w \cdot x - b = -1$$

⁴⁶Test Data



شکل ۲.۳: ابرصفحه‌ای با حداکثر حاشیه برای یک ماشین بردار پشتیبانی که با نمونه داده‌هایی از دو دسته آموزش داده شده است. [۲۵]

۲.۳.۳ حاشیه سخت^{۴۷}

اگر داده‌های آموزشی جدایی پذیر خطی^{۴۸} باشند، ما می‌توانیم دو ابر صفحه در حاشیه نقاط به طوری که هیچ نقطه مشترکی نداشته باشند، در نظر بگیریم و سپس سعی کنیم، فاصله آن‌ها را حداکثر کنیم. با استفاده از هندسه، فاصله این دو صفحه $\frac{2}{\|w\|}$ است؛ بنابراین ما باید $\|w\|$ را کمینه^{۴۹} کنیم. برای اینکه از ورود نقاط به حاشیه جلوگیری کنیم، شرایط زیر را اضافه می‌کنیم: برای هر i می‌تواند به صورت زیر نوشته شود:

$$w \cdot x_i - b \geq 1, \text{ if } y_i = 1$$

$$w \cdot x_i - b \leq -1, \text{ if } y_i = -1$$

$$y_i (w \cdot x_i - b) \geq 1, \quad \forall 1 \leq i \leq n$$

⁴⁷Hard-Margin

⁴⁸Linearly Separable

⁴⁹Minimum

با کنار هم قرار دادن این دو یک مسئله بهینه‌سازی به دست می‌آید:

$$\begin{aligned} & \min_{(w,b)} \|\mathbf{w}\| \\ & \text{s.t. } \forall 1 \leq i \leq n \\ & y_i (\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1. \end{aligned}$$

فرم اولیه

مسئله بهینه‌سازی مشاهده شده در قسمت قبل، مسئله سختی برای حل کردن است، زیرا به $\|\mathbf{w}\|$ وابسته است. خوشبختانه می‌توانیم، بدون تغییر در مسئله $\|\mathbf{w}\|$ را با $\frac{1}{2}\|\mathbf{w}\|^2$ جانشین می‌کنیم. به‌طور واضح تر:

$$\begin{aligned} & \min_{(w,b)} \frac{1}{2}\|\mathbf{w}\|^2 \\ & \text{s.t. } \forall 1 \leq i \leq n \\ & y_i (\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1. \end{aligned}$$

می‌توان عبارت قبل را با استفاده از ضرایب نامنفی لاگرانژ به صورت زیر نوشت که در آن α_i ضرایب لاگرانژ^{۵۰} هستند:

$$\min_{\mathbf{w}, b, \alpha} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i - b) - 1] \right\}$$

اما فرمول فوق اشتباه است. فرض کنید ما بتوانیم خانواده‌ای از ابر صفحات که نقاط را تقسیم می‌کنند، پیدا کنیم. پس همه $y_i (\mathbf{w} \cdot \mathbf{x}_i - b) - 1 \geq 0$ خواهند شد. بنابراین ما می‌توانیم کمینه را با فرستادن همه α_i به $+\infty$ پیدا کنیم. با این حال شرط پیش گفته می‌تواند به صورت پایین بیان شود:

$$\min_{\mathbf{w}, b} \max_{\alpha} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i - b) - 1] \right\}$$

ما به دنبال نقاط زینی^{۵۱} می‌گردیم. حالا می‌توان این مسئله را به کمک برنامه‌نویسی غیرخطی

⁵⁰Lagrange Multiplier

⁵¹Saddle Points

استاندارد حل کرد. جواب می‌تواند به صورت ترکیب خطی α_i از بردارهای آموزشی \mathbf{x}_i بیان شود:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$

تنها چند α_i بزرگتر از صفر خواهد بود. \mathbf{x}_i متناظر، دقیقاً همان بردار پشتیبان خواهد بود و شرط را ارضا خواهد کرد. از این رو می‌توان نتیجه گرفت که بردارهای پشتیبان شرط زیر را نیز ارضا می‌کنند:

$$y_i (\mathbf{w} \cdot \mathbf{x}_i - b) = 1$$

که اجازه می‌دهد مقدار b تعریف شود. در عمل الگوریتم مقاوم تر خواهد بود اگر از تمام N_{SV} بردار پشتیبان میانگین گرفته شود:

$$b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (\mathbf{w} \cdot \mathbf{x}_i - y_i)$$

فرم دوگان

استفاده از این واقعیت که $\|\mathbf{w}\|^2 = \mathbf{w} \cdot \mathbf{w}$ و جانشینی $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ می‌توان نشان داد که دوگان (SVM) به مسئله بهینه‌سازی زیر ساده می‌شود:

$$\max_{\forall \alpha_i} \left(\tilde{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \right)$$

$$\text{s.t. } \forall i, \alpha_i \geq 0, \text{ and } \sum_{i=1}^n \alpha_i y_i = 0$$

در اینجا هسته به صورت $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ تعریف می‌شود. عبارت α تشکیل یک دوگان

⁵²Linear Combination

⁵³Training Vectors

برای بردار وزن‌ها مجموعه آموزشی می‌دهد:

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

۳.۳.۳ حاشیه نرم^{۵۴}

استفاده از روش سخت می‌تواند در مواردی محدودیت‌هایی ایجاد کند. یکی از آنها وابستگی شدید جداکننده به داده‌های مرزی است و اگر داده‌ای در مرز ابرصفحه جداکننده قرار گیرد، منجر به کاهش حاشیه و تغییر شدید در ابرصفحه مورد نظر می‌شود. این نکته حائز اهمیت است که داده‌های ما در واقعیت در اکثر موارد مقداری نویز دارند و بنابراین مشکل بالا می‌تواند اثر نامطلوبی روی مرز تصمیم‌گیری^{۵۵} ما داشته باشد؛ بنابراین مدل حاشیه سخت به شدت قابلیت بیش‌برازش شدن دارد.

