قدم اول : طرز فکر خود را تغییر دهید.

باور داشته باشید که می توانید ماشین لرنینگ را تمرین کرده و در راستای علائق خود به کار بگیرید.

* چه چیزهایی شما را از اهداف ماشین لرنینگتان باز می دارد؟

الف ) باورهای محدود کننده

باورهای محدود کننده ی خود را شناسایی کرده و با آن ها مقابله کنید تا پیشرفت کنید.

باورهای محدود کننده چیزهایی هستند که در ابتدا آن ها را درست فرض می کنید و تلاشی برای زیر سوال بردن این فرضیات انجام نمی دهید:

* باور های if – else ای : اگر من ماشین لرنینگ را شروع کنم، چون به اندازه کافی خوب نیستم شکست می خورم.
* باورهای عمومی و جهانی: همه ی Data scientist ها مدرک دکتری دارند و خدای ریاضیات هستند.
* باورهای شخصی و عزت نفس: من آنقدر خوب نیستم که یک یادگیرنده ماشین باشم.

شما آدم منطقی ای هستید..موارد بالا را در تفکرات خود به چالش بکشید.

ب ) منتظر شروع ماندن

بزرگترین گروه از باورهای محدود کننده این است که تا وقتی دانش خاصی نداشته باشید نمی توانید شروع کنید.مسئله این است که آن دانش پیش نیازی که فکر می کنید برای شروع لازم دارید یا اصلا به کارتان نخواهد آمد، و یا آنقدر آن مبحث گسترده است که خود مهندسان و دانشمندان آن حوزه هم همه ی آن را نمی دانند.

به عنوان مثال: "من باید آمار یاد بگیرم." ببینید این حرف چقدر مبهم است.چه مقدار از آمار؟ در چه بخش هایی از آمار؟ و اصلا چرا باید قبل از یادگیری ماشین آمار را بلد باشید؟

به موارد زیر توجه کنید..من نمی توانم یادگیری ماشین را شروع کنم تا وقتی که...

* مدرک بالاتری بگیرم.
* یک دوره آموزشی را تمام کنم.
* در جبر خطی تسلط پیدا کنم.
* تئوری آمار و احتمال را بلد باشم.
* در زبان برنامه نویسی پایتون یا R به تسلط برسم.

شما می توانید همین امروز یادگیری ماشین را شروع کنید و اولین classification خود را ظرف مدت 5 دقیقه اجرا کنید.بعد از آن شما داخل ماشین لرنینگ قرار دارید.حالا شروع کنید بفهمید که واقعا از ماشین لرنینگ چه چیزی می خواهید.

ج) منتظر یک شرایط عالی هستید.

دسته ی دیگری از اعتقادات محدود کننده این است که ما قبل از شروع، منتظر یک شرایط عالی هستیم که در آن شروع کنیم.شرایط هیچ وقت عالی و مناسب نخواهد شد. شروع کنید و به تکاپو بیوفتید.شرایط قابل تحمل و بهتر خواهد شد. سپس دوباره به تکاپو بیوفتید و جلو بروید.

من نمی توانم ماشین لرنینگ را شروع کنم چون...

* الان وقت کافی ندارم.
* سیستم قوی و کامپیوتر و سخت افزار خوبی ندارم.
* من الان فقط یک دانشجو هستم.
* من در این لحظه برنامه نویس خوبی نیستم.
* شغل و برنامه هایم خیلی فشرده و شلوغ است.

برای تسلط پیدا کردن در یادگیری ماشین زمان و تمرین زیادی لازم است؛ اما نه یک باره و در ابتدای کار.

شما می توانید با چند ساعت وقت گذاشتن در هفته یا ده ها دقیقه در طول روز پیشرفت خوبی داشته باشید.بسیاری از کارهای کوچک در ماشین لرنینگ وجود دارد که می توانید برای شروع کار بین کارهایتان انجام دهید.شما می توانید شروع کنید.این مثل تمام چیزهای خوب دیگر در زندگی، فقط کمی فداکاری نیاز دارد.

د ) اقدام کرده اید یا چندبار امتحان کرده اید و شکست خورده اید.

سومین دسته از باورهای محدود کننده این است که در جایی شروع کوچکی انجام داده اید و تلاش کردید اما شکست خورده و نتوانسته اید به اهداف خود دست پیدا کنید.

خب این فیلد یک مبحث سخت است.ماشین لرنینگ هم مثل تمام کارهای فنی دیگر سخت است، اما سخت تر از مهارت های فنی دیگری مثل برنامه نویسی نیست.مستلزم پشتکار و فداکاری است.این دانش کاربردی و تجربی است و نیازمند آزمون و خطاست.

من نمیتوانم وارد یادگیری ماشین شوم زیرا...

* احساس میکنم غرق شده ام و راه نجاتی ندارم.
* من موضوع x را نمی فهمم.
* هرگز نمی توانم مثل y ماهر و متخصص شوم.
* نمیدانم باید بعد از آن چه کاری انجام دهم.
* نمی توانم برنامه ام را عملی کنم.

توصیه ی من کاهش دامنه یا تغییر جهت است.من تا جایی که می توانم از پروژه های کوچک طرفداری می کنم،چون این روش برای من موفق بوده است.

باورهای خود محدود کننده چیست؟

به خود محدود کنندگی اعتقاد دارید؟ به آن فکر کنید. اهدافتان چیست؟ فکر می کنید چرا هنوز آن جا نیستید؟

هدف شما این است که وارد ماشین لرنینگ شوید و به یک مهندس داده ، دانشمند داده یا مهندس ML تبدیل شوید، اما هنوز اولین قدم خود را برنداشته اید؟

* آیا منتظر هستید تا قبل از شروع یک تخصص خاص را کسب کنید؟
* آیا قبل ازشروع منتظر شرایط مناسب هستید؟
* آیا قدم اول را برداشته اید و مسیر را رها کرده اید؟

می خواهید کجا باشید و برای چه چیزی می جنگید؟

* چرا ماشین لرنینگ آن قدر ها هم سخت نیست؟

مباحث فنی مانند ریاضیات ، فیزیک و حتی علوم کامپیوتر با استفاده از روش پایین به بالا آموزش داده می شود. این روش شامل بیان موضوعات در یک منطقه از مطالعه به روشی منطقی با پیشرفت طبیعی در پیچیدگی و توانایی است.

مشکل این است که ، انسان ها ربات هایی نیستند که یک برنامه یادگیری را اجرا می کنند. ما به انگیزه ، هیجان و از همه مهمتر رسیدن موضوع به نتایج ملموس نیاز داریم.

مهارت های مفیدی مانند خواندن ، رانندگی و برنامه نویسی که ما هر روز استفاده می کنیم از این طریق آموخته نشده اند و در واقع با استفاده از یک روش معکوس، یعنی از بالا به پایین یاد گرفته شده اند. این روش از بالا به پایین می تواند برای یادگیری مستقیم موضوعات فنی مانند یادگیری ماشین مورد استفاده قرار گیرد ، که باعث خواهد شد شما خیلی سریع بهره وری بیشتری داشته باشید و یادگیری برای شما بسیار سرگرم کننده باشد.

در ادامه تفاوت های بنیادی این دو روش را بررسی خواهیم کرد و خواهیم دانست که چرا باید از روش دوم در یادگیری های مربوط به ماشین لرنینگ و حتی ریاضیات مرتبط با آن استفاده کرد.

1 ) یادگیری پایین به بالا

رشته ای مثل ریاضیات را در نظر بگیرید.

یک روش منطقی برای طرح مباحث در ریاضیات وجود دارد که بر اساس پیشرفت طبیعی مهارت ها ، توانایی ها و درک آنها ساخته شده و به وجود می آید.

مسئله این است که این پیشرفت منطقی ممکن است فقط برای کسانی که قبلاً در آن رشته فعالیت داشته اند و می توانند روابط بین موضوعات را درک کنند ، منطقی باشد.

بیشتر مدارس حول این روش آموزش از پایین به بالا فعالیت می کنند و تعداد زیادی از رشته های تحصیلی فنی و علمی از این طریق آموزش داده می شود.

به تحصیلات دبیرستان یا لیسانس و زمینه های اساسی که ممکن است در آنها کار کرده اید فکر کنید؛ مثالهایی مانند:

-ریاضیات، همانطور که قبلا گفته شد.

-زیست شناسی

-شیمی

-فیزیک

-علوم کامپیوتر

به چگونگی طرح مطالب به صورت هفته به هفته ، ترم به ترم و سال به سال فکر کنید. از پایین به بالا ؛ پیشرفت منطقی!

مسئله این است که ، پیشرفت منطقی از طریق مطالب، ممکن است از نظر بهره وری،بهترین راه برای یادگیری مطالب نباشد.

ما ربات نیستیم که یک برنامه یادگیری را اجرا کنیم. ما انسانهای احساسی هستیم که نیاز به انگیزه ، علاقه ، توجه ، تشویق و نتیجه داریم.

شما می توانید موضوعات فنی را از پایین به بالا بیاموزید و البته درصد کمی از مردم این روش را ترجیح می دهند ، اما لازم است بدانید که این تنها راه نیست.

اکنون ، اگر یک موضوع فنی را به سرانجام رسانده اید ، به چگونگی یادگیری آن فکر کنید. شرط می بندم از پایین به بالا نبوده است.

2) مهارت خواندن

فکر کنید. چطور یاد گرفتید که بخوانید؟

وقتی یک بچه شروع به خواندن می کند، بدون اینکه زیاد فکر کنیم، روش ها و تکنیک های کلی که برای آموختن به او استفاده می شود، به صورت زیر است:

* برای ایجاد علاقه و نشان دادن مزیت ها و منافعی که در این کار وجود دارد، با خواندن شروع کنید.
* الفبا را یاد گرفته و صداهای صحیح ایجاد کنید.
* متداول ترین کلمات ، اصوات آنها و نحوه هجی کردن آنها را بخاطر بسپارید
* برای مقابله با کلمات ناشناخته،روش های کلیدی "هجی کردن کلمه" را بیاموزید
* کتاب ها را با نظارت بخوانید.
* کتاب ها را بدون نظارت بخوانید.

این مهم است که او به طور مداوم ، با توجه به ارتباط خواندن به چیزهای ملموس زیادی که می خواهد انجام دهد ، بداند چرا خواندن مهم است.برای فهمیدن این اهمیت، کارهای زیر را برای او انجام دهید:

- زیرنویس ها را در نمایش های تلویزیونی بخوانید

- داستانهایی را درباره موضوعاتی که او دوست دارد مانند جنگ ستارگان بخوانید

- وقتی بیرون می روید ، علائم و منوها را بخوانید

- و ...

همچنین مهم است که او به نتایجی دست یابد که بتواند آنها را دنبال کند و در آن بهبود یابد

- دامنه واژگان گسترده تر

- سبک خواندن روان تر

- کتاب هایی با افزایش پیچیدگی

خواندن را اینگونه یاد نگرفته اید:

-تعاریف انواع کلمات (فعل ، اسم ، قید و غیره)

- قوائد گرامر

- قوائد علائم نگارشی

- نظریه زبان های انسانی

3 ) مهارت رانندگی

رانندگی می کنید؟

اگر این کار را نکنید عالی است ، اما اکثر بزرگسالان این کار را به دلیل ضرورت انجام می دهند.جامعه و طراحی شهر حول محور تحرک شخصی بنا شده است.

