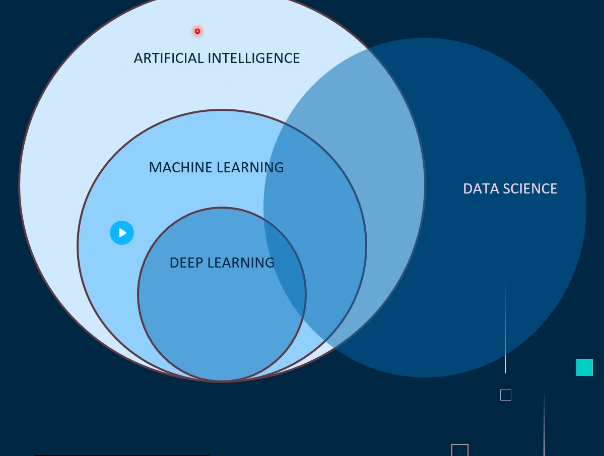
بنام خدا

علم داده (Data Science) یک حوزه میان رشته ای(ترکیبی از چندین رشته مختلف مانند آمار، ریاضیات، علوم کامپیوتر و ... ) است که از متدها، فرآیندها(شامل مراحل مختلفی مانند جمع آوری دادهها، پاکسازی دادهها، تحلیل دادهها و تفسیر نتایج)، الگوریتمها و سیستمهای علمی برای استخراج دانش و بینش از داده های ساختاریافته و غیرساختاریافته استفاده میکند.

داده های ساختاریافته به دادههایی گفته میشود که در قالبهای مشخص و سازمان یافته مانند جداول پایگاه داده ذخیره میشوند. داده های غیرساختاریافته شامل داده هایی مانند متن، تصاویر و ویدئوها هستند که قالب مشخصی ندارند.



Data Science بیشتر بر روی جمع آوری، پردازش و تحلیل دادهها تمرکز دارد، در حالی که هوش مصنوعی و یادگیری ماشین بیشتر بر روی توسعه الگوریتمها و مدلهای هوشمند تمرکز دارند. Data Science به دلیل گستردگی و تنوع تکنیکها و ابزارهایی که استفاده میکند، به عنوان یک حوزه مستقل شناخته میشود. این حوزه شامل استفاده از تکنیکهای مختلف از جمله آمار، تحلیل داده ها، یادگیری ماشین و حتی هوش مصنوعی است، اما محدود به این تکنیکها نیست. به همین دلیل، علم داده به عنوان یک حوزه مستقل شناخته میشود که از تکنیکهای مختلف برای استخراج دانش و بینش از داده ها استفاده میکند. فرض کنید شما یک فروشگاه آنلاین دارید و داده های مربوط به فروش، رفتار مشتریان و بازدیدهای وبسایت را جمع آوری کردید. با استفاده از تکنیکهای استخراج دانش، میتوانید الگوهای خرید مشتریان را شناسایی کنید. سپس با تحلیل این الگوها، میتوانید بینشهایی مانند محصولات پرفروش، زمانهای اوج خرید و نیازهای مشتریان را استخراج کنید. این بینشها به شما کمک میکنند تا استراتژیهای بازاریابی و فروش خود را بهبود بخشید.

AI (Artificial Intelligence): هوش مصنوعی مثل یک ربات هوشمند هست که میتواند فکر کند و انجام وظایفی را یاد بگیرد که معمولا فقط انسان ها میتوانند انجام دهند مانند درک کلمات یا تشخیص تصاویر

ML (Machine Learning) : وقتی است که یک ربات هوشمند یاد میگیرد با جستجوی اطلاعات زیاد، در کار خود خوب عمل کند. مثل اینکه با توجه به نحوه بازی تون در گذشته، چه اسباب بازی میخواید. در واقع ML ، روش یادگیری ربات های AI است. بنابراین فرایند یادگیری، در این بخش است.

DL (Deep Learning): زمانی است که ربات با نگاه کردن مکرر به چیزی ، به روش خاص (مثل الگوریتم های یادگیری ماشین) یاد میگیرد. فرض کنید شما میخواهید یک ربات را آموزش دهید تا گربه ها را در تصاویر تشخیص دهد. برای این کار، شما به ربات تعداد زیادی تصویر از گربه ها نشان میدهید. ربات با مشاهده مکرر این تصاویر، یاد میگیرد که ویژگی های خاصی مانند شکل گوش ها، چشمها و بدن گربه را شناسایی کند.