برای جلوگیری و حل این مشکل مدل حاشیه نرم ارائه شد، که اجازه می‌دهد در فرایند تمرین مدل تعدادی از داده‌ها به درستی طبقه‌بندی نشوند و حاشیه تعیین شده را نقض کنند تا از بیش‌برازش در هنگام آزمون جلوگیری کنیم. به عبارتی با اینکار نوعی عمومیت^{۵۶} به مدل داده می‌شود تا بر روی داده‌های آزمون عملکرد بهتری داشته باشد. البته که اینکار با افزودن یک سری پارامتر و به صورت کنترل شده انجام می‌شود تا دقت مدل پایین نیاید. به این متغیرها، متغیر کمکی^{۵۷} گفته می‌شود و با نماد ξ نمایش داده می‌شوند. این متغیرها برای هر داده میزان تخطی آن از مرز تعیین شده را مشخص می‌کند اگر مقدار آنها صفر باشد به این معنی است که داده در سمت درستی از مرز قرار گرفته و اگر مقدار آنها بزرگتر از صفر باشد مقدار خطا را نشان می‌دهند.

در ادامه همه ξ ها جمع زده می‌شوند و در ضریب C ضرب می‌شوند و به عبارتی که در حاشیه سخت داشتیم اضافه می‌شوند تا عبارت خطای جدید به دست آید و ما برای رسیدن به هدف خود می‌بایست

⁵⁴Soft-Margin

⁵⁵Decision Boundary

⁵⁶Generalization

⁵⁷Slack Variable

عبارت زیر را کمینه کنیم:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^N \xi_i$$

$$\text{s.t. } y^{(i)} (w^T x^{(i)} + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \quad \forall i \in \{1, 2, \dots, N\}$$

در عبارت خطا، مقدار C نشان می‌دهد که چقدر اجازه می‌دهیم در زمان آموزش مدل، داده‌ها در دسته اشتباه قرار گیرند. این یعنی هرچقدر C بزرگتر باشد کمتر به مدل اجازه می‌دهیم تا اشتباه کند و بالعکس. کاهش بی‌رویه و بدون کنترل C موجب افزایش سوگیری و در نتیجه افزایش واریانس خواهد شد، پس باید در انتخاب C بسیار دقت کرد. به C متغیر تنظیم^{۵۸} گفته می‌شود.

۴.۳.۳ ماشین بردار پشتیبان غیرخطی

در سال ۱۹۹۲، ایزابل گیون^{۵۹} و برنهارد بوزر^{۶۰} راهی را برای ایجاد دسته‌بند غیرخطی، با استفاده قرار دادن هسته برای پیدا کردن ابرصفحه با بیشترین حاشیه، پیشنهاد دادند. [۲۶] الگوریتم پیشنهادی ظاهراً مشابه است، به جز آنکه تمام ضرب‌های نقطه‌ای با یک تابع هسته غیرخطی جایگزین شدند. این خصوصیت اجازه می‌دهد، الگوریتم، برای ابرصفحه با بیشترین حاشیه در یک فضای ویژگی تغییرشکل داده، مناسب باشد. ممکن است، تغییرشکل غیرخطی باشد و فضای تغییر یافته، دارای ابعاد بالاتری باشد. به هر حال دسته‌بندی‌کننده، یک ابرصفحه در فضای ویژگی با ابعاد بالا است که ممکن است در فضای ورودی نیز غیرخطی باشد. اگر از هسته با تابع گوسی^{۶۱} استفاده شود، فضای ویژگی متناظر، یک فضای هیلبرت^{۶۲} نامتناهی است. دسته‌کننده^{۶۳} بیشترین حاشیه، خوش ترتیب^{۶۳} است، بنابراین ابعاد نامتناهی نتیجه را خراب نمی‌کند. هسته‌های متداول به صورت زیر هستند:

• چندجمله‌ای همگن:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$$

⁵⁸Regularization Parameter

⁵⁹Isabelle Guyon

⁶⁰Bernhard Boser

⁶¹Gaussian Function

⁶²Hilbert Space

⁶³Well-Order

• چندجمله‌ای ناهمگن:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$$

هسته با انتقال $\varphi(\mathbf{x}_i)$ با تساوی $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$ در ارتباط است. همچنین مقدار w فضای انتقال یافته برابر $w = \sum_i \alpha_i y_i \varphi(\mathbf{x}_i)$ می‌باشد. ضرب نقطه‌ای با w می‌تواند توسط هسته محاسبه شود یعنی:

$$w \cdot \varphi(\mathbf{x}) = \sum_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{x})$$

۵.۳.۳ طبقه‌بند بردار پشتیبان چندطبقه^{۶۴}

ماشین بردار پشتیبان اساساً یک جداکننده است. در بخش قبلی پایه‌های تئوری ماشین‌های بردار پشتیبان برای دسته‌بندی دو طبقه تشریح شد. یک تشخیص الگوی چند طبقه‌ای می‌تواند به وسیله ترکیب ماشین‌های بردار پشتیبان دو طبقه‌ای حاصل شود. به‌طور معمول دو دید برای این هدف وجود دارد. یکی از آن‌ها استراتژی «یک در مقابل همه»^{۶۵} برای دسته‌بندی هر جفت طبقه و طبقه‌های باقی‌مانده است. دیگر استراتژی «یک در مقابل یک»^{۶۶} برای دسته‌بندی هر جفت است. برای مسائل چند طبقه‌ای، رهیافت کلی کاهش مسئله چند طبقه‌ای به چندین مسئله دودویی است. هر یک از مسائل با یک جداکننده دودویی حل می‌شود. سپس خروجی جداکننده‌های دودویی SVM با هم ترکیب شده و به این ترتیب مسئله چند طبقه حل می‌شود.

