

# Special Topics Exercises

SPECIAL TOPICS  
ALIREZA HASSANZADEH  
ALI AKBAR MAKARIAN

## فهرست مطالب

Machine Learning	.1
Python Programming	.2
Data Structures and Algorithms	.3
Data Preprocessing	.4
<b>Visualization</b>	.5
Advanced Topics	.6

---

## A. تفاوت‌های یادگیری ناظارت شده (Supervised Learning) و یادگیری بدون ناظارت (Unsupervised Learning)

یادگیری ناظارت شده (Supervised Learning) و یادگیری بدون ناظارت (Unsupervised Learning) دو روش اصلی در یادگیری ماشین هستند. در یادگیری ناظارت شده، مدل با استفاده از داده‌های برچسب‌گذاری شده آموزش می‌بیند، به این معنی که هر نمونه دارای یک خروجی مشخص است. این روش برای مسائل طبقه‌بندی (Classification) و پیش‌بینی (Regression) به کار می‌رود. در مقابل، یادگیری بدون ناظارت از داده‌های بدون برچسب استفاده می‌کند و هدف آن یافتن الگوهای خوش‌بندی داده‌ها است. این روش برای کاهش ابعاد داده‌ها و کشف ساختارهای مخفی در داده‌ها مفید است. به طور کلی، یادگیری ناظارت شده دقیق‌تر و با امتیازی دارد، اما یادگیری بدون ناظارت برای تحلیل داده‌های ناشناخته و بدون ساختار مناسب‌تر است.

## B. چرا Feature Scaling در الگوریتم‌های Machine Learning ضروری است؟

به این دلیل مهم است که بسیاری از الگوریتم‌های یادگیری ماشین مانند رگرسیون خطی، SVM و شبکه‌های عصبی حساس به مقیاس داده‌ها هستند. اگر ویژگی‌های داده‌ها در مقیاس‌های مختلف باشند، مدل ممکن است نتواند به درستی همگرا شود یا وزن‌های بهینه را پیدا کند. روش‌های متداول Feature Scaling شامل Normalization و Standardization هستند.

## C. تأثیر Feature Scaling بر الگوریتم‌های مختلف

- الگوریتم‌های مبتنی بر گرادیان (مانند شبکه‌های عصبی و رگرسیون لجستیک) : سرعت همگرایی بهینه افزایش می‌یابد.
- الگوریتم‌های مبتنی بر فاصله (مانند KNN و SVM) : دقت مدل بهبود پیدا می‌کند.
- الگوریتم‌های درختی (مانند تصمیم‌گیری درختی) : معمولاً نیازی به Feature Scaling ندارند.

## D. چه تفاوتی دارند؟ Normalization و Standardization

- استانداردسازی (Standardization) : داده‌ها را به نحوی تبدیل می‌کند که میانگین آن برابر با صفر و واریانس آن برابر با یک باشد. این روش برای الگوریتم‌هایی مانند KNN و SVM کاربرد دارد.

[ $1,1$ ] نرمال‌سازیNormalization ( • مقياس‌بندی می‌کند. اين روش در شبکه‌های عصبی و الگوريتم‌های مبتنی بر فاصله مؤثر است.

### مقاييسه کاربردهای Normalization و Standardization

زمانی که استفاده می‌شود	روش
زمانی که توزيع داده‌ها نرمال نیست	<b>Standardization</b>
زمانی که مقياس داده‌ها بسيار مختلف است	<b>Normalization</b>

### D. چرا Min-Max Normalization برای مقياس‌بندی داده‌ها استفاده می‌شود؟

داده‌ها را در بازه‌ای مشخص (معمولًاً  $[0,1]$ ) مقياس‌بندی می‌کند و برای زمانی مناسب است که می‌خواهيم توزيع داده‌ها را حفظ کنيم. اين روش معمولاً در شبکه‌های عصبی و الگوريتم‌های مبتنی بر فاصله استفاده می‌شود.