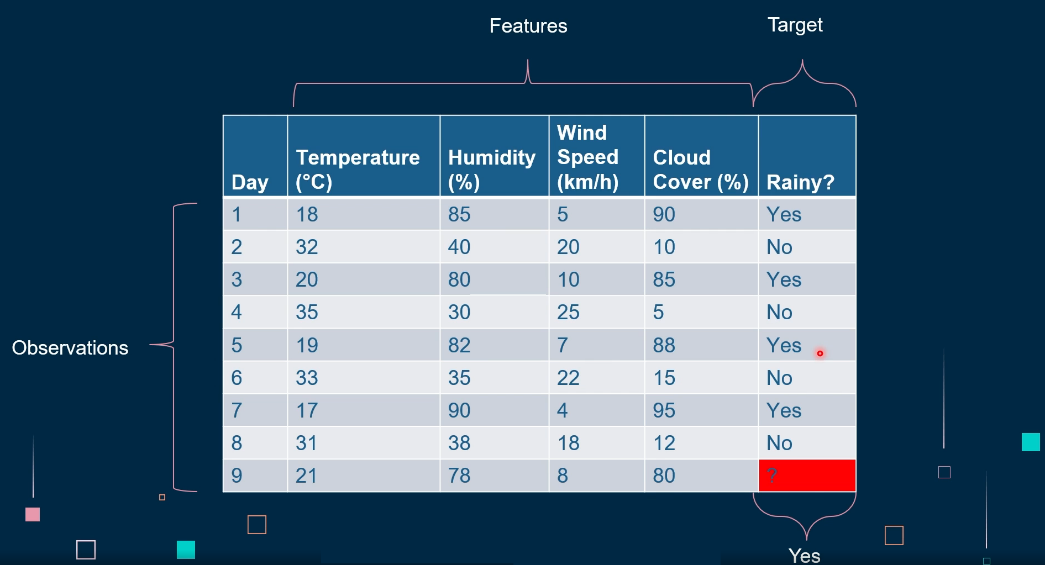
Data Science : مثل کارآگاهی است که برای حل یک معما، به سرنخ (داده) های زیادی نگاه میکند. Data Science چیزی بیش از نوشتن الگوریتم ها است. بیشتر شبیه حل یک موقعیت با استخراج دانش و بینش است.

Data Scientist ها از الگوریتم های machine learning استفاده میکنند. Machine learning الگوهای موجود در داده ها را تحلیل میکند و از این الگوها برای پیش بینی یا گروه بندی موارد مشابه استفاده میکند. بطور کلی ما تجزیه و تحلیل برخی الگوها در داده های خود را پیش بینی میکنیم. استفاده از الگوریتم machine learning و ساخت machine learning بخش مهمی از دانشمند داده بودن است.

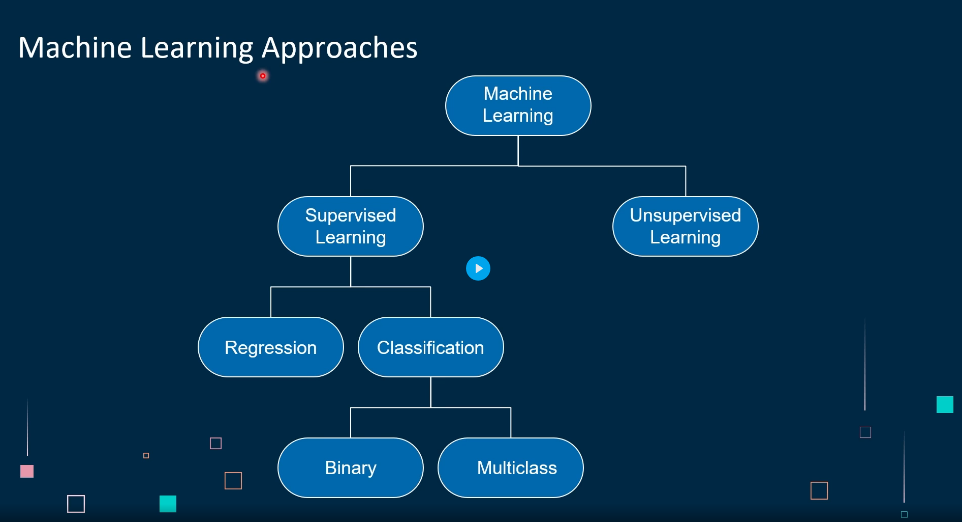
Modeling Algorithm :

‘learning’ چیست؟ یک الگوریتم یادگیری ماشین را مانند یک دانش آموز در حال یادگیری شطرنج در نظر بگیرید. این دانش آموز (الگوریتم) درس میخواند و چندین بازی (داده های آموزشی) انجام میدهد. بنابراین این قسمت تجزیه و تحلیل الگوها است. ما یک داده را میبینیم و مثل دانش آموزی هستیم که چندین بازی انجام میدهد. استراتژی ها و تکنیک ها(pattern - الگو) رو یاد میگیریم. و با گذشت زمان ، دانش آموزان در تصمیم گیری برای حرکت بعدی بهتر میشوند (predict). بعد از آن دانش آموزان رویکرد خود را بر اساس برد یا باخت و با هدف کسب مهارت تنظیم میکنند. هدف این است که نه تنها در سناریوهای تکراری بلکه در بازی های جدید نیز مهارت داشته باشید(generalization)

دقت کنید اگر دانش آموزی نتونه این موارد رو یاد بگیره ، اونا رو حفظ میکنه که این همون generalization هست.



میخوایم اولین مدل یادگیری ماشین رو بدون کد نویسی ، انجام بدیم. فرض کنیم یه سری داده برای نه روز داریم و میخوایم ببینم روز نهم بارونی هست یا نه. همونطور که از جول مشخصه دما در روزهای بارونیحدود 17 تا 20 درجه و درصد رطوبت حدود 80 تا 90 درصد هست. رطوبت کم و پوشش ابر زیاد هست. پس احتمال میدید که روز نهم بارانی باشد. در واقع مثل یک مدل machine leaning عمل میکنید. تا اینجا چون داده های ما کم هست ، میتونیم مثل یک machine learning رفتار کنیم ولی گاهی دیتاها به صدها یا هزاران ردیف میرسه. پس الگوریتم ها با بررسی مشاهدات و فیچرها، هدف تعریف نشده را پیش بینی میکنند.



Supervised learning : یک رویکردی است که با label گذاری روی دیتا خروجی های جدید رو پیش بینی میکنه.

