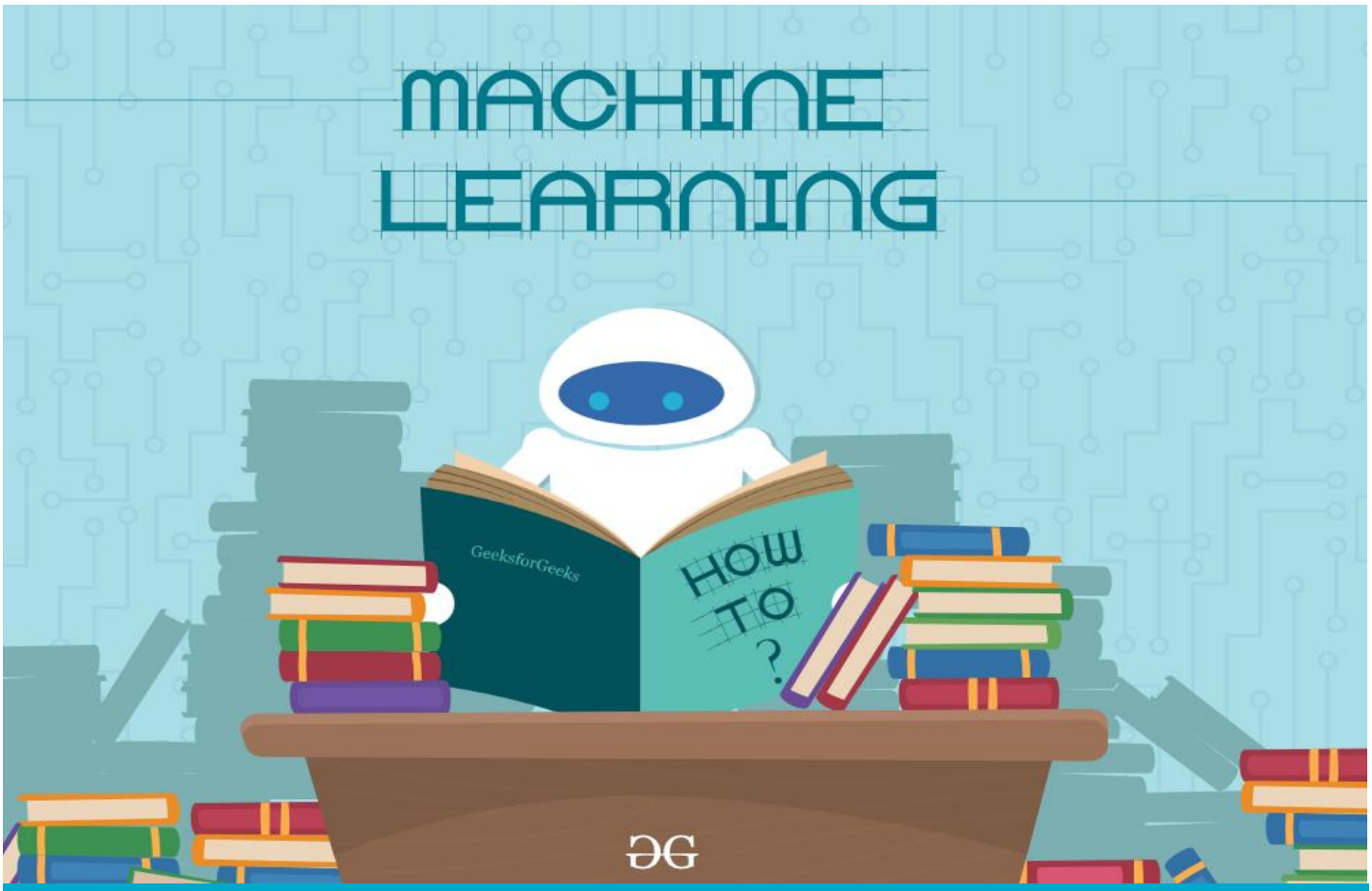


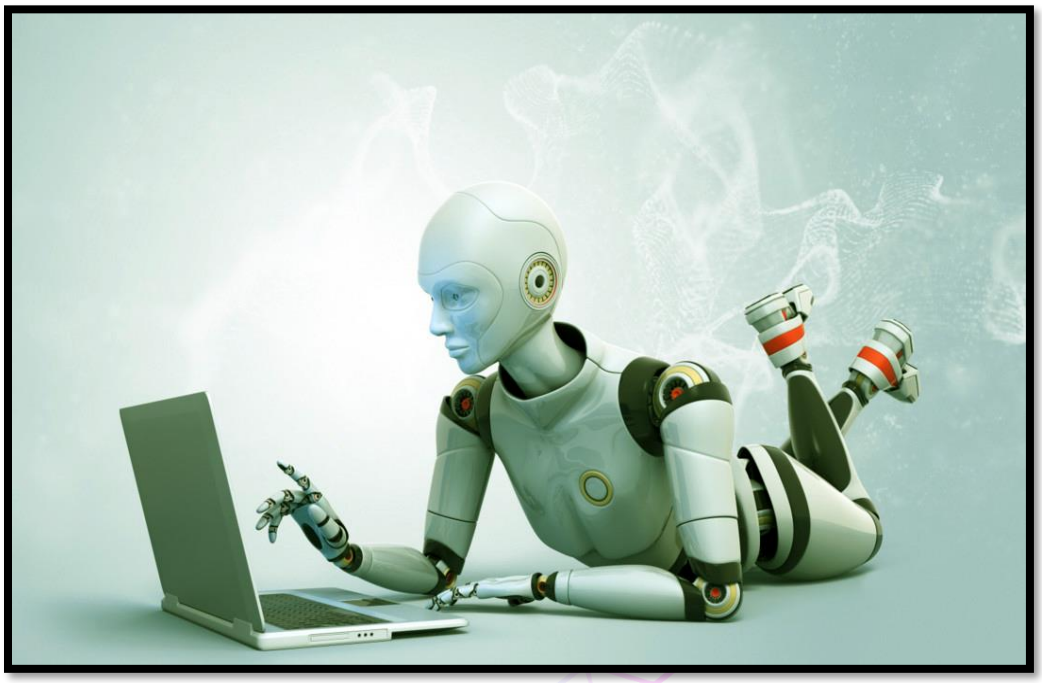

# دوره جامع یادگیری ماشین

قسمت دهم، رویکرد های مناسب برای ارزیابی ماشین و کارکرد آن!

## ارزیابی ماشین

Datalk.ir
Created by : Ali Arabshahi
Contact Us : Linkedin.com/in/mrAliArabshahi

# قدم نهایی در یادگیری ماشین



تا اینجا فهمیدیم یادگیری ماشین چی هست و این که به طور کلی چه فرایند هایی طی میشه. بیاین یه نگاهی بر روی آنچه تا الان یاد گرفتیم، بندازیم:

- یادگیری ماشین درباره هوشمند کردن و بهبود عملکرد ماشین ها در برخی وظایف، توسط یادگیری اون ها به وسیله داده ها به جای یادگیری توسط صرفا برنامه نویسی و ایجاد وظایف مختلف می باشد.
- انواع مختلفی از سیستم های یادگیری ماشین وجود دارد از جمله ( supervised or not, batch (or online, instance-based versus model based
- در یک سیستم یادگیری ماشین ابتدا داده هایی رو آماده و بعد اون ها رو به یک الگوریتم یادگیری تحویل میدیم. اگر سیستم مون model-based باشه، الگوریتم با تنظیم و رسیدن به

یک سری پارامتر ها، یک مدل بهینه به ماشین ارایه میده و اگر هم سیستم مون instance-based باشه، الگوریتم بر اساس شباهت بین داده های ورودی یک چیزی رو یاد می گیره.

- اگر دیتا هامون نمایانگر بدی از کل جامعه و یا همراه با نویز و یا دارای فیچر های به درد نخور باشن، ماشین نتایج مطلوبی رو به دنبال نخواهد داشت.
- و در آخر این که مدلمون نه باید خیلی ساده باشه که به مشکل underfitting بر بخوریم و نه این که اونقدر پیچیده باشه که خطای overfitting پیش بیاد.