رانندگی را چطور یاد گرفتید؟

چیزی که من به خاطر می آورم،چند تست کتبی و شاید یک تست روی کامپیوتر بوده باشد.من خاطره ای از مطالعه برای آن ها ندارم گرچه به احتمال زیاد این کار را کرده ام. یادم می آید که یک مربی رانندگی استخدام کردم و دروس رانندگی را هم گذراندم.

هر درس آن به صورت عملی بود.در ماشین، مهارتی را که میخواستم گواهینامه و لایسنس آن را بگیرم تمرین می کردم؛یعنی رانندگی وسیله نقلیه در ترافیک.

موارد زیر چیزهایی هستند که من در مورد آن ها مطالعه نکردم و یا با مربی رانندگی خود در مورد آن ها صحبت نکردم:

- تاریخچه اتومبیل

- تئوری موتور های احتراق

- خطاهای رایج مکانیکی در خودروها

- سیستم الکتریکی ماشین

- نظریه جریان های ترافیک

تا امروز من توانسته ام بدون اینکه هیچ اطلاعاتی در مورد موضوعات بالا داشته باشم، به طور ایمن رانندگی کنم.در واقع من اصلا انتظار ندارم این مباحث را یاد بگیرم.

علاقه و نیاز من به این موارد صفر است و آن ها به من در چیزی که به آن نیاز دارم و می خواهم، یعنی تحرک شخصی آسان و ایمن هیچ کمکی نمی کنند.

اگر ماشین خراب هم شود، من با یک متخصص تماس می گیرم.

4 ) مهارت برنامه نویسی

من برنامه نویسی را بدون اینکه از معنای کدگذاری یا مهندسی نرم افزار هیچ ایده ای داشته باشم، شروع کردم.در خانه من با دستورات بیسیک شروع کردم و کلنجار رفتم.من با دستورات Excel ور رفتم.من بازی های کامپیوتری را اصلاح کردم و... این کارها سرگرم کننده بود.

وقتی شروع به یادگیری برنامه نویسی و مهندسی نرم افزار کردم ،در دانشگاه بود و سیستم آموزش آن از پایین به بالا بود.

ما با موارد زیر شروع کردیم:

- نظریه زبان

- انواع داده ها

- ساختارهای کنترل جریان

- ساختارهای داده

- و...

وقتی می توانستیم کد بنویسیم ، این کد در خط فرمان بود و با مشکلات کامپایلر ، مشکلات مسیر ، و یک سری مشکلات که واقعا هیچ ربطی به یادگیری برنامه نویسی نداشتند روبرو بودیم.

من از برنامه نویسی متنفر بودم.

چند سال بعد – سرانجام من کارم را به عنوان یک مهندس نرم افزار حرفه ای در تعدادی از سیستم های پیچیده و پر کاربر شروع کردم.من واقعا در این کار مهارت داشتم و آن را دوست داشتم.

سرانجام ، من پروژه ای را انجام دادم که نحوه ایجاد رابط های گرافیکی کاربر را نشان می داد.و دیگری که نشان می داد چطور می توان کاری کرد که با برنامه نویسی سوکت ها، کامپیوترها با هم صحبت کنند.و دیگری راجع به دسترسی همزمان به چند چیز توسط مسیرها.

من چیزهای کسل کننده و خسته کننده را با چیزی که واقعا دوست داشتم مرتبط کردم. ساختن نرم افزاری که بتواند مشکلات را حل کند و دیگران بتوانند از آن استفاده کنند. من آن را به چیزی متصل کردم که مهم بود.این دیگر نامفهوم و ترسناک نبود.

حداقل به من و توسعه دهندگانی مثل من، آن ها اشتباه آموزش دادند.واقعا اشتباه آموزش دادند.و این اشتباه باعث شد تا من و دانش آموزانی مثل من،چندین سال از وقت آزاد و همچنین تلاش هاو نتایجی که می توانست به دست بیاید را تلف کنیم.

5 ) یادگیری از بالا به پایین

روش یادگیری پایین به بالا معمولا رایج ترین راه برای آموزش مباحث فنی نیست.بلکه به نظر می رسد تنها راه ممکن است. حداقل تا زمانی که به چگونگی یادگیری واقعی خود فکر کنید.

طراحان دوره های دانشگاهی ، یعنی کسانی که در حوزه موضوع خود استادند ، در تلاشند تا به شما کمک کنند. آنها همه چیز را ارائه می دهند تا در پروسه ی پیشرفت منطقی، از طریق یک سری مطالب که فکر می کنند درست است (یا حداقل امیدوارند)، شما را به مهارت ها و توانایی های مورد نیاز برسانند.

و همانطور که اشاره شد ممکن است این روش برای برخی افراد هم مفید باشد.اما این روش برای من مفید نبود و انتظار هم ندارم که برای شما جواب دهد.

در حقیقت ، تعداد کمی از برنامه نویسانی که من با آنها آشنا بوده ام و در کار خود واقعا خوب بودند، تحصیلات دانشگاهی علوم کامپیوتر داشتند؛ یا اگر تحصیلات دانشگاهی هم داشتند ، آن مهارت ها را به تنهایی در خانه، با هک کردن و صرف وقت روی پروژه های جانبی یاد گرفته بودند.

یک جایگزین برای این روش، رویکرد از بالا به پایین است.

رویکرد متعارف را از بخش بالایی آن بچرخانید.

با تعاریف و تئوری شروع نکنید. در عوض ، با اتصال موضوع به نتایج مورد نظر خود شروع کنید و نشان دهید که چطور به سرعت نتیجه می گیرید.

برنامه ای را تنظیم کنید که تمرکز آن روی تمرینِ روند به دست آوردن نتیجه است ، در برخی از مواردی که نیاز دارید عمیق تر شوید،اما همیشه به دنبال نتیجه ای باشید که آن ها به آن نیاز دارند.

این مسیر متفاوت است

این مسیر مثل مسیر قدیمی نیست. پس مراقب باشید هنگامی که این مسیر را طی می کنید از روش های قدیمی تفکر و مقایسه استفاده نکنید.

مسئولیت این مسیر با شماست.هیچ شخص یا سیستمی نیست که بخواهید آن را سرزنش کنید.شما فقط وقتی شکست می خورید که متوقف شوید.

- این مسیر تکراری است و موضوعات هر بار با درکی عمیق تر مرور خواهند شد.

- یادگیری در این مسیر ناقص است. نتایج ممکن است در آغاز ضعیف باشد ، اما با تمرین بهبود می یابد

- این مسیر مستلزم کشف است. یادگیرنده باید برای یادگیری و کشف مداوم، آغوشی باز داشته باشد.

- این مسیر نیازمند مالکیت است؛مال شماست و نه هیچ کس دیگر. مسئول پیشرفت شما، خودتان هستید.

- این مسیر به کنجکاوی نیاز دارد. یادگیرنده باید به موارد مورد علاقه خود توجه کند و آن را دنبال کند.

این مسیر خطرناک است

به طور جدی من بارها شنیده ام که متخصصان بارها این جمله و جملاتی مانند آن را گفته اند:

قبل از استفاده از این تکنیک باید تئوری را خوب یاد بگیرید، وگرنه نمی توانید به درستی از آن استفاده کنید.

من موافقم که نتایج در ابتدا ناقص خواهد بود ، اما پیشرفت و حتی تخصص فقط از راه تئوری و فاندامنتال آن به دست نمی آیند.

اگر فکر می کنید که یک برنامه نویس مبتدی نباید تغییری در محصول نهایی ایجاد کرده و تکنیک ها را به کار گیرد ، مطمئناً باید باور داشته باشید که مبتدی ای که به دنبال یادگیری ماشین هست هم همان محدودیت ها را باید تحمل کند.

مهارت باید نشان داده شود. اعتماد باید کسب شود.این قضیه،صرف نظر از اینکه از چه راهی مهارت را کسب کرده اید، درست است.

شما یک تکنسین هستید

واقعا؟!

این یک انتقاد دیگر است که من در مورد این رویکرد یادگیری مشاهده کرده ام.

دقیقا!

ما می خواهیم تکنسین باشیم و از ابزارهای عملی برای کمک به مردم استفاده کنیم.ما نمی خواهیم محقق باشیم.

شما نیازی ندارید که همه زمینه های یکسان با محققان را پوشش دهید، زیرا هدف یادگیری متفاوتی دارید. اگرچه بعداً دانش زمینه ای را در اختیار خواهید داشت و برای تکمیل دانش انتزاعی خود، می توانید هر آنچه را دوست دارید بعداً یاد بگیرید.

توسعه دهندگان در صنعت، دانشمند کامپیوتر نیستند. آنها مهندس و تکنسین های پرافتخارِ صنایع ساخته ی دست انسان هستند.

یک روش کارآمد، موثر و سرگرم کننده برای یادگیری

مزایای یادگیری این مسیر بسیار بیشتر از چالش های آن است:

* شما مستقیما به سراغ چیزی که می خواهید می روید و شروع به تمرین با آن می کنید.
* شما زمینه ی لازم برای اتصال دانش عمیق تر و حتی نظریات را در اختیار خواهید داشت.
* می توانید بر اساس اهدافی که در آن زمینه دارید، به صورت موثر موضوعات را فیلتر کنید.

این روش سریع تر است.این روش سرگرم کننده تر است و شرط می بندم شما را خیلی بهتر می کند

چطور باعث می شود بهتر باشی؟

چون سوژه ی مورد مطالعه، از نظر احساسی با شما در ارتباط است ؛ شما آن را به نتیجه یا نتایجی متصل کرده اید که برای شما اهمیت دارد.شما سرمایه گذاری کرده اید.شما صلاحیتی دارید که قابل اثبات است. همه ی ما چیزهایی که در آن ها خوب هستیم را دوست داریم( حتی اگر نسبت به اینکه چقدر خوب هستیم دید خوبی نداشته باشیم.)که این مورد انگیزه و شور و اشتیاق را در ما برمی انگیزد.

کسی که با اشتیاق به دنبال یادگیری باشد، از همه ی اصولگرایان و کسانی که به دنبال مفاهیم علمی هستند جلو خواهد زد.

یادگیری ماشین را یاد بگیرید

بنابراین، چطور به موضوع یادگیری ماشین نزدیک شده اید؟

به طور صادقانه رویکرد خود را از بین گزینه های زیر مشخص کنید:

* آیا دوره ی دانشگاهی از پایین به بالا را می گذرانید؟
* آیا یادگیری خود را در چنین دوره ای مدل می کنید؟

یا نه بدتر از آن:

آیا یک دوره ی از بالا به پایین را دنبال می کنید، اما سرشار از حس گناه و حسادت ریاضی و ناامنی هستید؟

تو تنها نیستی! من این موضوع را همه روزه در بین مبتدیان مشاهده می کنم.