در یادگیری نظارت شده (Supervised Learning)، label یا برچسب به معنای اطلاعاتی است که به دادهها اضافه میشود تا مدل یادگیری بتواند از آنها برای پیشبینی استفاده کند. به عبارت دیگر، برچسب ها نتایج یا خروجیهای مورد انتظار هستند که به داده های ورودی متصل میشوند. فرض کنید میخواهیم مدلی بسازیم که بتواند ایمیل های اسپم را تشخیص دهد. برای این کار، نیاز داریم به مدل یاد بدهیم که چه ایمیل هایی اسپم هستند و چه ایمیل هایی نیستند. داده های ورودی : متن ایمیل ها و اسپم یا غیر اسپم میشه label. در اینجا، هر ایمیل یک label دارد که نشان میدهد آیا آن ایمیل اسپم است یا خیر. مدل با استفاده از این label ها یاد میگیرد که ویژگیهای ایمیل های اسپم و غیر اسپم را تشخیص دهد و در آینده بتواند ایمیل های جدید را به درستی دسته بندی کند.

Supervised learning شامل دو دسته است

1. Classification : نوعی یادگیری تحت نظارت است که در آن، مدل label های گسسته را پیش بینی میکند. فرض کنید میخواهیم مدلی بسازیم که نوع میوه ها را تشخیص دهد. دادههای ورودی ما میتواند شامل ویژگیهایی مانند رنگ، اندازه و وزن میوه ها باشد. برچسبهای گسسته در اینجا میتواند شامل انواع میوه ها مانند سیب، پرتقال و موز باشد.
   * باینری : یه نوعی از classification از Supervised learning هست که بین دو label پیش بینی میکنه مثلا اینکه ایمیل spam هست یا نیست
   * Multi-class : در این روش از چندین دسته ی ممکن، یک result داریم. مثلا ما چندین تصویر رو به machine learning میدیم و سعی میکنه پیش بینی کنه که آن میوه سیب هست یا موز یا ...
2. Regression : نوعی یادگیری تحت نظارت است که در آن مدل، نتایج مستمر را پیش بینی میکند. نه label های گسسته بلکه نتایج پیوسته. (برچسبهای پیوسته به معنای مقادیر عددی هستند که میتوانند هر مقداری در یک بازه مشخص باشند. مانند تخمین قیمت خانه بر اساس متراژ، موقعیت مکانی و سایر ویژگی های آن)

برمیگردیم به مثال باران. در اون مثال، چون نتیجه - بارانی است یا نه – پس یک supervised learning و از نوع classification و باینری هست. اگر بخوایم مقدار باران را بررسی کنیم، مشکل رگرسیون خواهیم داشت.

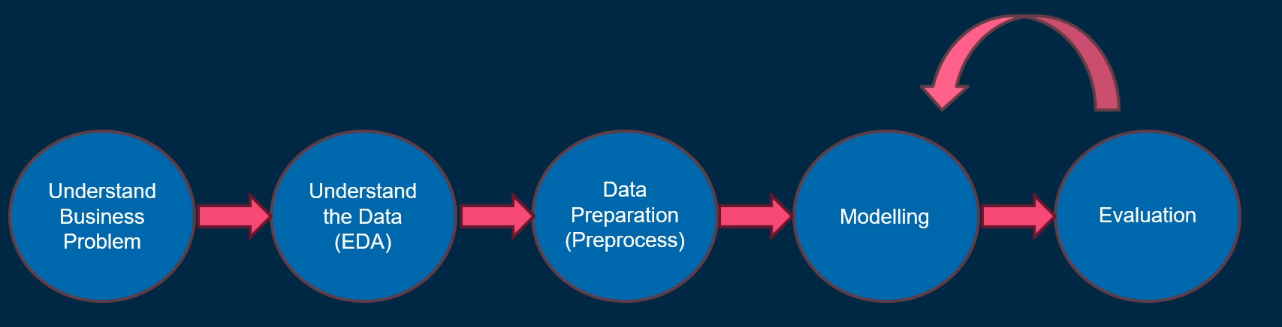
Unsupervised Learning : یادگیری بدون نظارت یک روش یادگیری ماشین است که در آن مدلها بر روی داده های بدون برچسب آموزش داده میشوند تا الگوها یا ساختارهایی را درون آنها پیدا کنند. به عنوان مثال، میتوان مشتریان را بر اساس رفتارهای خریدشان به گروه های مختلف دسته بندی کرد. در واقع گاهی یادگیری بدون نظارت را Clustering مینامیم. در همون مثال آب و هوا میتونیم بگیم روزها رو به دو دسته تقسیم کنیم. بعضی روزها رطوبت کمتر و سرعت باد بیشتر دارند و بعضی روزها برعکس

Self-supervised learning : آموزش خود نظارتی یک نوع یادگیری است که در آن مدل label های خود را از داده های ورودی دریافت میکنند. مانند ChatGPT

Semi-supervised learning : آموزش نیمه نظارتی – مقدار کمی از داده های label گذاری شده را با مقدار زیادی از داده های بدون label ترکیب میکند و چون label گذاری خیلی هزینه بر است یه سری از دیتاها را label گذاری میکنیم و بقیه رو بدون label

Reinforcement Learning : یک روش یادگیری ماشین است که در آن یک عامل (agent) با تعامل با محیط خود و دریافت پاداش ها یا تنبیه ها، یاد میگیرد که چگونه رفتار کند تا بیشترین پاداش ممکن را کسب کند. فرض کنید میخواهیم یک ربات را آموزش دهیم تا در یک اتاق حرکت کند و به یک نقطه خاص برسد. ربات با انجام هر حرکت (عمل) یک پاداش یا تنبیه دریافت میکند. هدف ربات این است که یاد بگیرد چگونه حرکت کند تا بیشترین پاداش را کسب کند و به نقطه مورد نظر برسد.