#### معايip و مزاياip

- **مزايا :** حفظ توزيع اصلي داده‌ها، مناسب برای مدل‌هایی که به مقدار مطلق ويژگی‌ها حساس هستند.
- **معايip :** حساس به مقادير پرت. (Outliers)

### E. چیست و چرا کاربرد دارد؟ Z-Score Normalization

Z-Score Normalization داده‌ها را به صورت استاندارد درمی‌آورد، به طوري که ميانگين صفر و انحراف معيار برابر يك باشد. اين روش برای الگوريتم‌هایی که از داده‌های دارای توزيع نرمال بهره می‌برند، بسيار مناسب است.

### F. چیست؟ regularization در الگوريتم‌های Machine Learning

Regularization تکنيکی است که برای جلوگيري از بيش‌برازش (Overfitting) استفاده می‌شود. دو نوع متداول آن عبارتند از:

- **L1 Regularization (Lasso Regression):** برحی از ويژگی‌های مدل را به صفر تبدیل می‌کند.
- **L2 Regularization (Ridge Regression):** وزن‌های مدل را کاهش داده ولی به صفر نمی‌رساند.

### تأثير regularization بر مدل‌ها

- کاهش پيچيدگی مدل
- جلوگيري از وابستگی بيش‌ازحد به داده‌های آموزش

## چه مشکلاتی را در **Model-building** و **Underfitting** و **Overfitting** می‌آورند؟

**Overfitting:** مدل بیش از حد به داده‌های آموزش وابسته می‌شود و در تعمیم به داده‌های جدید عملکرد ضعیفی دارد.

**Underfitting:** مدل نمی‌تواند الگوهای داده‌ها را یاد بگیرد و عملکرد ضعیفی روی داده‌های آموزش و تست دارد.

## راه‌های جلوگیری از Underfitting و Overfitting

- استفاده از داده‌های بیشتر

- به کارگیری Regularization

- انتخاب مدل مناسب با پیچیدگی داده‌ها

## چرا در **Train/Test Split** کاربرد دارد؟

K-Fold Cross-Validation ویژه Cross-Validation را چندین بار روی مجموعه‌های متفاوتی از داده‌های آموزش و تست ارزیابی می‌کند. این روش باعث افزایش دقت مدل و کاهش وابستگی به داده‌های خاص می‌شود.

## مزایای Cross-Validation

- بهبود قابلیت تعمیم مدل

- کاهش تأثیر داده‌های پرت

- کاهش احتمال Overfitting

## چگونه کار می‌کند؟

Gradient Descent یک الگوریتم بهینه‌سازی است که برای به روزرسانی وزن‌های مدل استفاده می‌شود. این روش از طریق محاسبه مشتق تابع هزینه، در جهت کاهش مقدار هزینه حرکت می‌کند. انواع متدائل آن عبارتند از:

**Batch Gradient Descent:** مجموعه داده انجام می‌شود.

**Stochastic Gradient Descent (SGD):** انجام می‌شود که باعث نوسانات بیشتری در روند یادگیری می‌شود.

**Mini-batch Gradient Descent:** عمیق کاربرد دارد.

## J. چرا Deep Learning برای پیچیده‌ترین مسائل استفاده می‌شود؟

Deep Learning قادر است الگوهای پیچیده‌ای را که الگوریتم‌های سنتی قادر به یادگیری آنها نیستند، شناسایی کند. این روش از شبکه‌های عصبی عمیق بهره می‌برد و در کاربردهایی مانند تشخیص تصویر، پردازش زبان طبیعی (NLP) و تحلیل داده‌های حجمی مورد استفاده قرار می‌گیرد.