روش «یک در مقابل همه»

در این روش، ما نیازمند ابرصفحه‌ای هستیم که داده‌های یک طبقه را از داده‌های سایر طبقه‌ها به‌طور همزمان جداسازی نماید؛ در واقع با این روش تمام نمونه‌ها در بدست آوردن ابرصفحه بهینه در نظر گرفته خواهند شد؛ به طوری که داده‌ها به ازای هر طبقه‌بند به دو دسته تقسیم می‌شوند: یک گروه به ازای داده‌های طبقه مورد نظر و گروه دیگر شامل تمام نقاط سایر طبقه‌ها. بدین ترتیب برای یک مسئله طبقه‌بندی با K دسته، نیازمند K طبقه‌بند ۲ طبقه‌ای هستیم. در زمان آموزش مدل همه K طبقه‌بند را روی مجموعه آموزش، آموزش می‌دهیم و سپس در زمان تست مدل هم بررسی می‌کنیم که

⁶⁴Multi-Class SVC

⁶⁵One-Vs-All

⁶⁶One-Vs-One

هر طبقه‌بند داده ورودی را در طبقه خود می‌داند یا خیر. در شرایطی امکان دارد که چند طبقه‌بند همزمان داده را در طبقه خود اعلام کنند که در اینصورت می‌توان از معیارهایی استفاده کرد، یکی از این معیارها انتخاب طبقه‌بندی است که دورترین ابرصفحه را از داده مورد نظر دارد.

روش «یک در مقابل یک»

«یک در مقابل یک» روش دیگری از الگوریتم‌های طبقه‌بندی دودویی به منظور طبقه‌بندی چند طبقه‌ای می‌باشد و برخلاف رویکرد روش قبلی، مجموعه داده طبقه‌بندی چند طبقه‌ای را به مسائل طبقه‌بندی دودویی تقسیم می‌کند. پس در نتیجه به $\binom{k}{2}$ طبقه‌بند نیاز داریم که K در این رابطه تعداد طبقه‌ها است.

فصل ۴

شبکه‌های عصبی بازگشتی

پیش از بررسی پیاده‌سازی مدل‌ها نیاز است تا با مقدمات شبکه‌های عصبی آشنا شویم. در این بخش به بررسی مفاهیم اولیه شبکه‌های عصبی می‌پردازیم.

۱.۴ شبکه‌های عصبی

شبکه‌های عصبی سیستم‌ها و روش‌های محاسباتی نوین برای یادگیری ماشین، نمایش دانش^۱ و در انتها اعمال دانش به دست آمده در جهت بیش‌بینی پاسخ‌های خروجی از سامانه‌های پیچیده هستند. ایده اصلی این گونه شبکه‌ها تا حدودی الهام‌گرفته از شیوه کارکرد سیستم عصبی زیستی^۲ برای پردازش داده‌ها و اطلاعات به منظور یادگیری و ایجاد دانش می‌باشد.

یک شبکه عصبی، از سه لایه ورودی، خروجی و پردازش تشکیل می‌شود. هر لایه شامل گروهی از سلول‌های عصبی^۳ است که عموماً با کلیه نورون‌های لایه‌های دیگر در ارتباط هستند، مگر این که کاربر ارتباط بین نورون‌ها را محدود کند؛ ولی نورون‌های هر لایه با سایر نورون‌های همان لایه، ارتباطی ندارند.

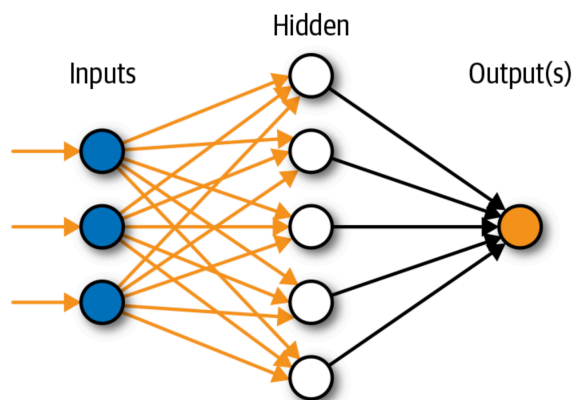
نورون کوچک‌ترین واحد پردازشگر اطلاعات است که اساس عملکرد شبکه‌های عصبی را تشکیل می‌دهد. یک شبکه عصبی مجموعه‌ای از نورون‌هاست که با قرار گرفتن در لایه‌های مختلف،

¹Knowledge

²Nervous System

³Neuron

معماری خاصی را بر مبنای ارتباطات بین نورون‌ها در لایه‌های مختلف تشکیل می‌دهند.



شکل ۱.۴: نمونه‌ای از معماری یک شبکه عصبی [۲۷]

۲.۴ شبکه‌های عصبی بازگشتی (مکرر)^۴

یک شبکه‌ی عصبی مکرر کلاسی از شبکه‌های عصبی مصنوعی هستند که در آن اتصالات مابین گره‌هایی از یک گراف جهت‌دار در امتداد یک دنباله‌ی زمانی می‌باشند و سبب می‌شود تا الگوریتم بتواند رفتار پویای موقتی را به نمایش بگذارد. برخلاف شبکه‌های عصبی پیشخور^۵، شبکه‌های عصبی مکرر می‌توانند از وضعیت درونی خود برای پردازش دنباله‌ی ورودی‌ها استفاده کنند که آن‌ها را برای مواردی نظیر تشخیص صوت، یا تشخیص دست‌نوشته‌های غیربخش‌بندی شده‌ی متصل [۲۸] مناسب می‌کند. [۲۹] [۳۰]

شبکه‌های عصبی مکرر می‌توانند دارای وضعیت‌های ذخیره‌سازی شده باشند و این حافظه‌ی ذخیره‌سازی نیز می‌تواند تحت کنترل مستقیم شبکه‌ی عصبی باشد. همچنین در صورتی که دارای تأخیر زمانی^۶ یا دارای چرخه‌های بازخوردی^۷ باشند، حافظه‌ی مذکور می‌تواند با شبکه‌ی دیگری یا گراف

⁴Recurrent Neural Network

⁵Feed Forward Neural Network

⁶Time Delay

⁷Feedback Loop

دیگری جایگزین شود. یکی از انواع مشهور این نوع شبکه‌ها، شبکه‌های حافظه‌ی کوتاه‌مدت بلند (LSTM) می‌باشد.