Data Science Project Process :



مرحله اول : درک مشکل بیزینس این است که هدف خود را روشن کنید. درک مسئله کسبوکار اولین و مهمترین مرحله در هر پروژه تحلیل داده یا یادگیری ماشین است. این مرحله تعیین میکند که تمام اقدامات آینده شما چگونه باید انجام شوند. این مهمترین بخش فرایند یادگیری ماشین است.

مرجله دوم : درک داده ها - در مورد انجام برخی از تجزیه و تحلیل ها است که گاهی اوقات EDA نامیده میشوند. (تجزیه و تحلیل داده های اکتشافی). هدف در اینجا این است که اطلاعات خود را بشناسید و باید از همان قدم اول آن را با درک کسب و کار خود ترکیب کنید.

مرحله سوم : آماده سازی داده ها – آماده سازی مدل داده ها پس از درک داده ها انجام میشود. در 99 درصد مواقع، شما یک مجموعه داده کامل ندارید. باید داده ها را دستکاری کنید تا بتوانید آن را مدل کنید.

این سه مرحله 90 درصد کل فرایند پروژه است. بیشتر اوقات چیزی که یک دانشمند داده جونیور را از یک دانشمند داده سنیور جدا میکند، کیفیت این سه مرحله است.

مرحله چهارم : الگوریتم یادگیری ماشین در این مرحله وارد میشود. در این مرحله، مدلهای مختلفی برای پیشبینی یا طبقه بندی داده ها ساخته و ارزیابی میشوند. یکی از زیرمراحل مهم در مدلسازی، تقسیم دادهها به مجموعه های آموزش (Train) و آزمون (Test) است.

آخرین مرحله ارزیابی است. شما نمیتوانید چیزی را که نمیتوانید اندازه گیری کنید، بهبود بخشید. باید عملکرد مدل ها را بسنجیم تا بهترین مدل را بدست آوریم. باید بخش مدلسازی خود را بصورت پویا تغییر دهیم و دوباره ارزیابی کنیم و این فرایند تکرار میشود. در مرحله آخر تا زمانیکه نتایج راضی کننده باشد ادامه خواهد داشت.

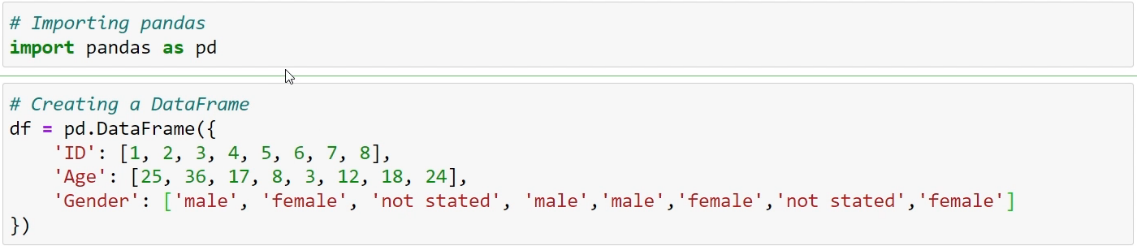
وارد سایت <https://www.anaconda.com/> شده و از نوت بوک Jupyter استفاده میکنیم.

انواع Data Type :

1. Numerical Data : برای اندازه گیری های کمی مثل ارتفاع و وزن
   * Discrete (اعداد گسسته) : موارد قابل شمارش (تعداد دانش آموزان)
   * Continuous (اعداد پیوسته) : موارد قابل اندازه گیری مثل دما (شما نمیتونید دما رو بشمارید ولی میتونید اندازه بگیرید)
2. Categorical Data : برای خاصیت های کیفی مثل ملیت و جنسیت
   * Nominal (اسمی) : هیچ ترتیبی نداره مثل غذاهای فرانسوی و غذاهای ایرانی و ... (نمیتونیم بین غذاها نظم داشته باشیم)
   * Ordinal (ترتیبی) : دسته بندی ترتیبی مثل خرید یک تیشرت که سایز کوچیک، متوسط و بزرگ داره

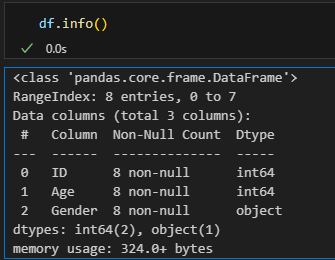
Pandas : یک ابزار قدرتمند تجزیه و تحلیل و دستکاری داده های open source است که بر روی پایتون ساخته شده است. این ساختار داده و عملیات را برای دستکاری جداول عددی و سری های زمانی ارائه میدهد. پاندا ابزاری است که به ما اجازه میده تا یک داده جدولیدر پایتون داشته باشیم مانند جدول اکسل

Pd.DataFrame به ما یه جدول میده واسه وقتی که دیتا رو از یه فایل اکسل یا csv میگیریم.



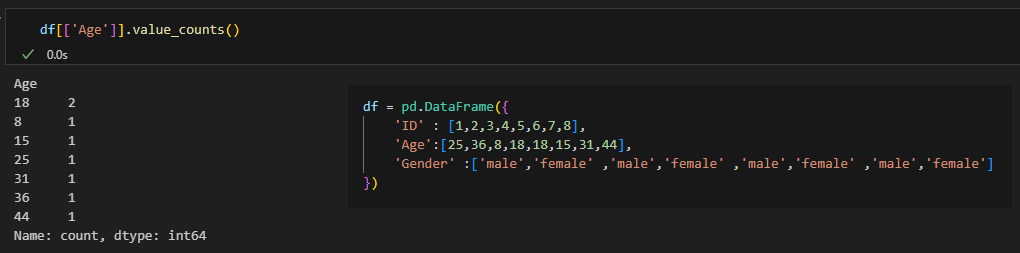
df.head() پنج سط اول را نمایش میدهد ولی اگر تعداد سطر مشخص بخوای باید بهش ورودی بدیم

df.tail() برعکس head را نمایش میدهد.