### مزایای Deep Learning

- توانایی استخراج ویژگی‌ها به صورت خودکار
  - مقیاس‌پذیری برای داده‌های حجمی
  - قابلیت یادگیری از داده‌های نامرتب و غیرساختاریافته
-

### A. چرا پایتون زبان برنامه‌نویسی محبوبی در علم داده است؟

پایتون به دلیل سادگی در یادگیری، نحو خوانا و قابلیت استفاده برای انواع مختلف برنامه‌ها، به ویژه در حوزه علم داده، یکی از پرکاربردترین زبان‌ها محسوب می‌شود. این زبان به دلیل دارا بودن کتابخانه‌های گسترده و پیشرفته برای پردازش داده‌ها، مانند **Scikit-learn**, **Pandas**, **NumPy**, **Seaborn** و **Matplotlib**، نیازهای مختلف در پردازش، تجزیه و تحلیل و تجسم داده‌ها را به خوبی پوشش می‌دهد. همچنین جامعه فعال پایتون و مستندات فراوان این زبان به راحتی به کاربران کمک می‌کند.

### B. تفاوت **Pandas** و **NumPy** چیست؟

هر دو ابزارهای اصلی در پردازش داده‌ها هستند، اما هدف و نحوه استفاده آنها تفاوت‌هایی دارد. برای انجام محاسبات عددی و علمی بهینه شده است و از آرایه‌های چندبعدی (arrays) برای ذخیره داده‌ها استفاده می‌کند. در مقابل، برای پردازش داده‌های جدولی و مختلط طراحی شده و از ساختارهای داده‌ای مانند **Series** و **Data Frame** برای ذخیره داده‌ها استفاده می‌کند. علاوه بر این، ابزارهای **Pandas** پیشرفته‌ای برای مدیریت داده‌های گمشده و انجام تحلیل‌های آماری فراهم می‌کند.

### C. چرا از **Matplotlib** برای تجسم داده‌ها استفاده می‌شود؟

به عنوان یکی از کتابخانه‌های پایه‌ای در تجسم داده‌ها در پایتون، ابزارهای زیادی را برای ترسیم نمودارهای مختلف از جمله **نمودارهای خطی**، **نمودارهای پراکندگی** (scatter plots) و **نمودارهای هیستوگرام** فراهم می‌کند. استفاده از این کتابخانه به تحلیلگران داده کمک می‌کند تا نتایج خود را به صورت بصری نمایش دهند و الگوهای مهم در داده‌ها را شناسایی کنند. همچنین این کتابخانه قابلیت سفارشی‌سازی بسیاری از جنبه‌های نمودارها مانند رنگ‌ها، فونت‌ها و سبک‌ها را دارد.

### D. چرا **Seaborn** برای تجسم داده‌های پیشرفته کاربرد دارد؟

کتابخانه‌ای است که بر پایه **Matplotlib** ساخته شده و امکانات بیشتری برای تحلیل و تجسم داده‌ها ارائه می‌دهد. این کتابخانه بیشتر برای تحلیل‌های آماری طراحی شده است و امکان رسم انواع نمودارهای **نمودارهای هیستوگرام**, **نمودارهای جعبه‌ای** (box plots) و **نمودارهای حرارتی** (heatmaps) را با کدهای کمتر و ظاهری زیبا فراهم می‌آورد. این ویژگی‌ها باعث می‌شود Seaborn به ابزاری عالی برای تحلیلگران داده تبدیل شود که نیاز به نمایش اطلاعات به صورت پیچیده دارد.

### E. چگونه می‌توانید یک تابع (Function) در پایتون تعریف کنید؟

برای تعریف یک تابع در پایتون از کلمه کلیدی **def** استفاده می‌شود. تابع در پایتون می‌تواند پارامترهای ورودی را بپذیرد و خروجی را با استفاده از دستور **return** بازگرداند. این ویژگی‌ها به

برنامه‌نویس اجازه می‌دهند که قطعاتی از کد را به صورت بلوک‌های مجزا تعریف کرده و در بخش‌های مختلف برنامه فراخوانی کنند.

#### F. چرا از List Comprehension در پایتون استفاده می‌شود؟

یک ویژگی قدرتمند در پایتون است که امکان ایجاد لیست‌ها را با کدهای مختصر و کارآمد فراهم می‌کند. این روش به خصوص زمانی که نیاز به اعمال یک عملیات بر روی مجموعه‌ای از داده‌ها داریم و می‌خواهیم کد مختصر و سریع بنویسیم، بسیار مفید است. علاوه بر این، استفاده از List Comprehension باعث کاهش تعداد خطوط کد می‌شود و عملکرد بهتری نسبت به استفاده از حلقه‌های سنتی دارد.