۳.۴ حافظه طولانی کوتاه-مدت^۸

حافظه طولانی کوتاه-مدت (LSTM) یک معماری شبکه عصبی بازگشتی است. در شبکه عصبی بازگشتی برخلاف شبکه عصبی پیشخور ورودی می‌تواند به صورت دنباله (مثل صوت یا ویدیو) باشد. این ویژگی باعث شده شبکه‌های عصبی بازگشتی برای پردازش داده‌های زمانی بسیار مناسب باشند زیرا الگوها در داده‌های سری زمانی می‌توانند در فواصل مختلف واقع شوند. برای مثال، شبکه (LSTM) در حوزه‌هایی همچون تشخیص دست خط^۹، ترجمه ماشینی^{۱۰}، تشخیص گفتار، کنترل ربات‌ها و بازی‌های ویدیویی قابل استفاده است.

۴.۴ شبکه‌های عصبی بازگشتی دو طرفه^{۱۱}

شبکه‌های عصبی بازگشتی دوطرفه (BiLSTM) انواعی از شبکه‌های عصبی بازگشتی هستند که دو لایه پنهان در دو جهت مختلف در یک شبکه عصبی را به یک خروجی یکسان وصل می‌کند. این رویکرد یکی از رویکردهای مدل‌های یادگیری عمیق سازنده^{۱۲} است که به این ترتیب خروجی می‌تواند اطلاعات را هم از گذشته و هم از آینده به صورت همزمان دریافت کند. ایده اولیه‌ای که از آن در شبکه‌های عصبی بازگشت پذیر دوطرفه استفاده می‌شود این است که نورون‌های یک شبکه عصبی بازگشت پذیر یک طرفه استاندارد را به دو طرف تقسیم می‌نماییم. به عبارتی یک جهت در زمان در جهت مثبت داریم و یک جهت در زمان در جهت منفی که به اولی حالت‌های پسین^{۱۳} و به دومی حالت‌های پیشین^{۱۴} می‌گوییم. تنها نکته‌ای که باید دقت کنیم این است که خروجی این دو حالت به ورودی در حالت‌های با استفاده از دو جهت در زمان بر خلاف

⁸Long Short-Term Memory

⁹Handwriting Recognition

¹⁰Machine Translation

¹¹Bidirectional Recurrent Neural Networks

¹²Generative Deep Learning

¹³Forward States

¹⁴Backward States

شبکه‌های بازگشت‌پذیر استاندارد در شبکه‌های بازگشت‌پذیر دوطرفه اطلاعات می‌تواند به صورت همزمان هم از آینده و هم از گذشته در استیت فعلی دریافت شود.

فصل ۵

بررسی مدل‌های پیاده‌سازی شده

حال که با مفاهیم اولیه مدل‌های پیاده‌سازی شده برای این پروژه آشنا شدیم، در این بخش به بررسی دقیق‌تر هر کدام از آنها می‌پردازیم.

در این پروژه به منظور نمونه‌سازی^۱ مدل‌ها با سرعت بالا، از قدرت کتابخانه‌ی Scikit-learn زبان برنامه‌نویسی پایتون^۲ بهره می‌گیریم. این کتابخانه بسیاری از مدل‌های مشهور در زمینه یادگیری با نظارت را پیاده‌سازی کرده است و امکان تنظیم کردن^۳ پارامترها را برای هر مدل تا حد بسیار خوبی فراهم می‌کند. تنظیم کردن صحیح پارامترها به منظور دستیابی به بهترین دقت ممکن از اهمیت بسیار بالایی برخوردار است.

پیش از توضیح مدل‌ها ابتدا به بررسی مجموعه ویژگی‌های^۴ استفاده شده برای آموزش هر کدام از مدل‌ها می‌پردازیم.

۱.۵ استخراج ویژگی‌ها^۵

بسیاری از ویژگی‌های مجموعه داده‌ی ما از نوع عددی می‌باشند، به همین علت داده‌هایی نظیر وزن و قد را به همان صورتی که هستند در آموزش مدل‌ها استفاده می‌کنیم.

^۱Prototyping

^۲Python

^۳Tuning

^۴Feature Set

^۵Feature Extraction

بخش دیگری از ویژگی‌های ما به صورت کیفی می‌باشند. مثال خوبی از این موضوع گروه سنی افراد است. در این مجموعه داده سن افراد به صورت گروه سنی نگهداری می‌شود. به همین منظور برای هر گروه سنی معرفی شده در مجموعه داده یک عدد به عنوان نماینده انتخاب می‌کنیم و به ویژگی سن برای آموزش مدل‌ها اختصاص می‌دهیم. در این پروژه این عدد را تعداد ماه زندگی افراد به طور تقریبی قرار داده‌ایم. به عنوان مثال برای نوزادها عدد ۶ را در نظر گرفتیم، که نماینده این است که در تقریب ما نوزادها ۶ ماه سن دارند.

بسیاری از داده‌های کیفی نیز به صورت دودویی می‌باشند که به سادگی آن‌ها را به صفر و یک تبدیل کردیم. به عنوان مثال برای جنسیت افراد به جنسیت زن عدد صفر و به جنسیت مرد عدد یک را اختصاص دادیم. بسیاری دیگر از ویژگی‌ها نظیر وضعیت بارداری افراد نیز به همین صورت به متغیرهای عددی تبدیل می‌گردند.

حال به سراغ مهم‌ترین بخش استخراج ویژگی‌ها، یعنی تبدیل داده‌های صوتی به متغیرهایی قابل فهم برای مدل‌های تحت آموزش می‌رویم. هر بیمار می‌تواند تا پنج فایل صوتی متفاوت داشته باشد که از نقاط مختلفی از قلب ضبط شده‌اند.