تابع describe در pandas یک خلاصه آماری از دادههای عددی در DataFrame شما ارائه میدهد. این تابع اطلاعات زیر را برای هر ستون عددی محاسبه و نمایش میدهد:

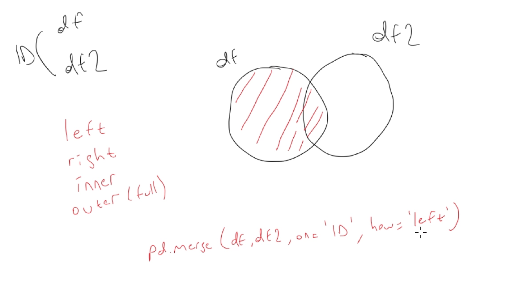
 

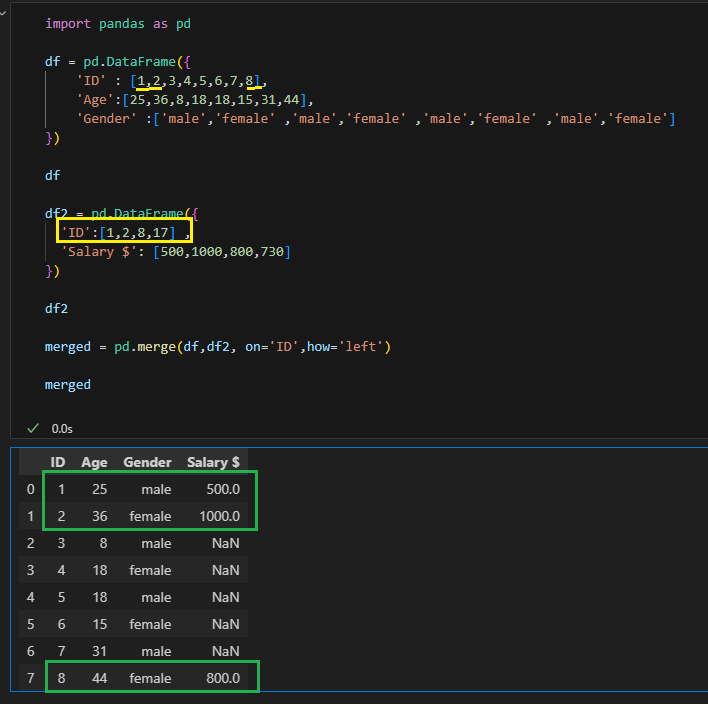


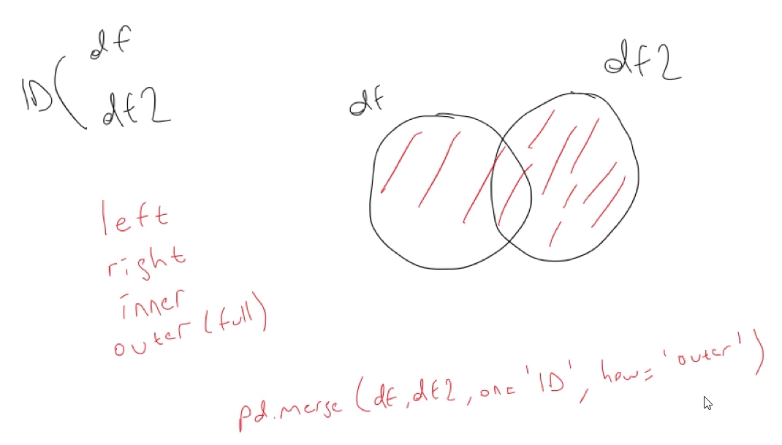
df.loc[1,'Age'] : از ستون Age ایندکس اول رو نمایش میده.

df.iloc[row index , column index]

pd.merge(df , df2 ,on =’common column’,how = ‘join type’)