برای اتصال نقاط برای شما ، من شدیداً تشویقتان می کنم که با استفاده از روش بالا به پایین ، یادگیری ماشین را مطالعه کنید.

* با ریاضیات مقدماتی شروع نکنید.
* با تئوری های یادگیری ماشین شروع نکنید.
* هر الگوریتمی را کد نزنید.

همه ی این ها بعدا به دست می آید و یادگیری شما وقتی که برای این دانش های مختصر و چکیده ارتیاطاتی داشته باشید، عمیق تر می شود.

1 - با استفاده از یک چارچوب ثابت به وسیله ی ابزارهای open-source رایگان که استفاده ی آسانی هم دارند شروع کنید و چگونگی کار با مشکلات بسیار ساده ی مدل سازی پیش بینی را یاد بگیرید.

2 - در تعداد زیادی از پروژه های کوچک تمرین کنید و به آرامی پیچیدگی آنها را افزایش دهید.

3- کارهای خود را با ایجاد نمونه کارهای عمومی به بقیه نشان دهید.

متخصصانی که در دانشگاه ها ریشه دارند به شما خواهند گفت که این کار خطرناک است؛ آن ها را نادیده بگیر.

تمرین کنندگان کلاس جهانی به شما خواهند گفت که این روشی است که آن ها یاد می گیرند و به یادگیری ادامه می دهند.از آن ها الگو بگیرید.

به یاد داشته باشید:

* شما خواندن را با تمرین خواندن یاد گرفتید ، نه با مطالعه تئوری زبان
* شما رانندگی را با تمرین رانندگی یاد گرفتید ، نه با مطالعه موتورهای احتراقی
* شما برنامه نویسی را با تمرین کدنوشتن یاد گرفتید ، نه با مطالعه تئوری محاسبه.

شما می توانید یادگیری ماشین را با تمرین مدل سازی پیش بینی یاد بگیرید ، نه با مطالعه ریاضیات و تئوری.

این نه تنها روشی است که من یادگیری ماشین را با آن یاد گرفتم و به تمرین با آن ادامه می دهم ، بلکه به ده ها هزار دانش آموز من نیز کمک کرده است.

یادگیری ریاضیات

در اینجا متوقف نشو!!

ممکن است زمانی فرا برسد که شما بخواهید یا نیاز داشته باشید که پرده ستون های ریاضی یادگیری ماشین مثل جبر خطی ، حساب ، آمار و احتمال و ... را کنار بزنید.

در این جا هم می توانید دقیقاً از همین روش از بالا به پایین استفاده کنید.

هدف یا نتیجه ای که برای شما مهم است را انتخاب کنید و از آن به عنوان لنز ، فیلتر یا غربالی برای مطالعه ی مباحث و یادگیری عمیق تری که نیاز دارید تا به آن نتیجه برسید، استفاده کنید.

به عنوان مثال، فرض کنیم شما جبرخطی را انتخاب کرده اید.

هدف ممکن است grok SVD or PCA باشد که روش هایی برای کاهش داده ها، featute selection و data projection هستند.

یک رویکرد از بالا به پایین می تواند به صورت زیر باشد:

1 – این روش را در یک کتابخانه سطح بالا مثل sklearn اجرا کرده و نتیجه بگیرید.

2 – این روش را در کتابخانه ی سطح پایین تری مثل numpy/ scipy اجرا کرده و مجددا همان نتیجه را تولید کنید.

3 – روش را مستقیما با استفاده از ماتریس ها و عملیات بین ماتریس ها در numpy یا octave پیاده سازی کنید.

4 – آن تعدادی از عملیات های ریاضی حسابی ماتریس که در آن روش ها به کار رفته است را مطالعه و بررسی کنید.

5 – آن تعدادی از عملیات های ریاضی تجزیه ی ماتریس که در آن روش ها به کار رفته است را مطالعه و بررسی کنید.

6 – روش هایی که برای تقریب و ترکیب ماتریس ها به کار می روند را مطالعه کنید. و ...

هدف زمینه را مشخص می کند و شما می توانید اجازه دهید تا کنجکاوی تان عمق مطالعه را مشخص کند.

با توجه به روشی که ترسیم شد، مطالعه ی ریاضی با مطالعه ی هر موضوع دیگری در برنامه نویسی، یادگیری ماشین و یا سایر دروس فنی تفاوتی نخواهد داشت.

این روش بسیار پربازده تر و بسیار سرگرم کننده تر است!

* چطور در مورد یادگیری ماشین فکر کنیم؟

یادگیری ماشین یک زمینه مطالعاتی بزرگ و میان رشته ای است.

با یادگیری ماشینی می توانید به نتایج چشمگیری برسید و برای مشکلات بسیار چالش برانگیز، راه حل هایی پیدا کنید. اما این تنها گوشه کوچکی از زمینه گسترده تری از یادگیری ماشین است که اغلب به آن مدل سازی پیش بینی یا تحلیل پیش بینی می گویند.

در این بخش خواهید در این مورد یاد می گیریم که چطور فکر خود را نسبت به ماشین لرنینگ تغییر دهیم تا بتواند به عنوان یک یادگیرنده ی ماشین لرنینگ،بهترین خدمات را به ما ارائه دهد.

شما گیج شده اید

شما یک مشکل یادگیری ماشین دارید که باید آن را حل کنید ، اما در مورد اینکه یادگیری ماشین دقیقاً چیست گیج شده اید.

دلیل خوبی برای گیج شدن وجود دارد. برای مبتدیان گیج کننده است.

یادگیری ماشینی یک زمینه بزرگ مطالعاتی است و اگر روی حل یک مسئله متمرکز باشید ، بیشتر آن به شما مربوط نخواهد بود.

ما با توصیف یادگیری ماشین به صورت گسترده و چگونگی ارتباط آن با سایر زمینه های مطالعاتی مانند آمار و هوش مصنوعی شروع می کنیم. پس از آن ، ما روی جنبه هایی از یادگیری ماشین که واقعا برای مهندسی عملی و حل مسئله به دانستن آن نیاز دارید ، فوکوس خواهیم کرد.

ماشین لرنینگ چیست؟

یادگیری ماشینی یکی از رشته های علوم رایانه است که مربوط به برنامه هایی است که خودشان یاد می گیرند.

حوزه یادگیری ماشین به این سوال مربوط می شود که چگونه می توان برنامه های رایانه ای را ساخت که با تجربه کردن، به طور خودکار بهبود می یابند

این حوزه فوق العاده گسترده است. انواع مختلفی از روش های یادگیری ، انواع مختلفی از فیدبک ها و بازخوردهایی که می توان برای یادگیری از آن ها استفاده کرد و همچنین مسائل خیلی زیادی وجود دارد که ماشین می تواند آن ها را یاد بگیرد.

این مسئله می تواند انواع مختلفی از یادگیری را شامل شود:

-توسعه ی یک کد برای بررسی و فهمیدن اینکه ارگانیسم های مختلف، چگونه در طی زمان و در فرآیند تکامل یادگرفتند که با محیط خود به سازگاری برسند.

-توسعه ی یک کد برای فهمیدن اینکه یک نورون در مغز چگونه در پروسه ی پاسخ به محرک های سایر نورون ها، یاد می گیرد.

-توسعه ی یک کد برای دانستن اینکه چطور مورچه ها بهینه ترین مسیر از لانه ی خود تا محل آذوقه را یاد می گیرند.

این مثالهای ترسناک و پیچیده را با این هدف آورده ام تا به شما کمک کنم واقعاً کمی متوجه شوید که یادگیری ماشینی یک برنامه گسترده و دور از دسترس در حوزه ی تحقیقات است.

مورد دیگری که ممکن است با آن بیشتر آشنا باشید این است:

-توسعه ی یک کد برای فهمیدن چگونگی یادگرفتن الگوها در داده های تاریخی

این موضوع زرق و برق کمتری دارد اما پایه و اساس گوشه ی کوچکی از ماشین لرنینگ است که ما به عنوان تازه کاران آموزش ماشین لرنینگ، عمیقا به آن علاقه مندیم.

این بخش فرقی با سایر مثال ها ندارد و می تواند در روش های یادگیری،وظایف اساسی و بنیادی،راه های ارزیابی یادگیری و ... همپوشانی زیادی با آن ها داشته باشد.

در مورد هوش مصنوعی چه می دانید؟

ماشین لرنینگ، زیر شاخه ای از هوش مصنوعی است.هوش مصنوعی با ماشین لرنینگ همپوشانی دارد.

هوش مصنوعی نیز یکی از حوزه های علوم رایانه است ، اما درگیر توسعه برنامه هایی است که هوشمند هستند یا می توانند کارهای هوشمندانه انجام دهند.

هوشمندی شامل یادگیری است ، به عنوان مثال یادگیری ماشینی ؛ اما ممکن است شامل موارد دیگری مانند استدلال ، برنامه ریزی ، حافظه و ... هم باشد.

هوش مصنوعی می تواند شامل مثال های زیر باشد:

-توسعه ی یک کد برای دستیابی به یک برنامه بهینه ی لجستیکی

-توسعه ی یک کد برای استدلال در مورد یک پاراگراف از متن

-توسعه ی یک کد برای درک موضوع یک عکس

هوش مصنوعی غالباً در چارچوب یک عامل در یک محیط با هدف رفع برخی از مشکلات طراحی می شود ، اما لازم نیست همیشه اینطور باشد.

یادگیری ماشینی را به همین راحتی می توان یادگیری مصنوعی نامید تا در کنار مفهوم هوش مصنوعی در ذهن بماند و به مبتدیان کمک کند.

خطوط و مرزهای این دو مفهوم تفکیک خاصی نداشته و مبهم هستند.مشکلات ماشین لرنینگ، به نوعی مشکلات هوش مصنوعی نیز هست.

در مورد آمار چه می دانید؟

آمار یا آمار کاربردی که به وسیله ی کامپیوترها به کار می رود ، زیرمجموعه ای از ریاضیات است که مربوط به توصیف و درک روابط موجود بین داده ها است.

این بخش می تواند شامل یادگیری موارد زیر باشد:

- توسعه ی مدل هایی برای خلاصه کردن توزیع یک متغیر

- توسعه مدل هایی برای توصیف بهتر رابطه بین دو متغیر

- توسعه مدل هایی برای تست وجود شباهت هایی در بین دو مجموعه از نمونه ها

این بخش با بخشی از ماشین لرنینگ که علاقه مند به یادگیری الگوهای موجود در داده ها است، همپوشانی دارد.

بسیاری از روش های مورد استفاده برای درک داده ها در آمار را می توان در یادگیری ماشین برای یادگیری الگوهای موجود در داده ها استفاده کرد. این وظایف را می توان یادگیری ماشین یا آمار کاربردی نامید.

ماشین لرنینگ تو

ماشین لرنینگ یک مبحث گسترده ی مطالعاتی است و می تواند به شما کمک کند تا مشکلات به خصوصی را حل کنید.

اما شما نیازی به دانستن همه ی بخش های آن ندارید.