برای هر کدام از این فایل‌های صوتی ابتدا به وسیله تبدیل فوریه، فرکانس را استخراج نمودیم. حال برای این فرکانس‌ها ویژگی‌هایی نظیر میانگین،^۶ انحراف معیار، میانه^۷ و چولگی^۸ را محاسبه و به عنوان بخشی از ویژگی‌ها برای آموزش مدل‌ها استفاده می‌کنیم.

به عنوان بخش دیگری از ویژگی‌ها به سراغ محاسبه‌ی ویژگی‌های پیچیده‌تری می‌رویم. از خود داده‌های صوتی طول آن‌ها، تعداد فریم^۹ و همچنین شدت^{۱۰} به دسیبل^{۱۱} را استخراج می‌کنیم. یکی دیگر از ویژگی‌ها استخراج شده از داده‌های صوتی ویژگی MFCC می‌باشد. این ویژگی توان طیف

کوتاه مدت را در اختیار ما قرار می‌دهد و امروزه یکی از پرکاربردترین ویژگی‌ها در آموزش مدل‌ها بر پایه داده‌های صوتی می‌باشد.

⁶Mean

⁷Median

⁸Skew

⁹Frame

¹⁰Intensity

¹¹Decibel

۲.۵ پیاده‌سازی مدل‌ها

پس از آشنایی اجمالی با مجموعه ویژگی‌های استخراج شده برای آموزش، آماده بررسی پیاده‌سازی مدل‌ها می‌باشیم.

۱.۲.۵ جنگل تصادفی

همانطور که پیشتر ذکر شد، به منظور نمونه‌سازی سریع از قدرت کتابخانه Scikit-learn استفاده می‌نماییم. در این کتابخانه طبقه‌بند جنگل تصادفی با نام RandomForestClassifier وجود دارد که از آن استفاده می‌کنیم. پارامترهایی که آن‌ها را برای این مدل تنظیم می‌کنیم تعداد درخت‌های موجود در جنگل و همچنین تعداد بیشینه^{۱۲} برگ موجود در هر درخت می‌باشد.

۲.۲.۵ رگرسیون لوجستیک

در کتابخانه مذکور برای پیاده‌سازی رگرسیون لوجستیک تابع LogisticRegression وجود دارد که از آن بهره می‌بریم، رگرسیون لوجستیک به منظور طبقه‌بندی دو دسته اغلب کاربرد دارد و در مسائل اینچنینی که چندطبقه می‌باشند باید از تابع لفاف‌پیچی^{۱۳} موجود در کتابخانه Scikit-learn یعنی MultiOutputClassifier استفاده نماییم.

۳.۲.۵ بردار پشتیبان چندطبقه

همانطور که در بخش قبل به آن اشاره شده بردار پشتیبان به صورت عادی مانند رگرسیون لوجستیک برای تصمیم‌گیری میان دو دسته کاربرد دارد. به همین منظور باید از بردار پشتیبان چندطبقه برای این پروژه بهره بگیریم. این تابع با نام SVC در کتابخانه Scikit-learn پیاده‌سازی شده است. پس برای این پروژه از این کتابخانه به جای نسخه عادی آن یعنی SVM استفاده می‌کنیم. در تابع SVC

از رویکرد «یک در مقابل یک» که پیشتر با آن آشنا شدیم، استفاده شده است.

¹²Maximum

¹³Wrapper Function

۳.۵ بررسی یک مدل مبتنی بر یادگیری عمیق

به علت اهمیت این مسئله تیم‌های زیادی بر روی آن کار نموده‌اند. در این بخش به بررسی یکی از مدل‌های موفق که به دقت خوب ۷۵ درصد دست یافته است، می‌پردازیم. شرح مفصل این مدل در مقاله [۳۱] آمده است.

۱.۳.۵ استخراج ویژگی‌ها

سیگنال‌های PCG داده‌های پشت سر هم و در ابعاد بالا هستند و استفاده مستقیم از آن می‌تواند هزینه بسیاری را در پی داشته باشد. به همین علت به جای استفاده از سیگنال خام در این مقاله، از ۴ بسته^{۱۴} استخراج شده با فرکانس نمونه‌گیری^{۱۵} ۵۰ هرتز استفاده شده است. این بسته‌ها نه تنها اطلاعات فشرده‌ای از سیگنال در اختیار ما می‌گذارد [۳۲]، بلکه منجر به کاهش نویز در سیگنال می‌شود [۳۳].

تمام این صداهای ضبط شده با استفاده از یک فیلتر مرتبه دوم^{۱۶} با فرکانس قطع^{۱۷} ۲۵ و ۴۰۰ هرتز فیلتر شدند و به گونه‌ای نرمال‌سازی^{۱۸} شده‌اند که میانگین صفر و واریانس واحد داشته باشند. سپس با استفاده از روش توسعه داده شده توسط [۳۴] هم‌ریختی^{۱۹} هیلبرت^{۲۰} چگالی توان طیف و موجک^{۲۱} برای بسته‌ها محاسبه شده است. این بسته‌ها نیز مانند بخش قبلی نرمال‌سازی می‌شوند و یک سری زمانی چندمتغیره^{۲۲} ۴ بعدی را برای هر بخش از صدای قلب تشکیل می‌دهند.