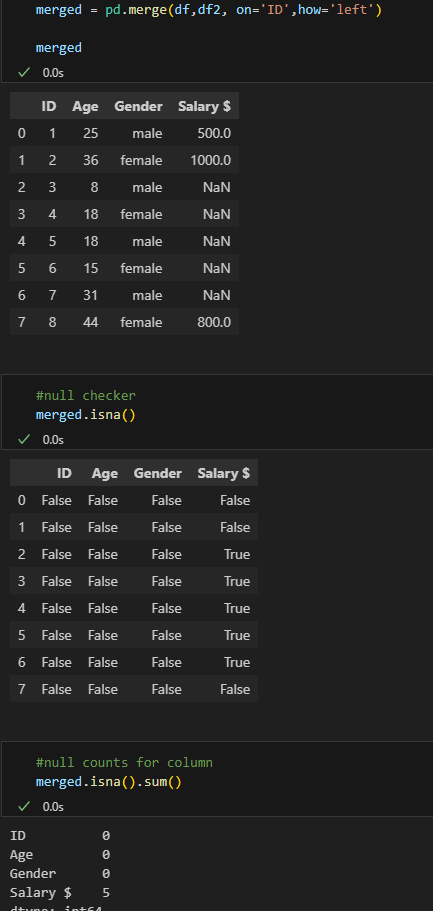






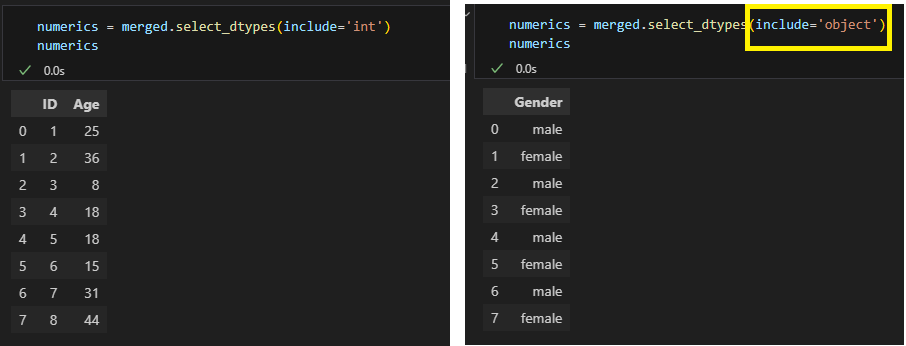
How = inner ; outer ; left ; right

تابع isna در pandas برای شناسایی مقادیر تهی یا NaN در DataFrame استفاده میشود. وقتی این تابع را روی یک DataFrame یا Series اعمال میکنید، یک DataFrame یا Series جدید برمیگرداند که در آن هر مقدار تهی با True و هر مقدار غیر تهی با False نشان داده میشود. و تابع isna().sum() مجموع مقایر null برای هر ستون را نمایش میدهد.

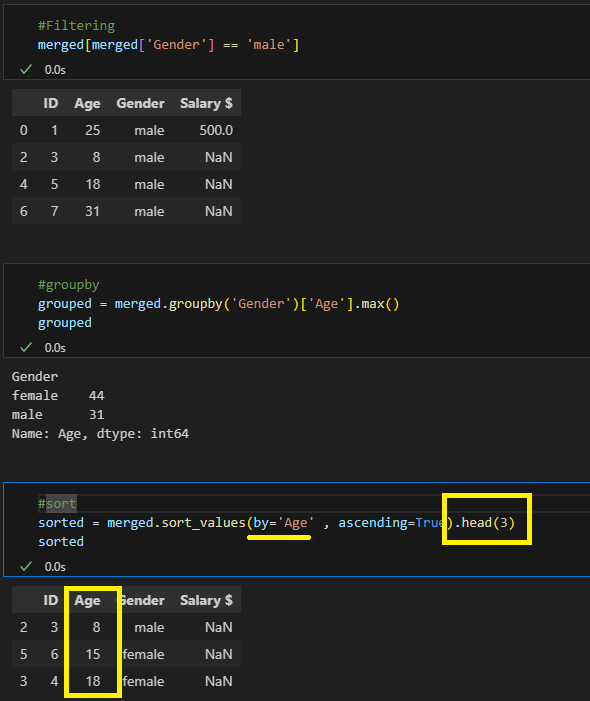


متد type نوع رو نشون میده مثلا اگر بنویسیم type(merged.loc[0,’Gender’]) خروجی str است





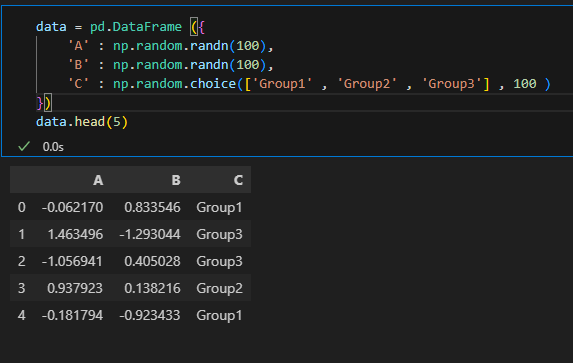
#Filtering && #groupby && #Sort



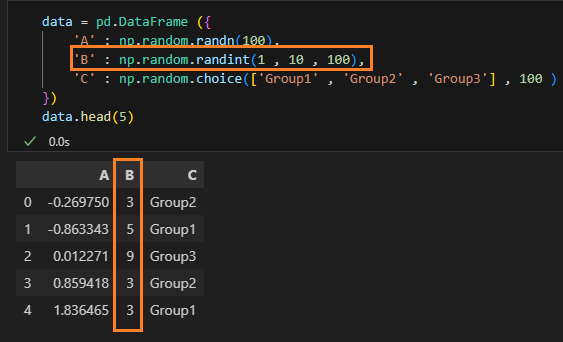
Numpy : یک library است که برای کار با سری ها ایجاد شده. پس Panda برای Data Frame و Numpy برای سری ها. Numpy برای کار با آرایه ها و محاسبات عددی طراحی شده است.

|  |  |  |
| --- | --- | --- |
| آرایه ای از ایندکس ها رو برمیگردونه که نشون میده چطوری میشه آرایه اصلی رو بصورت صعودی مرتب کرد |  | Np.argsort |
| مرتب سازی رو انجام میده | Arr[sorted\_array) |
| برای شرط گذاری  آرایه اصلی :  [1,3,2,-4,-7]  میخوایم شرط بذاریم آیتم های بزرگتر از یک رو نشون بده بقیه رو صفر نمایش بده |  | np.where(arr > 1,arr,0) |
| مشابه بالا : شرط بذاریم اعداد منفی رو چاپ کنه و بجای اعداد مثبت \* بذاره |  | result = np.where(arr<0 , arr , '\*')  print(result) |
| قدر مطلق آرایه |  | Np.abs(arr) |

* عدد 100 در کد ‘c’ تعداد عناصری است که باید به صورت تصادفی از بین گزینه های موجود انتخاب شوند (چون ما 5 تای اول رو خواستیم بقیه رو نشون نداد)



* اگر بخوام اعداد صحیح رو نشون بده از np.random.randint استفاده میشه. دقت کنیم که data frame رو از pd ساختیم و انتخاب یک عدد از اعداد یک تا 100 چون آرایه است از جنس np ساختیم.