* شما یک محقق دانشگاهی نیستید که دقیقا راجع به چگونگی انجام گرفتن یادگیری ماشینی تحقیق کنید.
* شما مثل حوزه ی هوش مصنوعی، سعی نمی کنید یک عامل هوشمند بسازید.
* شما مشتاق نیستید که راجع به این مسئله که چرا متغیرهای موجود در داده ها هم مثل آمار با هم ارتباط دارند، بیشتر بدانید.

در حقیقت وقتی صحبت از یادگیری روابط بین داده ها می شود،

- شما قصد بررسی توانایی های یک الگوریتم را ندارید.

- شما قرار نیست نظریه یا الگوریتم کاملا جدیدی بسازید.

- شما الگوریتم های موجود برای یادگیری ماشین را به موارد جدید گسترش نمی دهید.

این فعالیت ها،کارهایی است که در بخش دیگری از ماشین لرنینگ هستند و ممکن است ما هم به آن علاقه داشته باشیم . اما این ها فعالیت هایی برای دانشگاهیان است ، نه کسانی مثل ما که تازه شروع به یادگیری کرده ایم.

**با این حساب شما باید بر روی چه قسمت هایی از یادگیری ماشین تمرکز کنید؟**

فکر می کنم به دو صورت بتوانید در مورد ماشین لرنینگ فکر کنید:

1. از نظر مسئله ای که سعی در حل آن دارید.
2. از نظر راه حلی که نیاز دارید.

مسئله ی ماشین لرنینگ تو

مسئله ی شما می تواند به بهترین صورت، به شکل زیر توصیف شود:

مدل یا روشی را پیدا کنید که به بهترین نحو از داده های قبلی که از مجموعه ای از ورودی ها و خروجی ها تشکیل شده اند استفاده کند تا به صورت ماهرانه، بتواند خروجی های جدید و ورودی هایی که در آینده وارد خواهند شد را پیش بینی کنند.

این توانایی فوق العاده خاص است.

اول از همه ، این بخشهای فرعی یادگیری ماشین ، مانند یادگیری بدون نظارت ، را کنار می گذارد تا روی یک نوع یادگیری به نام یادگیری نظارت شده و همه الگوریتم های متناسب با آن تمرکز کند.

این بدان معنا نیست که شما نمی توانید از روش های بدون نظارت استفاده کنید؛این فقط به این معنی است که شما در آن بخش متمرکز نشوید؛ حداقل در شروع کار!

ثانیا این کار یک هدف مشخص به شما می دهد که بر همه ی آن ها ارجحیت دارد: مهارت و توانایی مدل؛ در عوض پرداختن به دغدغه های دیگری همچون پیچیدگی مدل،تفسیرپذیری مدل و...

باز هم ، این بدان معنا نیست که اینها مهم نیستند ، فقط این است که این موارد بعد از توجه به مهارت مدل یا در کنار آن در نظر گرفته می شوند.

ثالثا قالب بندی مسئله ی شما از این طریق، با شاخه ی مطالعاتی دیگری که مدل سازی پیش بینی نامیده می شود متناسب است. "مدل سازی پیش بینی" یک زمینه مطالعاتی است که متدهایی از ماشین لرنینگ را برای توسعه ی مدل هایی که پیش بینی های ماهرانه دارند ، استفاده می کند.در بعضی از بخش های بیزینس ، این حوزه ممکن است پیش بینی تجزیه و تحلیل نیز نامیده شود و مواردی فراتر از صرفا مدل سازی را شامل می شود که شامل فعالیت های مرتبط با جمع آوری و تهیه داده ها و استقرار و نگهداری مدل می باشد. اخیراً ، این فعالیت را علم داده نیز می نامند. اگرچه این عبارت علاوه بر کار با مسئله برای یافتن روش حل آن، به طراحی یا کشف مسئله هم ارتباط دارد.

فکر نمی کنم زیاد اهمیتی داشته باشد که شما این فعالیت را چه می نامید. اما من فکر می کنم مهم است که عمیقا درک کنیم که علاقه ی ما به ماشین لرنینگ بسیار خاص بوده و روش استفاده ما از ماشین لرنینگ با استفاده های دیگری که دانشگاهیان از آن می کنند،متفاوت است.این به ما اجازه می دهد که منابعی که مطالعه می کنیم یا ابزارهایی که استفاده می کنیم را با توجه به مسئله ای که در تلاش برای حل آن هستیم، فیلتر کرده و انتخاب کنیم.

راه حل ماشین لرنینگ تو

راه حلی که شما نیاز دارید، به خوبی در زیر تشریح شده است:

مدل یا رویه ای که به صورت خودکار، نزدیک ترین رابطه ی اساسی ناشناخته بین ورودی ها و خروجی های مرتبط با آن ها را در بین داده های قبلی می سازد.

باز هم این توانایی فوق العاده است.

شما به یک متد خودکار نیاز دارید تا برنامه یا مدلی را بسازد که شما بتوانید از آن برای پیش بینی استفاده کنید..شما نمی توانید بنشینید و برای حل مشکل خود کدنویسی کنید.این مسئله کاملا مختص داده هاست و شما هم تعداد زیادی داده در اختیار دارید.

در حقیقت مسائلی از این دست، در برابر راه حل های کدنویسی دستی از بالا به پایین مقاومت می کنند.اگر شما می توانستید بنشینید و برای حل مسائل خود تعدادی جمله ی if می نوشتید، به راه حل های ماشین لرنینگ احتیاج پیدا نمی کردید.در این صورت، آن مسئله یک مسئله ی برنامه نویسی می بود.

انواعی از متدهای ماشین لرنینگی که شما به آن ها نیاز دارید، روابط بین ورودی ها و خروجی ها را در بین داده های قبلی "یاد خواهند گرفت".

این قالب بندی به شما اجازه می دهد تا درباره ی این فکر کنید که آن زیربنای واقعی تابعی که هنوز شناخته شده نیست ممکن است چه شکلی باشد و نویز،انحراف و نمونه برداری از داده های قبلی چگونه ممکن است بر تقریب این مدلی که توسط روش های مختلف مدل سازی ساخته شده، اثر بگذارند.

بدون این قالب بندی، ممکن است از مواردی از این قبیل تعجب کنید:

-چرا فقط یک الگوریتم فوق العاده یا مجموعه ای از پارامترها وجود ندارد؟

-چرا متخصصان نمی توانند به ما بگویند که از چه الگوریتمی استفاده کنیم؟

-چرا با پیش بینی با استفاده از مدل های خود نمی توانیم به میزان خطای صفر دست پیدا کنیم؟

این موضوع به شما کمک می کند که همزمان تصویر نامناسبی که از ماهیت مسئله ی مدل سازی پیش بینی ای که سعی در حل آن دارید و مجموعه ای از انتظارات معقولی که می توان داشت را ببینید.

اکنون که می دانید چطور باید به ماشین لرنینگ فکر کنید، قدم بعدی این است که تفکر خود در مورد پروسه ی حل مسائل به وسیله ی روش های ماشین لرنینگ را تغییر دهید.

* قبیله ی ماشین لرنینگ خود را پیدا کنید

شروع کنید و از دریافت توصیه های اشتباه خودداری کنید

ماشین لرنینگ یک حوزه ی مطالعاتی جذاب و قدرتمند است که پر از داده و الگوریتم ها است.

مسئله این است که دسته های بسیار متفاوتی از آدم ها به ماشین لرنینگ علاقه مندند و هر کدام از آن ها نیز نیاز های متفاوتی دارند. مهم است که بدانید از ماشین لرنینگ چه می خواهید تا مطالعه خود را متناسب با آن نیازها انجام دهید. اگر این کار را نکنید ، می توانید خیلی راحت از مسیر منحرف شوید، علاقه خود را از دست داده و به آنچه که می خواستید نرسید.

قبیله ی خود را پیدا کنید

در این بخش با ده گروه اصلی از افرادی که به ماشین لرنینگ علاقه مندند آشنا خواهیم شد.من این ده گروه را، قبایل ماشین لرنینگ می گویم.

شما با مهمترین نیازهای عمومی و انواع منابعی که هر قبیله آن ها را کشف می کند، آشنا خواهید شد.نکته ی مهم این است که شما می توانید این ده قبیله را مرور کرده و بفهمید کجا مناسب شماست.از اینکه افراد دیگری که شبیه به شما هستند در آنجا حضور دارند احساس راحتی کنید و در مورد قدم های بعدی تان ایده پردازی کنید.

کدام قبیله مناسب شماست؟

از عدم تطابق با قبیله نترسید؛ این مسئله خیلی معمول است

شما به ماشین لرنینگ علاقه دارید.از اطرافیان خود می پرسید و یک دوره ی آموزشی یا کتاب خاص توصیه می شود.چند ساعت بعد ناامید می شوید.اما نمی دانید دقیقا چرا!

نمی دانید چه اتفاقی برایتان افتاده است.دلیلش این است که منبعی که انتخاب کرده اید عالی است، اما ممکن است برای شرایط کنونی شما عالی نباشد.

این یک مشکل رایج است و من آن را مشکل عدم تطابق می نامم.

نمونه های کلاسیک توسعه دهنده و کتب درسی

یک مثال کلاسیک، توسعه دهنده ای است که به حل یک مشکل یکبار مصرف علاقه مند است. به آنها کتاب درسی ماشین لرنینگ توصیه می شود.

آنها کتاب گفته شده را خریداری می کنند ، شروع به خواندن می کنند ولی هرگز فصل اول را تمام نمی کنند.

یک کتاب درسی راجع به ماشین لرنینگ،برای یک دانش آموز که از کلاس ماشین لرنینگ فارغ التحصیل شده و در چهارسال گذشته هم کلاس های ریاضی را گذرانده مناسب است.

این همچنین برای یک توسعه دهنده بی فایده است . ده سال وارد شغلشان می شوند و به ماشین لرنینگ به عنوان ابزاری برای رسیدن به نتیجه نگاه می کنند

به همین دلیل شناختن گروههای مختلفی که علاقه مند به یادگیری ماشین هستند و دانستن اینکه به کدام گروه تعلق دارید، حیاتی است. به این صورت که می توانید افراد دیگری که مثل شما هستند را پیدا کرده و از منابعی استفاده کنید که واقعا در پیداکردن راه حلی که به آن نیاز دارید به شما کمک خواهند کرد.

**مرور قبایل**

این یک دید کلی از ده قبیله ی گفته شده است:

\* قبایل تجاری((Business Tribes

1- شخصیت تجاری با منافع عمومی

2- مدیری که علاقه مند به ارائه ی یک پروژه است.

\* قبایل دانشگاهی(Academic Tribes)

3- دانشجوی یادگیری ماشین در یک کلاس کارشناسی یا کارشناسی ارشد

4– یک محقق در حوزه ی ماشین لرنینگ که علاقه مند به تاثیر گذاری در این زمینه است

5– یک محقق عمومی که علاقه مند است مسائل و مشکلات آن ها را مدل سازی کند

\*قبایل مهندسی (Engineering Tribes)

6- برنامه نویسی که به الگوریتم های در حال اجرا علاقه مند است.