#### G. چگونه می‌توانید یک فایل CSV را در پایتون بخوانید؟

برای خواندن داده‌های موجود در فایل‌های CSV در پایتون، یکی از ساده‌ترین و سریع‌ترین روش‌ها استفاده از کتابخانه Pandas است. این کتابخانه به راحتی با استفاده از تابع pd.read\_csv() قادر به خواندن فایل‌های CSV و تبدیل آنها به یک Data Frame است که پردازش و تحلیل داده‌ها را بسیار آسان‌تر می‌کند.

#### H. چه تفاوت‌هایی دارند؟ XML و JSON

هر دو فرمتهایی XML (eXtensible Markup Language) و JSON (JavaScript Object Notation) برای تبادل داده بین سیستم‌ها هستند، اما تفاوت‌های اساسی دارند. سبکتر است و ساختاری ساده و قابل خواندن برای انسان دارد. از سوی دیگر، XML به دلیل استفاده از تگ‌های بسته و باز بیشتر در سیستم‌های پیچیده و داده‌های ساختار یافته کاربرد دارد. یکی دیگر از تفاوت‌های مهم، JSON از انواع داده‌های ساده مانند رشته، عدد، آرایه و شیء پشتیبانی می‌کند، در حالی که XML قابلیت استفاده از انواع داده پیچیده‌تر را دارد.

### A. آرایه (Array) و لیست (List) چه تفاوتی دارند؟

در پایتون، **لیست‌ها** ساختار داده‌ای انعطاف‌پذیر هستند که می‌توانند انواع داده‌های مختلف را در خود جای دهند و اندازه آن‌ها قابل تغییر است. برخلاف لیست‌ها، **آرایه‌ها** در زبان‌هایی مانند C یا Java ساختارهای داده‌ای با اندازه ثابت هستند که معمولاً فقط نوع داده‌ای خاصی را می‌پذیرند و برای ذخیره‌سازی داده‌های عددی بهینه شده‌اند.

### B. دیکشنری (Dictionary) در پایتون چگونه کار می‌کند؟

**دیکشنری‌ها** در پایتون ساختارهای داده‌ای هستند که به صورت جفت‌های کلید-مقدار عمل می‌کنند. هر کلید در دیکشنری باید یکتا و غیرقابل تغییر باشد (مانند رشته‌ها یا اعداد)، و مقدار مربوط به آن می‌تواند هر نوع داده‌ای باشد. دیکشنری‌ها به کمک **تابع هش** (hash function) به سرعت به مقادیر دسترسی پیدا می‌کنند که کارایی بالایی را در جستجو، افزودن و حذف عناصر فراهم می‌کند.

•

### C. تاپل (Tuple) و لیست (List) چه تفاوتی دارند؟

**تاپل‌ها و لیست‌ها** هر دو ساختارهای داده‌ای برای ذخیره مجموعه‌ای از عناصر هستند، اما تفاوت‌های کلیدی دارند:

- **تغییرپذیری**: لیست‌ها **تغییرپذیر** (mutable) هستند، به این معنی که می‌توان عناصر آن‌ها را تغییر داد یا حذف کرد. اما تاپل‌ها **تغییرنایپذیر** (immutable) هستند و پس از ایجاد، نمی‌توان آن‌ها را تغییر داد.
- **کارایی**: به دلیل تغییرنایپذیر بودن، تاپل‌ها معمولاً کارایی بهتری در عملیات‌هایی مانند پیمایش دارند و می‌توانند به عنوان کلید در دیکشنری‌ها استفاده شوند، در حالی که لیست‌ها به دلیل انعطاف‌پذیری بیشتر، برای مواردی که نیاز به تغییر داده‌ها داریم مناسب‌تر هستند.