۲.۳.۵ معماری شبکه عصبی

معماری شبکه عصبی استفاده شده در این مقاله که شکل آن در زیر مشخص است. از ۶۴ واحد انباشته شده BiLSTM تشکیل شده است. برخلاف شبکه‌های عصبی بازگشتی استاندارد شبکه

¹⁴Envelope

¹⁵Sampling Rate

¹⁶2nd-order Filter

¹⁷Cutoff Frequency

¹⁸Normalize

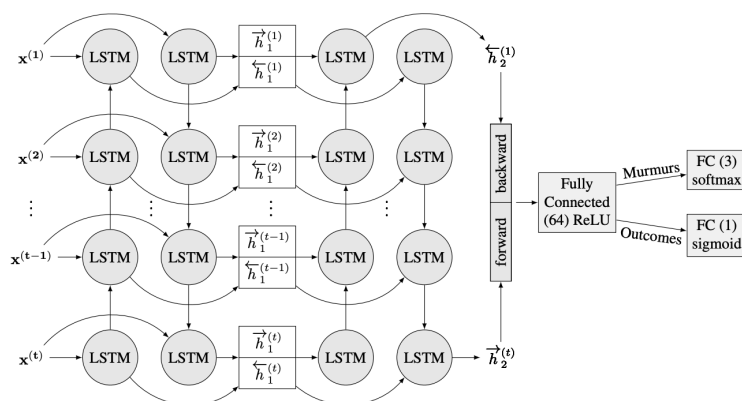
¹⁹Homomorphic

²⁰Hilbert

²¹Wavelet

²²Multivariate Time Series

LSTM می‌تواند مقادیر را در دوره‌های دلخواه فراموش یا بخاطر بسپارد و در نتیجه می‌تواند داده‌های ترتیبی که وابستگی در فواصل بالا دارند را، پردازش کند. این روش همچنین بسته‌ها را در دو جهت پردازش می‌کند و می‌تواند از اطلاعات سینگال از آینده و گذشته بهره‌گیری کند.



شکل ۱.۵: معماری شبکه عصبی [۳۱]

BiLSTM تمام یک توالی ^{۲۳} را می‌خواند سپس با استفاده از الحاق حالت نمونه قبلی در لایه جلو و حالت نمونه اول در لایه عقب، یک خروجی واحد تولید می‌کند. پس از این لایه یک لایه کاملاً متصل به منظور تخصیص یک طبقه از بین طبقات موجود استفاده شده است. در لایه آخر از تابع فعالسازی ^{۲۴} بیشینه فعال ^{۲۵} برای طبقه‌بندی سوفل و از تابع سیگموید برای طبقه‌بندی دودویی نتیجه بالینی استفاده شده است [۳۵].

۴.۵ ارزیابی مدل‌ها

پس از اینکه مدل‌های پیاده‌سازی شدند، نوبت به آزمون آن‌ها در میزان تشخیص صحیح بر روی داده‌های آزمون فرا می‌رسد.

می‌توان این بخش را اینطور خلاصه نمود که مدل‌های ساده مبتنی بر یادگیری با نظارت نمی‌توانند دقت مناسبی در داده‌های پیچیده‌ای نظیر داده‌های صوتی بدست آورند. به طوری که جنگل تصادفی

²³Sequence

²⁴Activation Function

²⁵Softmax

بهترین عملکرد را در میان مدل‌های یاد شده بدست آورد. که دقت آن در پیش‌بینی وجود یا عدم وجود سوफल در حدود ۳۰ درصد و همچنین در تشخیص موارد عادی یا غیرعادی بالینی دقت ۴۰ درصد را نتیجه می‌دهد. وضعیت برای رگرسیون لجستیک و بردار پشتیبان چندطبقه بسیار بدتر و در حدود ۱۵ درصد دقت می‌باشد.

نتیجه‌گیری نهایی از پیاده‌سازی مدل‌های یاد شده می‌تواند این باشد که هر چه به جلوتر پیش می‌رویم و مسائل پیچیده‌تر می‌شوند، مدل‌های ساده‌ای نظیر موارد استفاده شده در این پروژه کارایی چندانی ندارند. بهترین راه رسیدن به دقت‌های مناسب در اینگونه مسائل پیاده‌سازی مدل‌های پیچیده یادگیری عمیق^{۲۶} می‌باشد که نمونه آن را در بخش قبل مشاهده کردیم. این مدل‌ها معمولاً نیازمند صرف هزینه بالایی برای آموزش می‌باشند.

Model Name	Accuracy
Random Forest	30%
Logistic Regression	14.5%
SVC	16%
BiLSTM	75.5%

جدول ۱.۵: مقایسه مدل‌های شرح داده شده.

²⁶Deep Learning

فصل ۶

جمع‌بندی

در این پایان‌نامه به بررسی چند مدل مشهور در عرصه یادگیری با نظارت پرداختیم. سعی در این پروژه حل مسئله پیچیده‌ای متشکل از داده‌های عددی، کیفی و صوتی برای هر بیمار بود. که نهایتاً باید با استفاده از صداهای مختلف ضبط‌شده از نقاط مختلف قلب بیمار تشخیص داده می‌شد که آیا بیمار مورد آزمون سफल دارد یا خیر و همچنین آیا تشخیص بالینی آن عادی یا غیرعادی است. پس از پیاده‌سازی مدل‌ها همانطور که در قسمت ارزیابی مدل‌ها به آن اشاره شد، با استفاده از مدل‌های ساده یادگیری با نظارت نمی‌توان به دقت بالایی در چنین مسئله پیچیده‌ای دست یافت و باید از قدرت مدل‌های مبتنی بر یادگیری عمیق به حل چنین مسائلی پرداخت.

داده‌های صوتی، تصویری و موارد دیگری اینچنینی امروزه از چنان پیچیدگی و حجم بالایی برخوردار است که تنها یک مدل قوی یادگیری عمیق که ساعت‌ها نیاز به آموزش و تنظیم پارامتر دارد، می‌تواند به نتایج قابل قبولی در چنین مسائلی دست یابد.