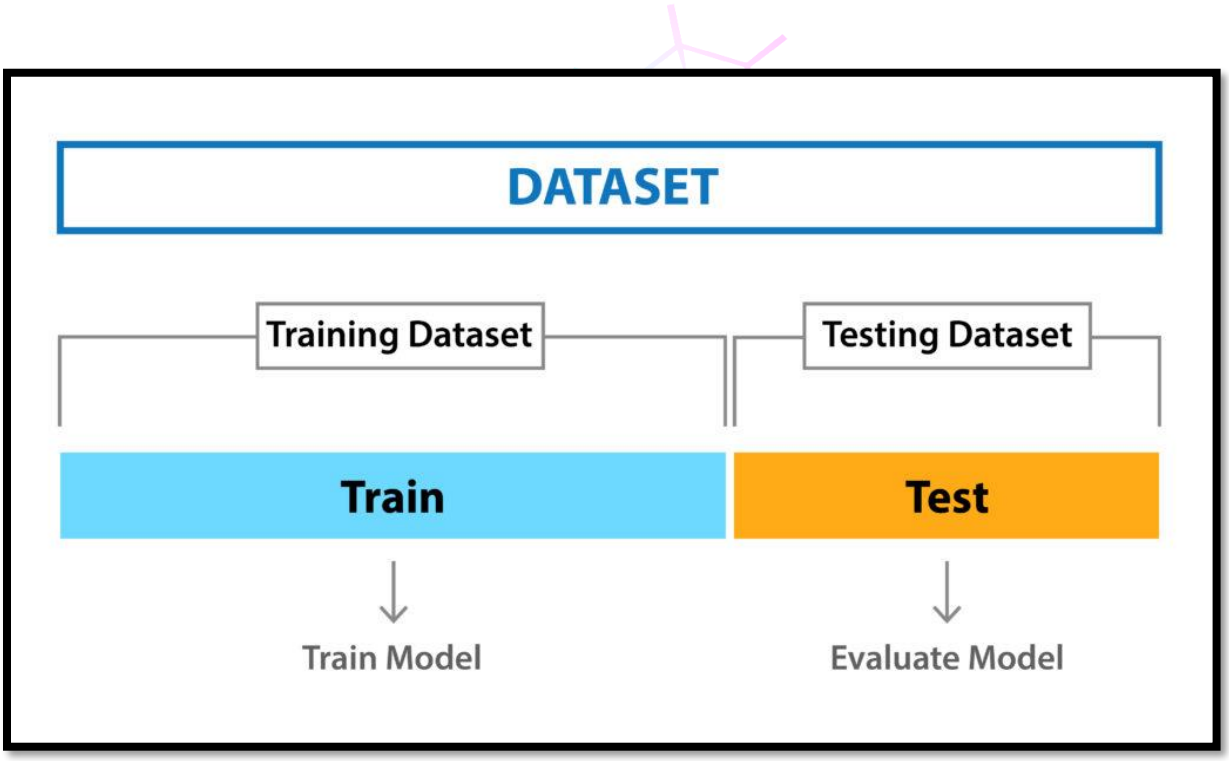
فقط یک موضوع مهم از فرایند یادگیری ماشین باقی مانده که در این جلسه به اون خواهیم پرداخت. زمانی که داده ها رو به مدل آموزش دادیم، این که فقط بگیم امیدواریم مدل روی داده های جدید بتونه خروجی (پیشبینی) خوبی رو بهمون ارایه بده که نمیشه! 😊

باید خودمون دست به کار بشیم و از تعمیم دهی (generalization) و کارکرد صحیح مدل اطمینان پیدا کنیم. بر اساس یک سری از روش ها و همچنین شاخص های عملکردی معین( performance measures)، به این مهم یعنی **ارزیابی عملکرد** ماشینمون دست پیدا خواهیم کرد.

## Testing and Validation

تنها راهی که میشه از کارکرد صحیح ماشین مطمئن شد اینه که اون رو بر روی داده های جدید امتحان کنیم. یکی از رویکردا میتونه این باشه که مدل رو بندازیم رو غلتک! و ببینیم مثلا پیشبینی ماشین از سلیقه کاربرا و فیلم هایی که بهشون پیشنهاد میشه چه قدر صحیحه. اما چنین رویکردی ممکنه باعث دلخوری کاربرامون بشه و در کل هزینه های به سیستم تحمیل کنه .🤐

اما یه راه خیلی باحال اینj که، از همون ابتدا(قبل از این که ماشین رو آموزش بدین) میایم و داده هامون رو به دو بخش تقسیم می کنیم. **Training Data** و **Test Data** .




به این صورت که ماشین رو با Training Data آموزش می دیم و زمانی که به خروجی قابل قبولی رسیدم، عملکرد سیستممون رو بر روی Test دیتا رصد می کنیم. به خطا یا اروری که در این بخش بهش می رسیم، **Generalization error** گفته می شه. اگر این ارور پایین بود، میشه با احتمال زیادی

به این نتیجه رسید که ماشین به خوبی کار می کنه ~😀~ و البته اگر مقدار این ارور بالا بود یعنی ماشین نسبت به training data ، دچار overfitting شده است. (جلسه قبل راجع به این مسئله کلی با هم صحبت کردیم).