### D. سمت (Set) در پایتون چرا برای حذف داده‌های تکراری استفاده می‌شود؟

**سمت‌ها** در پایتون ساختارهای داده‌ای هستند که مجموعه‌ای از عناصر یکتا را ذخیره می‌کنند و به صورت خودکار داده‌های تکراری را حذف می‌کنند. این ویژگی به دلیل استفاده از **تابع هش** برای ذخیره‌سازی عناصر است که امکان بررسی سریع وجود یا عدم وجود یک عنصر را فراهم می‌کند.

## E. استک (Stack) و کیو (Queue) چه تفاوتی دارند؟

استک‌ها و کیوها هر دو ساختارهای داده‌ای خطی هستند، اما ترتیب دسترسی به عناصر در آنها متفاوت است:

- استک (Stack) از اصل آخر وارد، اول خارج (LIFO) پیروی می‌کند، به این معنی که آخرین عنصری که به استک اضافه می‌شود، اولین عنصری است که حذف می‌شود.
- کیو (Queue) از اصل اول وارد، اول خارج (FIFO) پیروی می‌کند، به این معنی که اولین عنصری که به کیو اضافه می‌شود، اولین عنصری است که حذف می‌شود.

## F. هش‌تیبل (Hash Table) چیست و چرا کاربرد دارد؟

هش‌تیبل یک ساختار داده‌ای است که از یک **تابع هش** برای تبدیل کلیدها به اندیس‌های آرایه‌ای استفاده می‌کند تا امکان دسترسی سریع به مقادیر مربوط به آنها فراهم شود. این ساختار برای پیاده‌سازی دیکشنری‌ها و مجموعه‌ها در زبان‌های برنامه‌نویسی استفاده می‌شود و عملیات‌هایی مانند جستجو، درج و حذف را با زمان تقریبی **O(1)** انجام می‌دهد.

## G. درخت دودویی (B-Tree) و درخت (Binary Tree) چه تفاوتی دارند؟

درخت دودویی یک ساختار داده‌ای است که هر گره آن می‌تواند حداقل دو و فرزند داشته باشد، که به ترتیب فرزند چپ و فرزند راست شناخته می‌شوند. این ساختار برای پیاده‌سازی عملیات‌هایی مانند جستجو و مرتب‌سازی استفاده می‌شود.

درخت B یک نوع درخت جستجوی خودمتعادل است که در سیستم‌های پایگاه داده و سیستم‌های فایل برای ذخیره‌سازی داده‌ها استفاده می‌شود. درخت‌های B به گونه‌ای طراحی شده‌اند که عملیات جستجو، درج و حذف را با کارایی بالا حتی با حجم بالای داده‌ها انجام دهند.

## H. چرا ساختار داده گراف برای شبکه‌های اجتماعی استفاده می‌شود؟

گراف‌ها ساختارهای داده‌ای هستند که برای مدل‌سازی روابط بین موجودیت‌ها (مانند افراد یا صفحات) بسیار مناسب هستند. در شبکه‌های اجتماعی، کاربران به عنوان گره‌ها (nodes) و روابط بین آنها (مانند دوستی یا دنبال کردن) به عنوان یال‌ها (edges) مدل می‌شوند. این مدل‌سازی امکان تحلیل‌هایی مانند شناسایی جوامع، پیشنهاد دوستان و مسیرهای ارتباطی را فراهم می‌کند.

## I. چرا Dynamic Programming در حل مسائل بیچیده کاربرد دارد؟

یک تکنیک بهینه‌سازی است که برای حل مسائل بیچیده با Dynamic Programming (DP) تقسیم آنها به زیرمسائل کوچک‌تر استفاده می‌شود. این روش به جای محاسبه مجدد نتایج مشابه، آنها را ذخیره می‌کند و از این طریق کارایی الگوریتم را بهبود می‌بخشد. این روش به ویژه برای حل مسائل بهینه‌سازی و مسائل که دارای ویژگی **فرعی به هم پوشانی** (Overlapping Subproblems) هستند، بسیار مفید است. به عبارت دیگر، اگر یک مسئله را بتوان به زیرمسائل

کوچکتر تقسیم کرد که ممکن است دوباره محاسبه شوند، DP با ذخیره نتایج قبلی از محاسبات اضافی جلوگیری می‌کند.