واژه‌نامه

Chord.....	آکورد
Hyperplane	ابر صفحه
Association Rule Mining	استخراج قوانین انجمنی
Cross Validation	اعتبارسنجی متقابل
Greedy Algorithm	الگوریتم حریصانه
Ideal	ایده‌آل
Standard Deviation	انحراف معیار
Maximum Likelihood Estimation.....	برآورد درست‌نمایی بیشینه
Label	برچسب
Envelope	بسته
Training Vectors	بردارهای آموزشی
Nonlinear Programming	برنامه نویسی غیرخطی
Optimization	بهینه‌سازی
Overfit	بیش برآزش
Maximum	بیشینه
Softmax	بیشینه فعال
Python	پایتون
Prediction	پیش‌بینی
Function	تابع
Link Function	تابع پیوند
Activation Function	تابع فعالسازی

Gaussian Function	تابع گوسی
Wrapper Function	تابع لفاف پیچی
Periodic Function	تابع متناوب
Loss Function	تابع هزینه
Time Delay	تاخیر زمانی
Integral Transform	تبدیل انتگرالی
Fourier Transform	تبدیل فوریه
Machine Translation	ترجمه ماشینی
Linear Combination	ترکیب خطی
Handwriting Recognition	تشخیص دست خط
Speech Recognition	تشخیص گفتار
Recursive Partitioning	تقسیم بندی بازگشتی
Regularization	تنظیم
Tuning	تنظیم کردن
Sequence	توالی
Bernoulli Distribution	توزیع برنولی
Gaussian Distribution	توزیع گوسی
Linearly Separable	جدایی پذیر خطی
Random Forest	جنگل تصادفی
Feedback Loop	چرخه بازخوردی
Polynomial	چند جمله ای
Skew	چولگی
Safety Margin	حاشیه اطمینان
Hard Margin	حاشیه سخت
Soft Margin	حاشیه نرم
Long Short-Term Memory	حافظه طولانی کوتاه-مدت
Forward States	حالت های پسین
Backward States	حالت های پیشین
Optimal Line	خط بهینه

Well-Order	خوش ترتیب
Clustering	خوشه‌بندی
Sparse	خلوت
Testing Data	داده آزمایش
Training Data	داده آموزشی
Validation Data	داده اعتبار
Data Mining	داده کاوی
Frequency Domain	دامنه فرکانس
Knowledge	دانش
Decision Tree	درخت تصمیم
Aortic Valve	دریچه آئورت
Pulmonary Valve	دریچه ریوی
Tricuspid Valve	دریچه سه‌تی
Mitral Valve	دریچه میترال
Decibel	دسیبل
Binary	دودویی
Duality	دوگانگی
Regression	رگرسیون
Linear Regression	رگرسیون خطی
Logistic Regression	رگرسیون لوجستیک
Multivariate Time Series	سری زمانی چندمتغیره
Fourier Series	سری فوریه
Bias	سوگیری
Nervous System	سیستم عصبی
Multi-Agent Systems	سیستم‌های چند عاملی
Sigmoid	سیگموئید
Recurrent Neural Network	شبکه عصبی بازگشتی
Bidirectional Recurrent Neural Network	شبکه عصبی بازگشتی دو طرفه
Feedforward Neural Network	شبکه عصبی پیشخور

Intensity	شدت
Shimmer	شیمر
Mel Frequency Cepstral Coefficients	ضرایب کپسترال مل
Lagrange Multiplier	ضرایب لاگرانژ
Class	طبقه
Classifier	طبقه‌بند
Multi-class SVC	طبقه‌بند بردار پشتیبان چندطبقه
Classification	طبقه‌بندی
Linear Data Classification	طبقه‌بندی خطی داده
Power Spectrum	طیف قدرت
AI Agent	عامل هوش مصنوعی
Operation	عملیات
Generalization	عمومیت
Euclidean Distance	فاصله اقلیدسی
Cutoff Frequency	فرکانس قطع
Sampling Rate	فرکانس نمونه‌گیری
Frame	فریم
Higher Dimensional Space	فضا با ابعاد خیلی بالاتر
Hilbert Space	فضای هیلبرت
Phonocardiogram	فونوکاردیوگرام
2nd-order Filter	فیلتر مرتبه دوم
Interpretability	قابلیت تفسیر
Lagrangian Duality Theory	قضیه دوگانی لاگرانژ
Minimum	کمینه
Bagging	کیسه‌گذاری
Stochastic Gradient Ascent	گرادیان افزایشی تصادفی
Lasso	لاسو
Logit	لوجیت
Support Vector Machine	ماشین بردار پشتیبانی

Regularization Parameter	متغیر تنظیم
Slack Variable	متغیر کمکی
Feature Set	مجموعه ویژگی
Model	مدل
Decision Boundary	مرز تصمیم‌گیری
Wavelet	موجک
Mean	میانگین
Median	میانه
Norm	نرم
Euclidean Norm	نرم اقلیدسی
Normalize	نرمال‌سازی
Information Theory	نظریه اطلاعات
Game Theory	نظریه بازی‌ها
Saddle Points	نقاط زینی
Prototyping	نمونه‌سازی
Neuron	نورون
Noise	نویز
Variance	واریانس
Kernel	هسته
Homomorphic	هم‌ریختی
Artificial Intelligence	هوش مصنوعی
Ensemble Learning	یادگیری ترکیبی
Deep Learning	یادگیری عمیق
Generative Deep Learning	یادگیری عمیق سازنده
Machine Learning	یادگیری ماشین
One-Vs-All	یک در مقابل همه
One-Vs-One	یک در مقابل یک

Bibliography

کتاب نامه

- [1] Danielle S Burstein, Pirouz Shamszad, Dingwei Dai, Christopher S Almond, Jack F Price, Kimberly Y Lin, Matthew J O'Connor, Robert E Shaddy, Christopher E Mascio, and Joseph W Rossano. Significant mortality, morbidity and resource utilization associated with advanced heart failure in congenital heart disease in children and young adults. *American heart journal*, 2019.
- [2] Salim G. M. Jivanji, Sulaiman Lubega, Bhupi Reel, and Shakeel A. Qureshi. Congenital heart disease in east africa. *Frontiers in Pediatrics*, 7, 2019.
- [3] Liesl Zuhlke, Mariana Mirabel, and Eloi Marijon. Congenital heart disease and rheumatic heart disease in africa: recent advances and current priorities. *Heart*, 99(21):1554–1561, 2013.
- [4] Simone Manso de Carvalho, Ivete Dalben, Jose Eduardo Corrente, and Claudia Saad Magalhaes. Apresentacao e desfecho da febre reumatica em uma serie de casos. *Revista Brasileira de Reumatologia*, 52(Rev. Bras. Reumatol., 2012 52(2)), Mar 2012.
- [5] Adrian Gheorghe, Ulla Griffiths, Adrianna Murphy, Helena Legido-Quigley, Peter Lamptey, and Pablo Perel. The economic burden of