Matplotlib && seaborn :

* برای رسم یک هیستوگرام از داده های ستون A در DataFrame data :

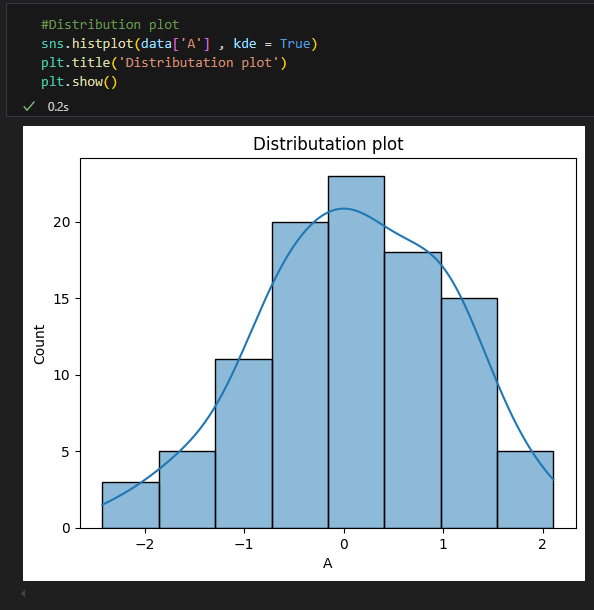
دستور نصب :

* pip install matplotlib
* pip install seaborn

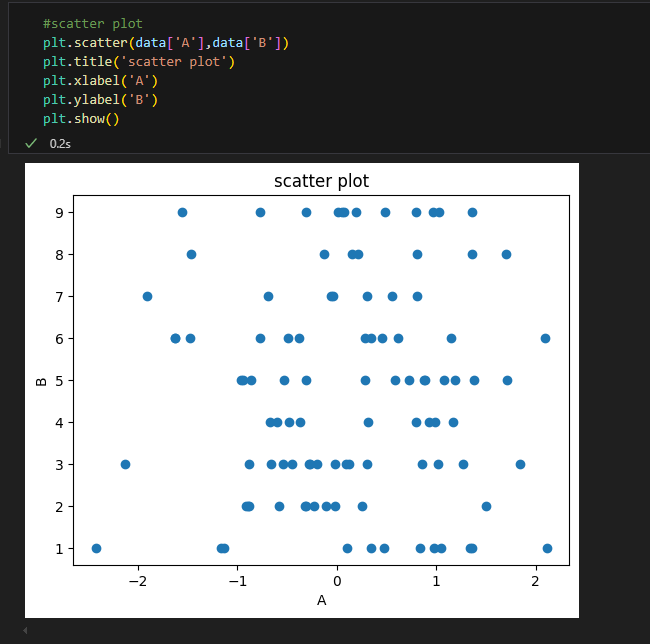
matplotlib.pyplot و seaborn دو کتابخانه محبوب برای رسم نمودارها و تجسم داده ها در پایتون هستند.

kde=True: این آرگومان باعث میشود که یک منحنی چگالی کرنل (Kernel Density Estimate) نیز روی هیستوگرام رسم شود. این منحنی به شما کمک میکند تا توزیع دادهها را بهتر ببینید

نمودار توزیع بر اساس مقادیر یک ستون :

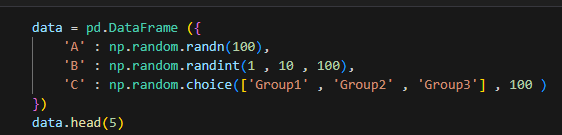


نمودار پراکندگی :



فرق Distribution plot و scatter plot در این است که اولی روی یک ستون است و دومی در واقع یک ستون در مقابل ستون دیگرخواهیم داشت (برای درک رابطه بین دو ستون)

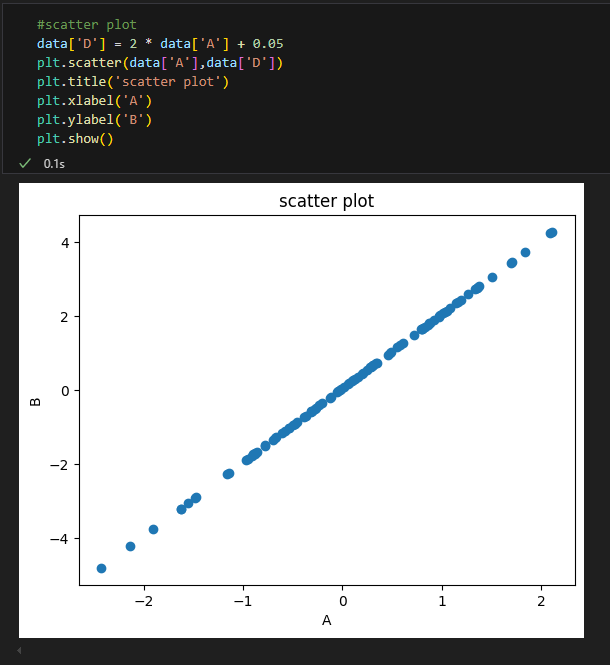
طبق نمودار بالا ما هیچ رابطه ای بین A و B نمی بینیم. پس چیکار کنیم ارتباط بین اینا رو ببینیم. بنابراین یه ستون D اشاقه میکنیم که مقدارش دو برابر دیتای A + پنج درصد باشه.