معمولا در شیوه های سنتی تر، ۸۰ درصد داده ها رو به عنوان training set و ۲۰ درصد داده ها رو به عنوان test set در نظر می گیرن. البته این کاملا به حجم داده های شما بستگی داره. زمانی که خیلی دیتا دارین، برای مشاهده کارکرد صحیح ماشین، قطعا حجم خیلی کمتری هم از کل داده ها هم می تونه ما رو از این مسئله آگاه کنه.

## Hyperparameter Tuning and Model Selection

پس به نظر میاد ارزیابی مدل کار چندان سختی هم نیست. فقط کافیه از test set استفاده کنیم. حالا بیاین فرض کنیم که ما بین دو تا مدل شک دارین. (مثلا بین مدل خطی ساده و مدل خطی چند جمله ای). چجوری بفهمیم کدومشون بهتره؟ یکی از گزینه ها می تونه این باشه که هر دو تا مدل رو به ماشین آموزش بدیم و بعد ببینیم کدومشون بهترین نتیجه رو روی داده های test set، نشون می دن. (generalize using test set) .

حالا بیاین تصور کنیم که مدل چند جمله ای بهترین خروجی رو به ما تحویل داد. خب بعد از انتخاب نوع مدل نوبت می رسه به این که از overfitting شدن مدل جلوگیری کنیم. همون طور که در جلسه قبل گفتیم، برای این کار می تونیم از عمل regularization بهره ببریم. پس نیاز داریم که ضریب hyper – parameter رو تعیین کنیم (برای جریمه ی پارامتر ها و کاهش اثرشون). اما چطوری بهترین

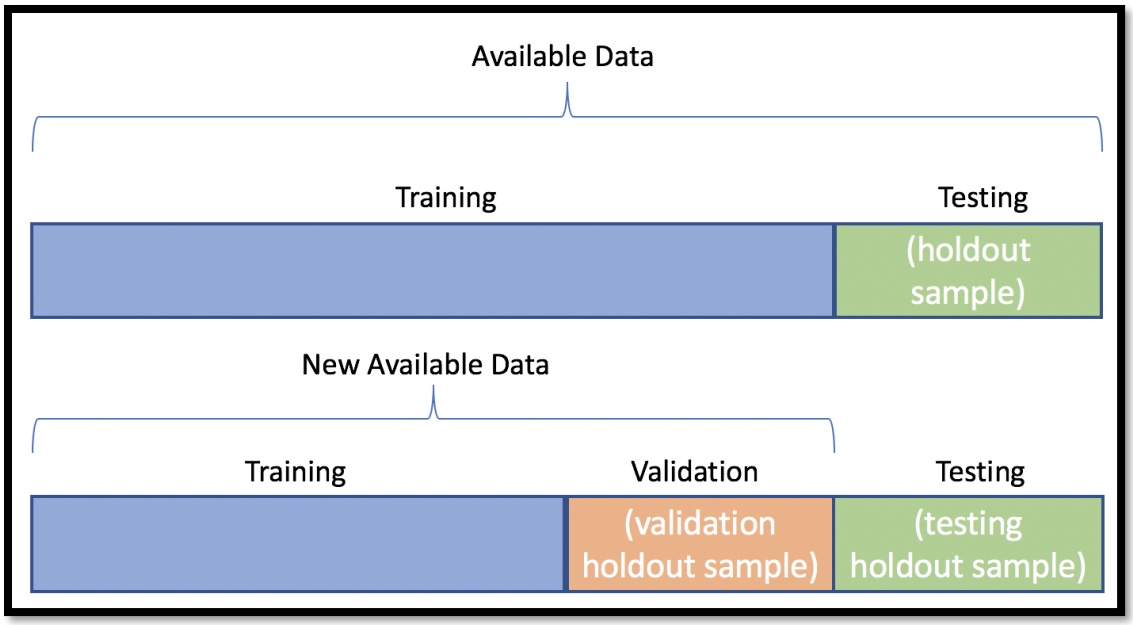
مقدار رو برای این ضریب تعیین کنیم؟ یک راه حل می تونه این باشه که از ۱۰۰ تا ضریب مختلف استفاده کنیم و بهترین شون رو برای مدل انتخاب کنیم. تا این جا همه چی خوب به نظر می رسه!