### مزایای استفاده از DP:

1. **کاهش پیچیدگی زمانی**: در مسائل پیچیده، با ذخیره نتایج محاسبات قبلی می‌توان تعداد محاسبات مورد نیاز را کاهش داد.

2. **حل مسائل بهینه‌سازی DP**: معمولاً برای حل مسائل بهینه‌سازی مانند **مسئله کوله‌پشتی** (Knapsack Problem) و **مسئله کوتاه‌ترین مسیر** (Shortest Path Problem) استفاده می‌شود.

## J. چیست و چرا در الگوریتم‌های پیشرفته استفاده می‌شود؟

یک تکنیک در برنامه‌نویسی است که در آن تابع خود را فراخوانی می‌کند. این تکنیک به حل مشکلاتی که به طور طبیعی به صورت بازگشتنی تعریف شده‌اند کمک می‌کند. مثلًاً در بسیاری از مسائل ترکیبیاتی مانند **محاسبه فاکتوریل**، **حل معادلات بازگشتنی** یا **جستجو در درخت‌ها و گراف‌ها**، الگوریتم‌های بازگشتنی بهترین گزینه هستند.

### ویژگی‌های مهم Recursion:

1. **پایه بازگشتنی**: هر تابع بازگشتنی باید یک شرایط توقف (base case) مشخص داشته باشد تا از بی‌نهایت فراخوانی جلوگیری کند.

2. **سادگی کد**: الگوریتم‌های بازگشتنی معمولاً کدهای ساده‌تری دارند که به راحتی می‌توانند مسائل پیچیده را مدل‌سازی کنند.

3. **حل مسائل پیچیده**: بسیاری از الگوریتم‌های پیچیده در علوم کامپیوتر مانند **جستجوی عمق اول** (Depth-First Search) در گراف‌ها، **درخت‌های دودویی** و **پیدا کردن راه حل‌های بهینه** به روش بازگشتنی پیاده‌سازی می‌شوند.