- cardiovascular disease and hypertension in low- and middle-income countries: a systematic review. *BMC Public Health*, 18(1):975, Aug 2018.
- [6] Michael D. Seckeler and Tracey Hoke. The worldwide epidemiology of acute rheumatic fever and rheumatic heart disease. *Clinical Epidemiology*, page 67, February 2011.
- [7] Anshula Tandon, S Sengupta, Vinayak Shukla, and Sumita Danda. Risk factors for congenital heart disease (chd) in vellore , india. 2010.
- [8] Jorge Oliveira, Francesco Renna, Paulo Dias Costa, Marcelo Nogueira, Cristina Oliveira, Carlos Ferreira, Alipio Jorge, Sandra Mattos, Thamine Hatem, Thiago Tavares, Andoni Elola, Ali Bahrami Rad, Reza Sameni, Gari D Clifford, and Miguel T. Coimbra. The circor digiscope dataset: From murmur detection to murmur classification. 2021.
- [9] S Madeh Piryonesi and Tamer E El-Diraby. Data analytics in asset management: Cost-effective prediction of the pavement condition index. *J. Infrastruct. Syst.*, 26(1):04019036, March 2020.
- [10] S Madeh Piryonesi and Tamer E El-Diraby. Role of data analytics in infrastructure asset management: Overcoming data size and quality problems. *J. Transp. Eng. B Pavements*, 146(2):04020022, June 2020.
- [11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning. Springer Series in Statistics. Springer, New York, NY, December 2001.
- [12] Teaching guide: decision trees — aqa.org.uk. <https://www.aqa.org.uk/resources/business/as-and-a-level/business-7131-7132/teach/teaching-guide-decision-trees>. [Accessed 28-Feb-2023].
- [13] J R Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986.

- [14] Leo Breiman. *Machine Learning*, 45(1):5–32, 2001.
- [15] David W Hosmer and Stanley Lemeshow. *Applied Logistic Regression*. Wiley Series in Probability and Statistics. John Wiley & Sons, Nashville, TN, 2 edition, October 2000.
- [16] Peter Buhlmann and Sara van de Geer. *Statistics for high-dimensional data*. Springer series in statistics. Springer, Berlin, Germany, 2011 edition, June 2011.
- [17] Peter Buhlmann and Sara van de Geer. Theory for ℓ_1/ℓ_2 -penalty procedures. In *Statistics for High-Dimensional Data*, Springer series in statistics, pages 249–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [18] Christopher M. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, NY, August 2016.
- [19] B K Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, April 1995.
- [20] Preface. In *Pattern Recognition*, pages xv–xvii. Elsevier, 2009.
- [21] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, England, March 2000.
- [22] Te-Ming Huang, Vojislav Kecman, and Ivica Kopriva. *Kernel based algorithms for mining huge data sets*. Studies in Computational Intelligence. Springer, Berlin, Germany, 2006 edition, March 2006.

- [23] Bernhard Scholkopf and Alexander J Smola. Learning with kernels. Adaptive Computation and Machine Learning series. MIT Press, London, England, December 2001.
- [24] Bernhard Scholkopf and etc., editors. Advances in kernel methods. The MIT Press. MIT Press, London, England, December 1998.
- [25] File:Svm max sep hyperplane with margin.png - Wikipedia — en.m.wikipedia.org. https://en.m.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png. [Accessed 28-Feb-2023].
- [26] Ingo Steinwart and Andreas Christmann. Support Vector Machines. Information science and statistics. Springer, New York, NY, 2008 edition, August 2008.
- [27] Alex Castrounis. AI Explained — whyofai.com. <https://www.whyofai.com/blog/ai-explained>. [Accessed 04-Mar-2023].
- [28] A Graves, M Liwicki, S Fernandez, R Bertolami, H Bunke, and J Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell., 31(5):855–868, May 2009.
- [29] Hasim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Interspeech 2014, ISCA, September 2014. ISCA.
- [30] Xiangang Li and Xihong Wu. Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. 2014.
- [31] “sofia M. Monteiro, Ana Fred, and Hugo P. Silva”. Detection of heart sound murmurs and clinical outcome with bidirectional long short-term

- memory networks. In Computing in Cardiology Conference (CinC). Computing in Cardiology, December 2022.
- [32] Yao Chen, Yanan Sun, Jiancheng Lv, Bijue Jia, and Xiaoming Huang. End-to-end heart sound segmentation using deep convolutional recurrent network. *Complex Intell. Syst.*, 7(4):2103–2117, August 2021.
- [33] Jose Javier Gonzalez Ortiz, Cheng Perng Phoo, and Jenna Wiens. Heart sound classification based on temporal alignment techniques. In 2016 Computing in Cardiology Conference (CinC). Computing in Cardiology, September 2016.
- [34] David B. Springer, Lionel Tarassenko, and Gari D. Clifford. Logistic regression-hsmm-based heart sound segmentation. *IEEE Transactions on Biomedical Engineering*, 63(4):822–832, 2016.
- [35] Yoshua Bengio. *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press, London, England, November 2016.

Abstract

; Audio data analysis using machine learning-based models has come to the aid of many sciences such as medicine today. By using audio data analysis, many diseases can be diagnosed and high treatment costs can be prevented. In this thesis, we try to detect whether people have a murmur problem in their heart and also whether their heart is in a normal or abnormal state by using audio data analysis using several models based on training with supervision. At the end, we measure the accuracy of each model and conclude which one performed better.



College of Science
School of Mathematics, Statistics, and Computer Science

Diagnosis the Presence of Murmur in the heart of Patients with Supervised Learning Models

Fatemeh Saburruh Monfared

Supervisor: Dr. Seyed Morteza Amini

A thesis submitted in partial fulfillment of the requirements for the degree of

B.Sc. in Statistics

for the degree of B.Sc.in Statistics

2023