مدل رو بر مدار قرار می دیم! (روی دیتا های واقعی به اجرا می گذاریم.) و به طرز غیر قابل باوری مشاهده می کنیم که عجب! انگار خطایی که روی دیتا های واقعی اتفاق می افته خیلی بیشتر از خطایی هست که با بهترین مدل، بر روی test data ها گرفته بودیم. (نگران نباشین، از این غیر قابل باور ها تو دنیای داده ها خیلی زیاد پیش می یاد 😀)

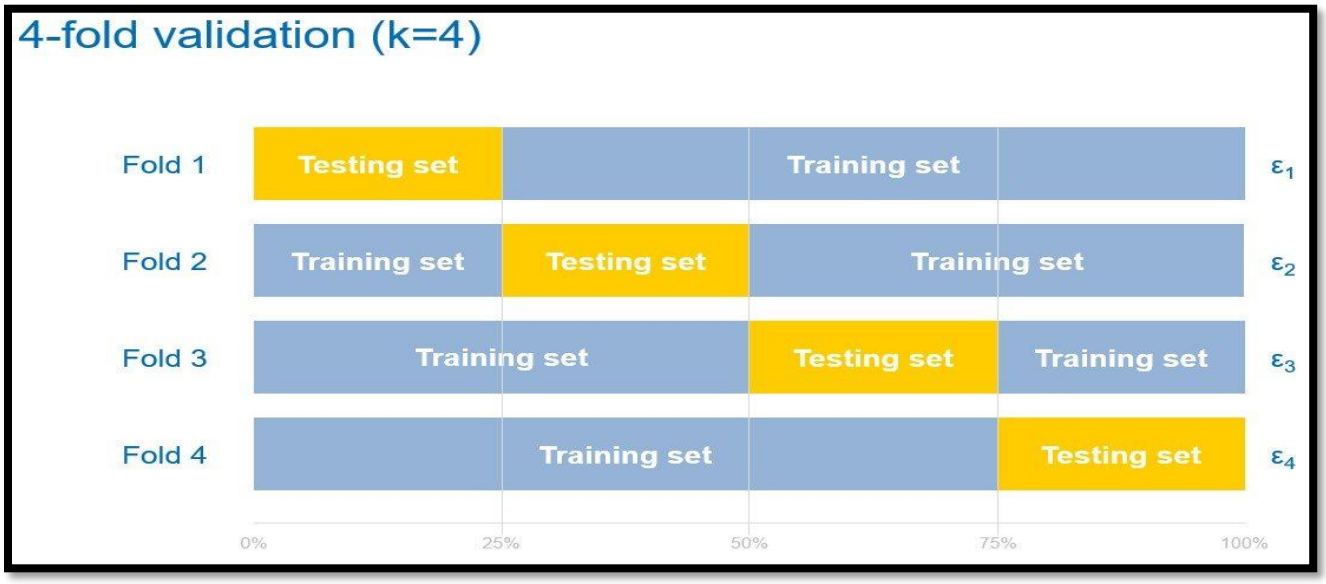
اما مشکل کار چی بوده!؟ دلیل اصلی این اتفاق، رسیدن به بهترین hyper – parameter ، صرفا بر روی تست کردن ارور های مختلف بر روی داده های محدود و به خصوصی یعنی، test data بوده. انگار ما مدل رو طوری تعیین کردیم که از کارکرد صحیحش فقط بر روی test data مطمئن باشین. و این ممکنه این رو در پی داشته باشه که مدل بر روی داده های جدید خوب کار نکنه.

برای حل این چالش، راه حل رایجی با عنوان **holdout-validation** به این صورت که:

یک قسمتی از training set را نگه می داریم تا در آینده مدل های مختلف را روی آن تست کنیم و بر اساس آن بهترین مدل را انتخاب نماییم. سپس از باقی مانده training set ها که ازشون با عناوین (**validation set, development set, dev set**) یاد می شه ، برای تنظیم hyper – parameter های مختلف و رسیدن به بهترین آن ها استفاده می کنیم. بعد که به پارامتر های بهینه رسیدیم، این بار ماشین را بر روی کل training set ها (با پارامتر های بهینه ای که به دست شون آوردیم)، آموزش می دیم. در آخر نیز مدل نهایی را بر روی test data ، اجرا می کنیم و ارور نهایی (generalization error) رو محاسبه می کنیم.




این راه حل معمولا خوب کار می کنه مگر در شرایطی خاص و در واقع پیشنهاد می شه در صورت امکان از روش توسعه یافته رویکرد بالا با نام **cross validation** استفاده کنیم. به این صورت که میایم و validation set های مختلفی رو در نظر می گیریم و با اجرای تک تک اون ها و در نهایت میانگین گیری از خطا های به دست آمده، به مدل قابل قبول تری می رسیم. البته در این روش نیاز به مدت زمان بیشتری برای رسیدن به جواب داریم.