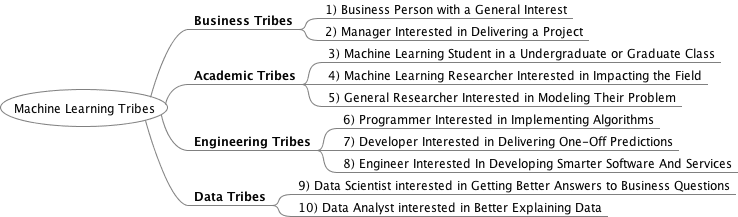
7- برنامه نویس علاقه مند به ارائه پیش بینی های یکبارمصرف

8- مهندس علاقه مند به توسعه نرم افزارها و خدمات هوشمند تر

\*قبایل داده (Data Tribes)

9- دانشمند داده ای که به ارائه ی پاسخ های بهتر به پرسش های حوزه ی کسب و کار علاقه مند است

10 – تحلیلگر داده ای که علاقه مند است داده ها را بهتر توضیح دهد.



* قبایل تجاری

به طور کلی ، این افراد علاقه مند به استفاده موثر از ماشین لرنینگ در سازمان خود هستند ، اما لزوما به جزئیات و اجزای الگوریتم ها و ابزارها علاقه ای ندارند.

تاجران ممکن است از عباراتی مانند هوش تجاری یا تجزیه و تحلیل پیش بینانه استفاده کنند ، که هر دو زمینه عمومی تری هستند که ممکن است از تکنیک های ماشین لرنینگ استفاده کنند.

1- شخصیت تجاری با منافع عمومی

این شخص ممکن است هر کسی باشد.از یک مدیر اجرایی گرفته تا یک مشاور که در مورد ماشین لرنینگ شنیده و به دنبال این است که در پروژه های آینده یا شاید ابتکارات خود،از آن به صورت استراتژیک استفاده کند.

این در واقع منطقه من نیست ، اما برخی از منابع مفید برای تفکر استراتژیک تر عبارتند از:

Gartner’s [Magic Quadrant for Advanced Analytics Platforms](https://www.gartner.com/technology/reprints.do?id=1-2AHPOU0&ct=150225&st=sb), 2015

Gartner’s [Machine Learning Drives Digital Business](https://www.gartner.com/doc/2820120/machine-learning-drives-digital-business), 2014

McKinsey’s [An executive’s guide to machine learning](http://www.mckinsey.com/insights/high_tech_telecoms_internet/an_executives_guide_to_machine_learning), 2015

2- مدیری که علاقه مند به ارائه ی یک پروژه است.

این فرد ممکن است یک مدیر پروژه یا چیزی شبیه رهبر پروژه ای باشد که در آن از ماشین لرنینگ استفاده می شود.کاربرد ماشین لرنینگ می تواند یک ویژگی برجسته یا هسته ی اصلی پروژه ی مورد نظر باشد.

منابعی که در این بخش مفید خواهند بود، می تواند داشتن یک دید سطح بالا بر این زمینه باشد که بتواند کلاس های مختلف مسئله و الگوریتم ها را بدون اینکه زیاد وارد جزئیات آن ها شود، ببیند.

کتاب های زیر را بررسی کنید:

[Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die](https://amzn.to/31UcgYY)

[Data Science for Business: What you need to know about data mining and data-analytic thinking](https://amzn.to/3askDif)

[Data Smart: Using Data Science to Transform Information into Insight](https://amzn.to/2Y0HuMX)

* قبایل دانشگاهی

به طور کلی ، این افراد علاقه مند به یادگیری ماشین از منظر دانشگاهی هستند. ممکن است دانشجو (کارشناسی یا کارشناسی ارشد) باشند و یا به نوعی با دانشگاه در ارتباط باشند.

مثال های دیگر این افراد، محققان پسا دکترا،دستیاران پژوهشی و سخنرانان حوزه های مختلف است.

قبایل دانشگاهی ممکن است در مقالات تحقیقاتی خود زمان زیادی را صرف تحقیق در مورد یک الگوریتم ماشین لرنینگ خاص کنند.

3- دانشجوی ماشین لرنینگ در یک کلاس کارشناسی یا کارشناسی ارشد

یک دانشجوی ماشین لرنینگ به احتمال زیاد در یک کلاس شرکت می کند و به سوالات بیش از حد خاص مربوط به تکنیک ها و الگوریتم ها علاقه مند است.یک دانشجو ساختاری برای متمرکز شدن و زمانی برای عمیق تر غوطه ور شدن در مطالب دارد.

بهتر است آن ها با یک کتاب کار کنند.برخی از بهترین کتاب های حوزه ماشین لرنینگ عبارتند از:

[Machine Learning: A Probabilistic Perspective](https://amzn.to/3iFRTWc)

[Pattern Recognition and Machine Learning](https://amzn.to/2Y1s76G)

[The Elements of Statistical Learning: Data Mining. Inference. and Prediction](https://amzn.to/31SA3bt)

4– یک محقق در حوزه ی ماشین لرنینگ که علاقه مند به تاثیر گذاری در این زمینه است

یک محقق ماشین لرنینگ علاقه مند است تا به درک عمیقی از یکی از جنبه های ماشین لرنینگ برسد تا جایی که بتواند جزئیات بیشتری برای توسعه ی این حوزه به وجود آورد.

این محقق به مقاله های پژوهشی ، مجلات و سازمان ها و شبکه هایی که همراه آنها هستند علاقه مند است.کتاب های درسی به درد این بخش نمی خورند.چون منابع فدیمی و دست دوم به حساب می آیند. برخی از مجلات و مقالات سطح بالا مرتبط با ماشین لرنینگ عبارتند از:

[Journal of Machine Learning Research](http://www.jmlr.org/) (JMLR)

[Neural Information Processing Systems](https://nips.cc/) (NIPS)

[Knowledge Discovery and Data Mining](http://www.kdd.org/) (SIGKDD)

[International Conference on Machine Learning](https://en.wikipedia.org/wiki/International_Conference_on_Machine_Learning) (ICML)

5-یک محقق عمومی که علاقه مند است مسائل و مشکلات آن ها را مدل سازی کند.

یک محقق کلی ممکن است به یادگیری ماشین علاقه داشته باشد ، اما به عنوان یک ابزار.

آنها به احتمال زیاد علاقه مند به ساختن یک مدل توصیفی یا پیش بینی با استفاده از داده های خود هستند.

به عنوان مثال ، محققی از حوزه تحقیقات مشتریان ، زمین شناسی یا زیست شناسی مجموعه داده های خاص خود را دارد و به دنبال ایجاد مدلی برای پیش بینی و یا درک بهتر مسئله اساسی حوزه ی خود است.

آنها معمولاً کمتر به دقت مدل علاقه مند هستند و بیشتر به قابل توضیح بودن مدل علاقه دارند.بنابراین ، روشهای ساده تری که از آمار وام گرفته شده و به خوبی درک شده اند،ترجیح داده می شوند.به عنوان مثال رگرسیون خطی و رگرسیون لجستیک.

با این اوصاف روند منظم و مطلوبی مورد نظر است*.*

به این دسته، من منابع "قبایل مهندسی" به خصوصDeveloper Interested in Delivering One-Off Predictions را توصیه می کنم.همچنین به گروه "دانشمندان داده" زیر مجموعه "قبایل داده" هم نگاهی بیاندازید.

* قبایل مهندسی

به طور کلی گروهی از توسعه دهندگان هستند که قصد دارند به وسیله ی نرم افزار، راه حل هایی برای یک سری مسائل ارایه دهند و قصد دارند آن را با ماشین لرنینگ ادغام کنند.

قبایل مهندسی می توانند در جوامع ماشین لرنینگ مثل انجمن های پرسش و پاسخ، از کمک و پشتیبانی زیادی برخوردار شوند.

6- برنامه نویسی که به الگوریتم های در حال اجرا علاقه مند است.

یک روش عالی برای یک برنامه نویس برای اینکه مهارت های ماشین لرنینگ خود را توسعه دهد،این است که با مهارت های برنامه نویسی ای که دارد،الگوریتم های ماشین لرنینگ را از ابتدا پیاده سازی کند.

سه کتابی که من برای این رویکرد توصیه می کنم، عبارتند از:

[Data Science from Scratch: First Principles with Python](https://amzn.to/31R8zmN)

[Machine Learning in Action](https://amzn.to/3gTgm9X)

[Machine Learning: An Algorithmic Perspective](https://amzn.to/2E7wMwJ)

7 - برنامه نویس علاقه مند به ارائه پیش بینی های یکبارمصرف

یک توسعه دهنده لزوماً یک برنامه نویس عالی نیست و برای تهیه و ارائه یک مدل پیش بینی دقیق و قابل اعتماد هم برنامه نویسی لازم نیست.

یک مدل پیش بینی یکبار مصرف ممکن است در محیط بیزینسی برای مجموعه ای از پیش بینی هایی که می تواند فراهم کند نیاز باشد. این بخش همچنین یک مدل قدرتمند برای خودآموزی ، از طریق کار با مجموعه داده های تمرینی و حتی مسابقات ماشین لرنینگ است. با کار سیستماتیک از طریق پردازش یک مسئله و ارائه یک مدل مستقل ، می توانید چیزهای زیادی یاد بگیرید.

[Data Mining: Practical Machine Learning Tools and Techniques](https://amzn.to/340LRLA)

[Applied Predictive Modeling](https://amzn.to/3iFPHhq)

8- مهندس علاقه مند به توسعه نرم افزارها و خدمات هوشمند تر

یک مهندس که علاقه مند است ماشین لرنینگ را به پروژه ی نرم افزاری خود اضافه کند، نیاز به دانش کمی در مورد الگوریتم ها ، نحوه کار کردن مشکلات end-to-end و نحوه اجرای قابل اعتماد الگوریتم در یک محیط عملیاتی دارد.

این گروه از آدم ها از دو دسته ی قبلی که توصیف شدند رشد می کنند و این شاید بهترین توصیف به عنوان یک مهندس ماشین لرنینگ باشد.

آنها به دنبال استفاده از الگوریتم های سریعی هستند که نتایج دقیقی را ارائه می دهند ، و سعی می کنند این دغدغه ها را متعادل کنند.

این گروه همچنین شدیدا از کتابخانه ها و زیرساخت های ماشین لرنینگ استفاده می کنند.

برخی از منابعی که برای شروع به کار در ماشین لرنینگ مفید هستند:

[Building Machine Learning Systems with Python](https://amzn.to/33ZK21B)

[Learning scikit-learn: Machine Learning in Python](https://amzn.to/3fYJLOD)

[Practical Data Science with R](https://amzn.to/31X4fCo)

[Machine Learning with R](https://amzn.to/2CtJ8yV)

* قبایل داده

به طور کلی ، این ها گروهی از افراد هستند که در درجه اول در نقش های مربوط به داده ها هستند اما ممکن است لازم باشد از ماشین لرنینگ استفاده کنند.