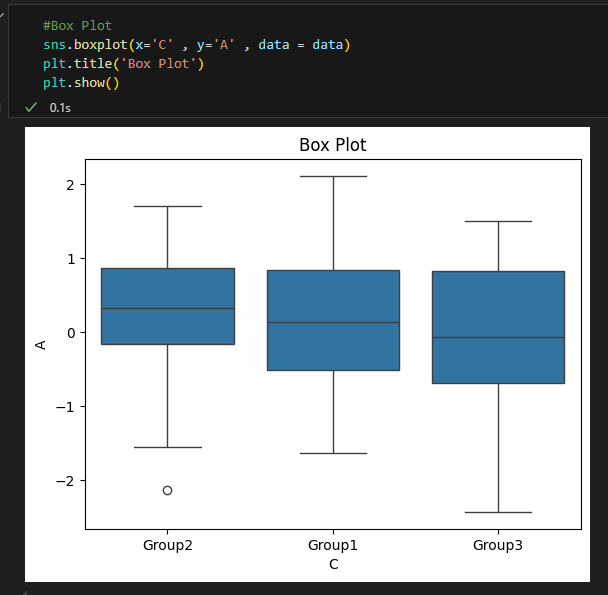
+



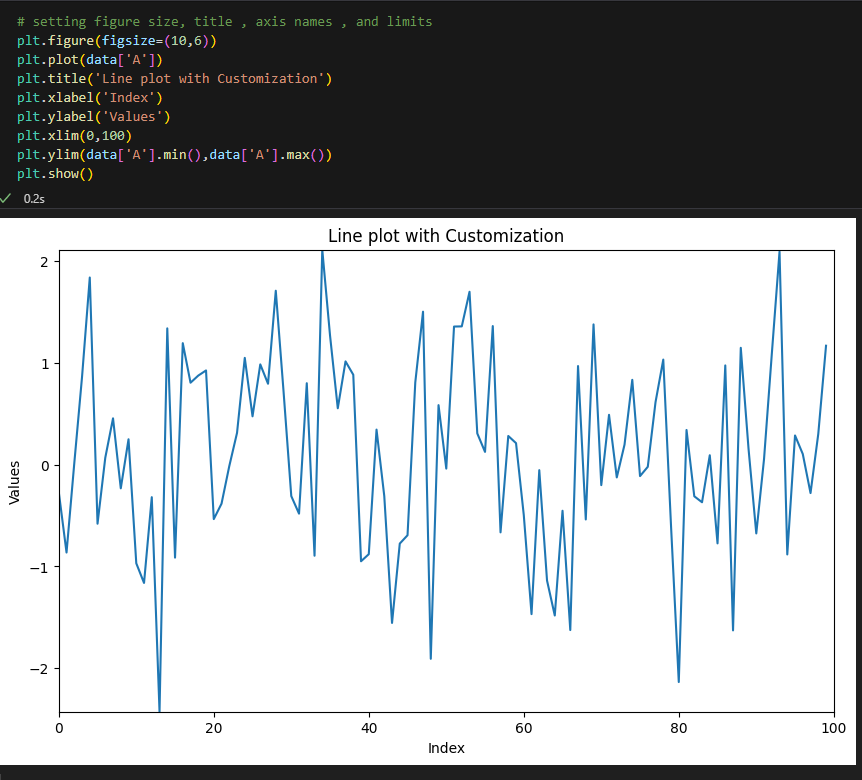
دقت : الان باید برای اینکه ارتباط نمایش داده بشه ، دیتای A,D رو مقایسه کنیم



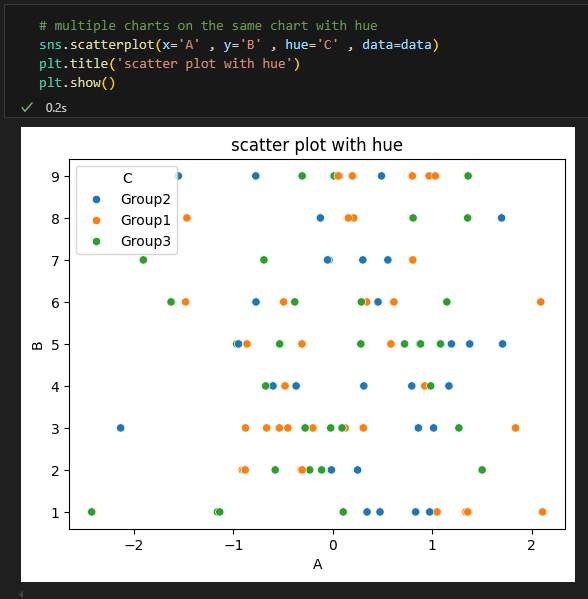
Box plot : بیشتر برای شناسایی نقاط پرت استفاده میشود. خطوسط میانه است و نقاط اطراف ، نقاط پرت محسوب میشوند. در واقع وقتی نقاط پرت رو میبینیم نمودار box رو بررسی میکنیم.



سفارشی سازی داده ها



مشابه نمودار پراکندگی با این تفاوت که رنگی نمایش داده میشود :



نمودار توزیع بر اساس گروه ها :