## Data Mismatch

در برخی مواقع یادگیری را انجام می دهیم و همه چیز  خوب پیش می ره اما یه ایراد بزرگ وجود داره! این که داده هایی که به مدل آموزش دادیم نماینده ی خوبی از داده های که در آینده ماشین قراره باهاشون مواجه بشه نبوده. (یادتونه که قبلا به این چالش از دیتا ها اشاره کرده بودیم)

به عنوان مثل فرض کنیم که ما میخوایم اپلیکیشنی طراحی کنیم که از گیاهان عکس بگیره و به صورت خودکار گونه (نژاد) اون گیاه رو بهمون بگه. برای این کار یک میلیون داده رو (عکس گونه های مختلف گیاهان) رو از گوگل دریافت و به ماشین آموزش می دیم. اما مشکل اینجاست که عکس هایی که کاربرها با گوشی شون می گیرن معمولا با عکس های رایج موجود در اینترنت متفاوت هست. شاید از این ۱ میلیون عکس، تنها ۱۰۰ هزار تای اون ها شبیه عکس های کاربران باشه و این یعنی داده های ما نماینده خوبی از کل جامعه مون نبودن و این با خودش خوب کار نکردن ماشین رو به همراه می یاره، علی رغم این که در هنگام آموزش همه چی اوکی بوده! فهمیدن این که چنین مشکلی ممکنه ناشی از  **data mismatch** باشه و نه overfitting،  موضوع بسیار مهمی هست و بعد از شناسایی این مسئله باید سعی کنیم روی داده هامون تجدید نظر کنیم.

در این فصل به بررسی کلی ماشین لرنینگ پرداختیم و این که از صفر تا ۱۰۰ چه مراحلی طی می شه تا ماشینمون یاد بگیره و بتونیم یک ترمیناتور بسازیم! در فصل های بعدی راجع به تک تک مباحثی که در این فصل بهشون اشاره کردیم، به صورت عمیق و با جزییات بیشتر صحبت خواهیم کرد. **اما سوال هایی تا اینجا که باید براشون جواب داشته باشین!** (در غیر این صورت یه مرور کوتاهی روی مطالب قبلی داشته باشین)

- ماشین لرنینگ چیه؟
- چند تا از دلایلی که ماشین لرنینگ رو در جایگاه خیلی بالاتری از رویکرد های سنتی قرار میده، نام ببرین.
- دیتاهایی که در اون ها لیبل داریم از چه نوع یادگیری ماشینی هستن؟
- چند نمونه از معروف ترین مثال های supervised & unsupervised learning رو نام ببرین.
- اگر بخواین یه رباتی طراحی کنین که بتونه به خوبی روی زمین راه بره، از چه الگوریتمی استفاده می کنین؟
- برای دسته بندی مشتریان یک فروشگاه چه الگوریتمی رو پیشنهاد می کنین؟
- برنامه تشخیص ایمیل اسپم رو supervised or unsupervised می دونین؟
- منظور از یادگیری لحظه ای (online) چیست؟
- تعریف دقیقی از out-of-core learning ارایه بدین.
- چه نوع از سیستم های یادگیری ماشین بر اساس پی بردن به شباهت بین فیچر های مختلف داده ها کار می کند؟
- تعریف و کاربرد hyper-parameter چیست ؟
- شیوه کارکرد الگوریتم های model-based learning algorithms رو توضیح دهید.
- چهار چالش اصلی در یادگیری ماشین را نام ببرید.
- اگر مدل روی training set خوب کار کند اما بر روی داده های جدید عملکرد مناسبی نداشته باشد چه اتفاقی افتاده است؟ سه سناریو محتمل را شرح دهید.
- دلیل استفاده از test set چیست؟
- هدف از ایجاد validation set چیست؟
- اگر برای تنظیم هایپرپارامتر ها از test set استفاده کنیم چه اشتباهی ممکن است رخ بدهد؟

**امیدوارم تا این جای کار ، براتون مفید بوده باشه و باعث خوشحالیمونه،  اگر نظرتون رو راجع به این مجموعه و این که آیا اصلا ادامه بدیم یا ... داشته باشیم. می تونیم از طریق ای دی تلگرامی زیر با ما در ارتباط باشین:**

**https://telegram.me/mraliarabshahi**

**موفق و موید باشین 😈**