9- دانشمند داده ای که به ارائه ی پاسخ های بهتر به پرسش های حوزه ی کسب و کار علاقه مند است

وقتی دانشمند داده باشید ، یادگیری متوقف نمی شود.

شما باید در بالاترین سطح از آخرین جریان داده ها ، تکنیک ها و الگوریتم ها باشید. این مسئله شامل تکنیک های ماشین لرنینگی است که برای توصیف داده ها و ایجاد مدل های پیش بینی به آنها نیاز دارید.

دانشمند داده می تواند آنچه را که نیاز دارند از منابع کاربردی ذکر شده در بخش "قبایل مهندسی" و همچنین منابع نظری بیشتری که در "قبایل دانشگاهی" ذکر شده است ، تهیه کند.

با این وجود ، برخی از منابع ماشین لرنینگ با محوریت علوم داده شامل ترکیب زیر هستند:

[Applied Predictive Modeling](https://amzn.to/3iFPHhq)

[An Introduction to Statistical Learning: with Applications in R](https://amzn.to/3gYt0V9)

[Machine Learning for Hackers](https://amzn.to/3kNsV92)

10 – تحلیلگر داده ای که علاقه مند است داده ها را بهتر توضیح دهد.

تحلیل گران داده در وهله اول علاقه مند به توضیح داده ها در زمینه منافع تجاری هستند. گاهی اوقات الگوریتم های ماشین لرنینگ برای ارائه مدل های قدرتمندتر مفید هستند. بیشتر مدل های این بخش توصیفی است ، اما در بعضی مواقع پیش بینی هم صورت می گیرد.

مانند گروه "محقق عمومی" ، این گروه احتمالاً از نظر آماری و استنباط آماری از بنیان خوبی برخوردار است. همچنین ، با توجه به اینکه آنها احتمالا به یک مدل توصیفی علاقه مند هستند ، روشهای کلاسیک مثل رگرسیون خطی و لجستیک ممکن است کافی باشد.مدل حاصل باید درعین حالی که دقت زیادی دارد،قابلیت توضیح داشته باشد.

بسیاری از منابع مشابه بالا می توانند مفید باشند ، البته شاید بیشتر آنهایی که به استنباط آماری مربوط می شوند.

قدم دوم : یک فرآیند را انتخاب کنید.

برای حل مشکلات از یک فرآیند سیستمیک استفاده کنید.

* مراحل به کارگرفتن ماشین لرنینگ

فرآیند سیستماتیک برای کار با مشکلات مربوط به مدل سازی پیش بینی، که نتایجی بالاتر از میانگین را ارائه می دهد.

با گذشت زمان،با کارکردن بر روی مسائل کاربردی ماشین لرنینگ،شما الگو یا فرآیندی خوب و قدرتمند را برای دستیابی سریع به نتایج توسعه خواهید داد.

بعد از توسعه، شما می توانید این فرآیند را دوباره و دوباره روی پروژه های خود یکی،پس از دیگری استفاده کنید.

هرچه فرآیند شما قوی تر و توسعه یافته تر باشد ، سریعتر می توانید به نتایج قابل اطمینان برسید.

در این جا قصد دارم اسکلت و شاکله ی اصلی فرآیند خودم را برای کار با یک مسئله ی ماشین لرنینگ با شما در میان بگذارم.می توانید از آن به عنوان یک نقطه ی شروع یا قالب برای پروژه ی بعدی تان استفاده کنید.

پنج قدم برای یک فرآیند سیستماتیک

1 – مسئله را تعریف کنید.

2 – داده ها را آماده کنید.

3 – الگوریتم ها را نقطه به نقطه بررسی کنید.( Spot Check)

4 – نتایج را بهبود دهید.

5 – نتایج را ارائه دهید.

انعطاف پذیری زیادی در این فرآیند وجود دارد. به عنوان مثال ، مرحله "آماده سازی داده ها" به طور معمول به دو مرحله تجزیه و تحلیل داده ها (خلاصه کردن و رسم نمودارها) و آماده سازی داده ها (آماده سازی نمونه ها برای آزمایش ها) تقسیم می شود. مرحله "بررسی نقاط الگوریتم ها" ممکن است شامل چندین آزمایش رسمی باشد.

این فرآیند مثل یک خط تولید واقعا بزرگ است که سعی می کنم به صورت خطی از آن عبور کنم.

نکته مهم در استفاده از ابزارهای خودکار این است که می توانید چند مرحله به عقب برگردید (مثلا بگویید از مرحله ی "بهبود نتایج"برگردیم به "آماده سازی داده ها") و یک شکل جدید از دیتاست را وارد کنید و آزمایشات را دوباره در مراحل مداخله انجام دهید تا ببینید چه نتایج جالبی خارج می شوند و چگونه آن هارا با آزمایشاتی که قبلا انجام داده بودید مقایسه می کنند.



BMW production line

فرایندی که استفاده می کنم از فرآیند استاندارد کاوش داده ها از "کشف دانش در دیتابیس ها" اقتباس شده است.(KDD)

^ تعریف مسئله

من می خواهم از یک فرآیند سه مرحله ای برای تعریف مسئله استفاده کنم.

دوست دارم سریع حرکت کنم و از این روند کوچک برای دیدن مشکل از چند منظر مختلف خیلی سریع استفاده می کنم:

**قدم اول:مشکل چیست؟** مسئله را به صورت غیررسمی و رسمی شرح دهید و مفروضات و مشکلات مشابه را فهرست کنید.

**قدم دوم:چرا مشکل باید حل شود؟** انگیزه خود را برای حل مسئله ، مزایایی که یک راه حل برای ما فراهم می کند و همینطور نحوه استفاده از راه حل ذکر کنید.

**قدم سوم:چطور می توان مشکل را حل کرد؟** توضیح دهید که چطور مشکل برای متعادل کردن دامنه ی دانش به صورت دستی حل خواهد شد.

^ آماده سازی داده ها

من مقدمه ی آماده سازی داده ها را با یک فاز تجزیه و تحلیل داده ها که شامل خلاصه کردن feature ها و visualization آن ها به وسیله ی نمودارهای scatter و histogram است انجام می دهم.من همچنین دوست دارم جزئیات هر feature و ارتباط بین feature ها را توضیح دهم.

این کار ناخوشایند من را مجبور می کند قبل از اینکه داده ها در الگوریتم ها گم شوند ، در راستای مسئله ی خود در مورد آنها فکر کنم.

در واقع آماده سازی داده ها،در سه مرحله به صورت زیر انجام می شود:

1 - **Data Selection : در نظر بگیرید که چه داده ای موجود است، چه داده ای missing value است و چه داده ای می تواند حذف شود.**

2 - **Data Preprocessing :** داده های انتخابی خود را با قالب بندی ، تمیز کردن و نمونه برداری از آنها سازماندهی کنید.

3 - **Data Transformation : transform کردن** داده های پیش پردازش شده آماده برای یادگیری ماشین با استفاده از مهندسی کردن feature ها به وسیله ی scaling، و همچنین تراکم و تجزیه ی feature ها.

^ بررسی نقطه به نقطه الگوریتم ها

من به طور پیش فرض از ضریب 10 برابر برای تهیه ی تست های خود استفاده کرده ام. تمام آزمایش ها (الگوریتم و ترکیب های مربوط به دیتاست) 10 بار تکرار می شوند و میانگین و انحراف معیارِ هر کدام از دقت ها جمع آوری شده و گزارش می شود.

من همچنین از آزمون های اهمیت آماری برای استخراج نتایج معنی دار از داده های نویز استفاده می کنم."باکس پلات" ها برای خلاصه کردن توزیع ها از نتایج دقیق هر جفتی از الگوریتم و دیتاست بسیار مفید هستند.

Spot check کردن الگوریتم ها به این معنی است که من دسته ای از الگوریتم های یادگیری ماشین استاندارد را در آزمایش خود امتحان می کنم و یک آزمایش رسمی انجام می دهم. من معمولاً 10-20 الگوریتم استاندارد را از بین خانواده الگوریتم های اصلی بر روی تمام نسخه های استانداردشده و مقیاس بندی شده ی دیتاستی که آماده کرده ام، اجرا می کنم.

هدف از spot check کردن این است که تمام ترکیب های الگوریتم و دیتاستی که برای انتخاب کردن ساختارهای مناسبی از مسئله که می توانند با آزمایشات متمرکز، با جزئیات بیشتری مورد مطالعه قرار بگیرند، مناسب هستند را استخراج کنیم.

آزمایشات متمرکزتر با خانواده های الگوریتم های با عملکرد خوب ممکن است در این مرحله انجام شود ، اما تنظیم الگوریتم برای مرحله بعدی باقی مانده است.

آزمایشات با تمرکز بیشتر با استفاده از خانواده هایی از الگوریتم های با عملکردخوب، ممکن است در این مرحله انجام شود؛اما تنظیم الگوریتم برای مرحله ی بعدی باقی مانده است.

^ بهبود نتایج

بعد از spot check کردن، وقت این است که بهترین نتیجه را از وضعیت موجود به دست آورید. من این کار را با استفاده از آنالیز خودکار حساسیت بر روی پارامترهای الگوریتم های با عملکرد بالا انجام می دهم. من همچنین آزمایش ها را با استفاده از متدهای استاندارد گروه الگوریتم های با عملکرد بالا طراحی و اجرا می کنم.

من وقت زیادی گذاشتم تا به این فکر کنم که چگونه می توان از دیتاست یا خانواده الگوریتم هایی که نشان داده اند عملکرد خوبی دارند ، بیشتر بهره برد.

باز هم ، اهمیت آماری نتایج در اینجا حیاتی است.تمرکز روی روش ها و بازی با پیکربندی های الگوریتم بسیار آسان است.نتایج تنها درصورتی معنادار خواهند بود که قابل توجه باشند، تمام پیکربندی ها قبلا فکر شده باشند و آزمایشات به صورت دسته ای اجرا شده باشند.همچنین من مایلم که مجموعه ای از بهترین نتایج خودم را که بر روی یک مسئله به دست آورده ام،حفظ کنم.

به طور خلاصه پروسه ی بهبود نتایج شامل موارد زیر است:

* **تنظیم الگوریتم:** که در آن با موضوع کشف بهترین مدل،مثل یک مسئله ی جستجو در میان فضای پارامتری مدل برخورد می شود.

**متدهای گروهی :** جایی که پیش بینی های انجام شده توسط چندین مدل با هم ترکیب می شوند.

**مهندسی حدود ویژگی ها:** جایی که تجزیه و تجمع مربوط به ویژگی ها که در مرحله ی آماده سازی داده ها دیده شده است، به محدودیت هایی بربخورد.

^ ارائه ی نتایج

نتایج یه مسئله ی پیچیده ی ماشین لرنینگ تا وقتی که به کار گرفته نشوند، معنی نخواهند داشت.این به طور معمول به معنی ارائه به سهامداران و ذینفعان آن پروژه است. حتی اگر این مسئله مربوط به یک مسابقه باشد یا مسئله ای باشد که خودم آن را برای خودم طرح کرده ام، باز هم مراحل ارائه نتایج را طی خواهم کرد. این یک روش خوب است و به من یادگیری واضحی می دهد که می توانم دفعه بعد بر اساس آن کار کنم.

قالبی که من برای ارائه نتایج از آن استفاده می کنم به صورت زیر است و ممکن است به صورت یک سند متنی ، گزارش رسمی یا ارائه در قالب اسلایدهای پاورپوینت باشد:

* **زمینه(چرا):** محیطی را که مشکل در آن وجود دارد تعریف کنید و انگیزه سوال تحقیق را تنظیم کنید.
* **مسئله(پرسش):** به طور خلاصه مسئله را به عنوان سوالی تشریح کنید که در پی آن رفته و به آن پاسخ داده اید.
* **راه حل(پاسخ):** به طور خلاصه راه حل را به عنوان پاسخی که به سوال بخش قبل داده اید، تشریح کنید.خاص باشید.
* **یافته ها:** لیست های جمع و جوری از یافته های خود را در قالب مسیری که مخاطب به آن علاقه مند است، ارایه کنید. این موارد ممکن است شامل کشف هایی در داده ها ، روشهایی که کار کرده یا نکرده اند و یا مزایای عملکردیِ مدلی که شما در طی تلاش خود به دست آورده اید باشد.
* **محدودیت ها:** جاهایی که مدل در آنجا کار نمی کند یا سوالاتی که به آن ها پاسخ نمی دهد را مطرح کنید.از این سوالات فرار نکنید.توضیح دادن در مورد جاهایی که مدل برتری دارد،قابل اعتمادتر است؛ اگر شما می توانید توضیح دهید که در کجاها برتری ندارد.
* **نتیجه گیری(چرایی + پرسش + پاسخ):** بازبینی "چرایی"،پرسش ها و پاسخ هایی که شما کشف کرده اید در یک بخش کوچک و مختصر، یادآوری و تکرار آن را برای خودتان و دیگران آسان می کند.

قدم سوم : یک ابزار انتخاب کنید.

ابزاری متناسب سطح خود انتخاب کرده و آن را روی پروسه ی خود اجرا کنید.

مبتدیان: WEKA WORKBENCH

در چهارده روز به یک تمرین کننده ی ماشین لرنینگ تبدیل شوید.

ماشین لرنینگ یک مبحث مطالعاتی جذاب است ، اما در واقع چگونه می توانید از آن برای مسائل خود استفاده کنید؟

ممکن است سردرگم شوید که چگونه داده های خود را به بهترین شکل برای ماشین لرنینگ آماده کنید ، از کدام الگوریتم ها استفاده کنید یا اینکه چگونه یک مدل را نسبت به مدل دیگر انتخاب کنید.

در این بخش شما یک دوره 14 قسمتی قدم به قدم را برای یادگیری ماشین لرنینگ کاربردی با استفاده از پلتفرم Weka خواهید آموخت.بدون حتی یک معادله ریاضی یا یک خط برنامه نویسی.

بعد از تکمیل این دوره کوتاه:

* شما خواهید آموخت که چگونه برای ارائه ی مجموعه ای از پیش بینی ها یا یک مدل با عملکرد بالا، به صورت end-to-end با یک دیتاست کار کنید.
* روش کار با پلتفرم ماشین لرنینگ weka را که شامل چگونگی انجام پیمایش بین الگوریتم ها و طراحی آزمایش های کنترل شده است را خواهید آموخت.
* یاد خواهید گرفت که چگونه چندین دیدگاه از مسئله خود ایجاد کنید ، چندین الگوریتم را ارزیابی کرده و از آمار برای انتخاب بهترین مدل برای مسائل مدل سازی پیش بینی خود استفاده کنید.

این دوره ی کوتاه برای چه کسانی است؟

قبل از شروع ، بیایید مطمئن شویم که در جای مناسب قرار گرفته اید. لیست زیر برخی از دستورالعمل های کلی در مورد اینکه این دوره برای چه کسی طراحی شده ارائه می دهد.

اگر دقیقاً با این نقاط مطابقت ندارید وحشت نکنید؛ ممکن است فقط لازم باشد تا در یک ناحیه یا دیگر نواحی دانسته های خود را افزایش دهید تا آماده شوید.

شما یک توسعه دهنده هستید که کمی ماشین لرنینگ می دانید.

این بدان معنی است که شما در مورد تعدادی از پایه های ماشین لرنینگ مانند cross-validation ، تعدادی از الگوریتم ها و مباحث واریانس چیزهایی می دانید.

این به آن معنا نیست که شما دکترای ماشین لرنینگ دارید ، فقط در این اندازه که این مفاهیم مهم را بدانید یا بدانید کجا باید آنها را جستجو کنید.

این دوره کوتاه، کتاب درسی ماشین لرنینگ نیست.

این دوره شما را از توسعه دهنده ای که کمی ماشین لرنینگ می داند، به توسعه دهنده ای تبدیل می کند که می تواند از پلتفرم Weka برای کار با یک دیتاست از ابتدا تا انتها استفاده کند و مجموعه ای از پیش بینی ها یا یک مدل با عملکرد بالا را ارائه دهد.

بررسی اجمالی این دوره(چه انتظاری از آن می رود؟)

این دوره ی کوتاه به چهارده قسمت تبدیل می شود. هر درس طوری طراحی شده که حدود 30 دقیقه وقت شما را بگیرد. ممکن است بعضی ها را خیلی زودتر تمام کنید و برای بعضی دیگر ترجیح دهید عمیق تر شوید و زمان بیشتری را سپری کنید.

شما می توانید هر قسمت را با سرعت یا به آرامی که دوست دارید کامل کنید. یک برنامه راحت که به شدت پیشنهاد می شود، تکمیل هر درس در یک روز، در طی یک دوره دو هفته ای است.

مباحثی که طی 14 درس بعدی پوشش خواهید داد به شرح زیر است:

1-دانلود و نصب برنامه

2-بارگیری دیتاست های استاندارد ماشین لرنینگ

3-توصیف آماری و visualization

4- rescale کردن داده ها

5- انجام feature selection بر روی داده ها

6- الگوریتم های ماشین لرنینگ در weka

7- تخمین عملکرد مدل

8- عملکرد پایه ای بر روی داده ها.

9-الگوریتم های classification

10-الگوریتم های regression

11- الگوریتم های گروهی

12-مقایسه ی عملکرد الگوریتم ها

13-تنظیم پارامترهای الگوریتم

14-ذخیره ی مدل

این دوره بسیار سرگرم کننده خواهد بود.فقط مجبورید یک سری کارها را انجام دهید،کمی مطالعه و جستجو در مورد weka.

1-دانلود و نصب برنامه

اولین کاری که باید انجام دهید این است که نرم افزار Weka را روی دستگاه خود نصب کنید.

Weka یک نرم افزار متن باز رایگان است. به زبان جاوا نوشته شده است و می تواند روی هر پلتفرمی که از جاوا پشتیبانی می کند از جمله windows،linux و Mac OS اجرا شود. می توانید Weka را به عنوان یک نرم افزار مستقل یا نسخه ای همراه با Java دانلود کنید.

اگر قبلا جاوا روی سیستم خود نصب نکرده اید ، توصیه می کنم نسخه ای همراه با جاوا را بارگیری و نصب کنید.

2-بارگیری دیتاست های استاندارد ماشین لرنینگ

اکنون که Weka را نصب کرده اید ، باید داده ها را دانلود کنید. Weka برای دانلود داده ها در قالب محلی ای به نام ARFF طراحی شده است. این قالب،یک قالب CSV اصلاح شده است که شامل اطلاعات بیشتری در مورد انواع مختلفی از هر ویژگی (ستون) است.

نسخه Weka شامل یک زیر شاخه با تعدادی از دیتاست های استاندارد ماشین لرنینگ در قالب ARFF می باشد که آماده ی دانلود است.

Weka همچنین از دانلود داده های خام CSV به طور مثال از یک دیتابیس پشتیبانی کرده و در صورت لزوم داده ها را به ARFF تبدیل می کند.

در این درس شما یک دیتاست استاندارد را در Weka Explorer بارگیری خواهید کرد.

سعی کنید چندین دیتاست را در محیط برنامه بارگیری کنید.همچنین چندین فایل CSV را به محیط برنامه وارد کنید.

3 – توصیف آمار و مصورسازی داده ها

بعد از اینکه داده ها را وارد نرم افزار کردید، مهم است که نگاهی به آن ها بیاندازید.Weka این امکان را به شما می دهد که آمار توصیفی محاسبه شده ای را از داده های خود مشاهده کنید.از طرفی ابزارهایی را هم برای مصورسازی داده ها در اختیارتان قرار می دهد.این ویژگی ها را می توان در صفحه preprocessing و همچنین سربرگ visualize مشاهده کرد.

4 – Rescale کردن داده ها

داده ی خام معمولا برای مدل سازی مناسب نیست.اغلب می توانید با Rescale کردن ویژگی های خود، عملکرد مدل های ML خود را بهبود دهید.

در این درس می آموزید که چطور از فیلترهای Weka برای تنظیم مجدد داده های خود استفاده کنید.تمام داده های خود را برای یک دیتاست نرمال کرده و سپس آن ها را در یک محدوده مشخص بین 0 و 1، Rescale خواهیم کرد.

+ بعد از وارد کردن دیتاست، از گزینه ی Choose در بخش Filters استفاده کرده و سپس unsupervised.attribute.Normalize را انتخاب کنید.با زدن کلید Apply تغییرات اعمال شده و فیلتر فعال می شود.

از همان مسیر می توان سایر فیلترها را هم بررسی کرده و تنظیمات مربوط به هر کدام را هم به صورت دلخواه تغییر داد.در انتها به کمک گزینه Save میتوان تغییرات صورت گرفته را بر روی دیتاست خود ذخیره کرده تا دفعات بعد بتوان از آن ها استفاده کرد .

5- انتخاب ویژگی را برای داده های خود انجام دهید.(feature selection)

ممکن است همه ویژگی های مجموعه داده شما با موضوعی که می خواهید پیش بینی کنید مرتبط نباشند. برای شناسایی ویژگی هایی که بیشترین ارتباط را با متغیر خروجی شما دارند می توانید از feature selection استفاده کنید.

+ بعد از وارد کردن دیتاست خود، سربرگ Select Attributes را انتخاب کنید.

در صفحه ی Attribute Evaluator روی گزینه ی Choose کلیک کرده و سپس CorrelationAttributeEval را بزنید.

صفحه ای باز می شود که از شما می پرسد متد Ranker فعال شود یا خیر.روی Yes کلیک کنید.

در انتها کلید Start را بزنید تا متد انتخاب ویژگی اجرا شود.

امتیازات وابستگی ویژگی های مختلف در پنجره Attribute selection output نمایش داده می شود.امتیازات بالاتر،نشان دهنده ی ارتباط بیشتر است.می توان متدهای دیگری که برای این منظور وجود دارند را هم بررسی کرد و سپس ویژگی هایی که ارتباط و استفاده ی کمتری داشتند را در پنجره ی Preprocessing به کمک دکمه Remove حذف کرد.

6- الگوریتم های یادگیری ماشین در weka

یک مزیت کلیدی این نرم افزار، تعداد زیاد الگوریتم های ماشین لرنینگ است که در اختیار شما قرار می دهد.شما باید مسیر مورد نظر خود را در بین الگوریتم های یادگیری ماشین پیدا کنید.در این درس نگاهی نزدیک تر به این الگوریتم ها خواهیم انداخت:

+ بعد از وارد کردن دیتاست، بر روی سربرگ Classify کلیک کنید.کلید Choose را زده و الگوریتم مورد نظر خود را از لیست انتخاب کنید.با کلیک بر روی نام الگوریتم، پنجره configuration باز می شود.کلید More شامل اطلاعات بیشتری در مورد اجرای الگوریتم، و کلید Capabilities اطلاعاتی در مورد چگونگی استفاده از آن است.توجه داشته باشید که کلیدهای open و save در این پنجره می توانند تنظیمات مختلفی را برای هر الگوریتم ذخیره و اجرا کنند.با قرارگیری روی هر پارامتر، یک جمله Help به نمایش در می آید که می تواند راهنمای شما باشد.با کلیک روی دکمه start الگوریتم اجرا می شود.

+ الگوریتم های در دسترس را بررسی کنید.دقت داشته باشید که با توجه به اینکه مسئله ی شما مربوط به classifier (پیش بینی یک دسته بندی) یا regression (پیش بینی یک عدد حقیقی) باشد، ممکن است بعضی از الگوریتم ها در دسترس نباشند. با الگوریتم های مختلفی که در weka وجود دارند بیشتر کار کرده و در مورد آن ها بیشتر بیاموزید.در زمینه ی انتخاب و پیکربندی تنظیمات مختلف الگوریتم های موجود، اعتماد به نفس پیدا کنید.

7- عملکرد مدل را تخمین بزنید.

اکنون که می دانید چگونه الگوریتم های مختلف را انتخاب و پیکربندی کنید ، باید بدانید که چگونه عملکرد یک الگوریتم را ارزیابی کنید. در این درس شما با روش های مختلف ارزیابی عملکرد یک الگوریتم در Weka آشنا می شوید.

+ بعد از بازکردن یک دیتاست در محیط برنامه، روی سربرگ Classify کلیک کنید.

قسمت Test options شامل لیست تکنیک های مختلفی است که می توانید برای ارزیابی عملکرد یک الگوریتم استفاده کنید. استاندارد طلایی برای Cross Validation مقدار 10 است اما برای دیتاست های کوچکتر می توان آن را روی 5 یا حتی 3 هم تنظیم کرد.

اگر دیتاست تان خیلی بزرگ است و شما می خواهید الگوریتم ها را به سرعت ارزیابی کنید، می توانید از گزینه ی Percentage split استفاده کنید.به طور پیش فرض این مورد از 66 درصد دیتاست شما برای train مدل تان استفاده می کند و 34 درصد مابقی را برای ارزیابی مدل به کار می گیرد.

همچنین اگر دیتاست و فایل مجزایی دارید که می خواهید تست مدل را با آن انجام دهید، می توانید از گزینه ی Supplied test set استفاده کنید.با این کار مدل شما با تمام دیتاست ای که در ابتدا وارد کرده بودید، آموزش می بیند و سپس از فایل دوم برای ارزیابی مدل استفاده می کند.

نهایتا می توانید از همان دیتاست ای که برای آموزش استفاده کرده بودید، برای ارزیابی هم استفاده کنید. این کار در مواردی که شما بیشتر به یک مدل توصیفی علاقه مندید تا یک مدل برای پیش بینی، می تواند مفید باشد.

وقتی کلید start را بفشارید، الگوریتم انتخاب شده با تنظیمات مشخصی که برای ارزیابی آن انتخاب کرده بودید اجرا می شود.گزینه های مختلف ارزیابی را امتحان کرده و تنظیمات آن را در بخش More options اصلاح کنید.

8- عملکرد پایه بر روی داده ها

هنگامی که شروع به ارزیابی چندین الگوریتم یادگیری ماشین در مجموعه داده خود می کنید ، برای مقایسه به یک خط مبنا نیاز دارید. یک نتیجه پایه، به شما یک نقطه مرجع می دهد تا متوجه شوید که نتایج یک الگوریتم خاص به چه میزان خوب یا بد بوده است.

الگوریتم ZeroR یکی از الگوریتم هایی است که می توان به عنوان مبنا به کار گرفت:

+ پس از بازکردن دیتاست مورد نظر، با کلیک روی سربرگ Classify، این الگوریتم به صورت پیش فرض انتخاب شده است.پس روی کلید start کلیک کنید.

از این الگوریتم می توانید برای به دست آوردن یک حالت مبنا برای تمام دیتاست های خود استفاده کنید.نتایج این الگوریتم "بدترین" نتیجه ی ممکن خواهد بود و هر الگوریتمی که نتایج حتی کمی بهتر را ارائه دهد، احتمالا در حل مسئله ی شما کمی مهارت دارد.

در الگوریتم Classification، الگوریتم ZeroR همیشه فراوان ترین گروه را پیش بینی می کند.اگر مجموعه داده تعداد کلاس های برابر داشته باشد ، مقدار دسته اول را پیش بینی می کند.در دیتاست دیابت ، این الگوریتم دقت 65٪ خواهد داشت.

برای مسائل Regression، این الگوریتم همیشه مقدار خروجی متوسط را پیش بینی می کند.الگوریتمZeroR را روی تعداد زیادی از دیتاست ها امتحان کنید.پس این الگوریتم، الگوریتمی است که باید همیشه قبل از بقیه برای به دست آوردن یک نقطه ی مبنا اجرا کنید.

9- تور الگوریتم های مربوط به classification

نرم افزار weka تعداد بسیار زیادی از این الگوریتم ها را در اختیار شما قرار می دهد.

+ پس از بازکردن دیتاست خود و ورود به سربرگ classify، از بخش choose می توانید پنج الگوریتم برتر زیر را برای کلاس بندی داده های خود استفاده کنید:

* Logistic Regression (functions.Logistic).
* Naive Bayes (bayes.NaiveBayes).
* k-Nearest Neighbors (lazy.IBk).
* Classification and Regression Trees (trees.REPTree).
* Support Vector Machines (functions.SMO).

با هر کدام از الگوریتم های بالا کار و تمرین کنید. از آن ها در دیتاست هایی که قرار است داده ها را در دو کلاس یا بیشتر طبقه بندی کنند، استفاده کنید.

10- تور الگوریتم های مربوط به Regression

الگوریتم های کلاس بندی تخصص weka است اما می توان از بسیاری از آن ها برای Regression هم استفاده کرد.رگرسیون، پیش بینی مقادیر واقعی خروجی مثل قیمت به دلار است و با کلاس بندی که مثلا پیش بینی گروه یک سگ از گربه است، تفاوت دارد.

در ادامه با پنج تا از برترین الگوریتم هایی که در زمینه ی رگرسیون می توانید استقاده کنید آشنا خواهیم شد؛اما قبل از آن از طریق آدرس( https://sourceforge.net/projects/weka/files/datasets/ ) تعدادی دیتاست مخصوص رگرسیون دانلود کنید.

+ پس از بارگیری دیتاست مورد نظر در محیط برنامه، بر روی سربرگ Classify کلیک کرده و به کمک کلید Choose، یکی از موارد زیر را انتخاب کنید:

* Linear Regression (functions.LinearRegression).
* Support Vector Regression (functions.SMOReg).
* k-Nearest Neighbors (lazy.IBk).
* Classification and Regression Trees (trees.REPTree).
* Artificial Neural Network (functions.MultilayerPerceptron).

هر کدام از الگوریتم های بالا را روی دیتاست های مختلف امتحان کرده و عملکرد آن ها را بررسی کنید.

11- تور الگوریتم های گروه

یکی از مزیت های weka در مقایسه با سایر پلتفرم های ماشین لرنینگ، سادگی در استفاده و مزیت دیگر، ارائه ی یک مجموعه ی بزرگ از الگوریتم های گروهی است.

وقت گذاشتن برای اینکه الگوریتم های گروهی weka را یاد بگیرید، ارزش آن را خواهد داشت.

در این قسمت با پنج الگوریتم گروهی برتر آشنا خواهید شد:

+ پس از بارگیری دیتاست مورد نظر، از سربرگ Classify، به کمک کلید Choose یکی از الگوریتم های زیر را انتخاب کنید:

* Bagging (meta.Bagging).
* Random Forest (trees.RandomForest).
* AdaBoost (meta.AdaBoost).
* Voting (meta.Voting).
* Stacking (meta.Stacking).

با هر یک از این الگوریتم ها کار کنید.

بیشتر این الگوریتم های گروهی به شما اجازه می دهند تا از دیتاست خود زیرمجموعه هایی را انتخاب کنید.آزمایشات خود را با ترکیبی از زیرمجموعه های مختلف تکرار کنید. ترکیبی از تکنیک ها که به روش های بسیار متفاوتی کار می کنند و پیش بینی های متفاوتی را ایجاد می کنند ، اغلب عملکرد بهتری دارند.آن ها را روی دیتاست های مختلفی از رگرسیون و کلاس بندی امتحان کنید.

12- کارایی الگوریتم ها را مقایسه کنید.

Weka ابزارهای مختلفی را مختص مقایسه ی الگوریتم ها توسعه داده است که اصطلاحا weka Experiment Environment یا "محیط آزمایش" نامیده می شود.

محیط آزمایش به شما این امکان را می دهد تا آزمایش های کنترل شده ای را به وسیله ی الگوریتم های ماشین لرنینگ طراحی و اجرا کرده و سپس نتایج را بررسی و آنالیز کنید.

در این جا اولین آزمایش خود را می سازید و یاد می گیرید چطور از محیط آزمایش weka برای بررسی عملکرد الگوریتم ها استفاده کنید.

+ در صفحه ی ابتدایی نرم افزار کلید Experimenter را برای ورود به محیط آزمایش بزنید.

بر روی کلید new کلیک کنید.با کلید Add new بخش دیتاست ها ابتدا یک دیتاست و سپس با همان کلید در صفحه ی الگوریتم ها، یک الگوریتم مثل ZeroR و Ibk را انتخاب کنید.سربرگ Run را فعال کرده و کلید Start را بزنید.

از سربرگ Analyse ، ابتدا کلید Experiment و سپس Perform test را بزنید.

به این ترتیب شما اولین محیط آزمایشی خود را ساخته و دو الگوریتم مذکور را با تنظیمات پیش فرضی که داشتند، اجرا و بررسی کردید.

با گزینه ی Test base از سربرگ Analyse می توانید مشخص کنید که کدام دسته از نتایج به عنوان مرجع برای مقایسه انتخاب شود.

این کار را برای دیتاست های دیگر و الگوریتم های بیشتر تکرار کنید.

قدم چهارم : روی دیتاست ها تمرین کنید.

قدم پنجم : یک رزومه و مجموعه کار بسازید.