**مثال**: برای محاسبه فاکتوریل یک عدد با استفاده از بازگشت:

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
```

## Data Preprocessing 4.

### A. چرا Data Cleaning در علم داده اهمیت دارد؟

(پاکسازی داده‌ها) یکی از مهمترین مراحل در پردازش داده‌هاست، زیرا کیفیت داده‌ها تأثیر مستقیم بر عملکرد مدل‌های یادگیری ماشین و تحلیل‌های آماری دارد. برخی از دلایل اهمیت آن عبارتند از:

- **افزایش دقیقت مدل‌ها**: داده‌های نادرست، ناقص یا نامرتبط می‌توانند مدل‌های یادگیری ماشین را گمراه کنند.
- **کاهش نویز در داده‌ها**: داده‌های اضافی و نویزی می‌توانند تحلیل‌ها را مختل کنند.
- **بهبود کارایی پردازش**: داده‌های پاکسازی شده به منابع پردازشی کمتری نیاز دارند.
- **افزایش قابلیت اطمینان تصمیم‌گیری**: داده‌های تمیزتر منجر به خروجی‌های دقیق‌تر و تصمیمات بهتر می‌شوند.

### B. داده‌های گم شده چگونه مدیریت می‌شوند؟

داده‌های گم شده (Missing Values) می‌توانند به روش‌های مختلف مدیریت شوند:

#### 1. حذف داده‌های دارای مقادیر گم شده :

- اگر درصد داده‌های گم شده کم باشد، حذف سطرها یا ستونها یک گزینه مناسب است.
- جایگزینی با مقدار متوسط، میانه یا مد :

- برای داده‌های عددی، مقدار میانگین (Mean)، میانه (Median) یا مد (Mode) جایگزین می‌شود.

#### 3. استفاده از روش‌های پیشرفته مانند KNN Imputation:

- مقدار گم شده با استفاده از مقادیر مشابه از داده‌های همسایه پر می‌شود.

#### 4. مدل‌سازی داده‌های گم شده :

- در برخی موارد، می‌توان از مدل‌های یادگیری ماشین برای پیش‌بینی و جایگزینی مقادیر گم شده استفاده کرد

### C. Outliers چیست و چگونه می‌توانید آنها را تشخیص دهید؟

(Outliers) داده‌های پرت (به داده‌هایی گفته می‌شود که نسبت به سایر نقاط داده‌ای به طور غیرعادی متفاوت هستند. این داده‌ها ممکن است ناشی از خطاهای اندازه‌گیری یا تغییرات واقعی باشند.

## روش‌های تشخیص Outliers

### 1. استفاده از Z-Score

- داده‌هایی که Z-Score آنها خارج از محدوده  $3 \pm$  باشد، معمولاً پر در نظر گرفته می‌شوند.

### 2. استفاده از IQR (Interquartile Range)

- داده‌هایی که خارج از محدوده  $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$  باشند، پر محسوب می‌شوند.

### 3. بصری‌سازی داده‌ها

- نمودارهای مانند Scatter Plot و Box Plot می‌توانند Outliers را نشان دهند.

### 4. استفاده از الگوریتم‌های یادگیری ماشین

- روش‌هایی مانند DBSCAN و Isolation Forest می‌توانند Outliers را شناسایی کنند.

## D. تبدیل داده‌ها چه کاربردی دارد؟

**Data Transformation** (تبدیل داده‌ها) فرآیند تغییر شکل داده‌ها به فرم مناسب‌تر برای تحلیل و مدل‌سازی است.

### دلالت اهمیت Data Transformation:

- کاهش پراکندگی داده‌ها**: برخی داده‌ها دارای توزیع‌های نامناسب هستند که با تبدیل‌های لگاریتمی یا جذر، پراکندگی آنها کاهش می‌یابد.
- ایجاد نرمال‌سازی در داده‌ها**: بسیاری از الگوریتم‌های یادگیری ماشین نیاز به داده‌هایی با توزیع نرمال دارند.
- بهبود همخطی بین متغیرها**: برخی تبدیلات باعث کاهش وابستگی‌های نامطلوب میان ویژگی‌ها می‌شوند.
- استانداردسازی و نرمال‌سازی**: تبدیل داده‌ها به محدوده‌ای مشخص (مانند  $[0,1]$  یا  $[-1,1]$ ) باعث بهبود عملکرد مدل می‌شود.

## E. Encoding Techniques (One-Hot Encoding چه تفاوتی دارد؟)

**Encoding Techniques** روش‌هایی برای تبدیل داده‌های طبقه‌ای (Categorical Data) به داده‌های عددی هستند.

تکنیک	توضیح	مزایا	معایب
<b>Label Encoding</b>	مقدار عددی یکتاوی به هر کلاس اختصاص می‌دهد	سادگی و کارایی بالا	ممکن است باعث ایجاد روابط کاذب بین مقادیر شود
<b>One-Hot Encoding</b>	برای هر کلاس یک ستون باینری ایجاد می‌کند	مناسب برای داده‌های بدون ترتیب	تعداد ویژگی‌ها را افزایش داده و حافظه بیشتری مصرف می‌کند

### F. اهمیت Feature Selection در Model-building

انتخاب ویژگی (Feature Selection) به فرآیند انتخاب مهم‌ترین ویژگی‌های داده برای بهبود عملکرد مدل اشاره دارد.

#### اهمیت Feature Selection:

- کاهش پیچیدگی مدل: ویژگی‌های غیرضروری حذف می‌شوند.
- بهبود دقیقت مدل: استفاده از ویژگی‌های مرتبط باعث افزایش دقیقت می‌شود.
- کاهش Overfitting: حذف ویژگی‌های اضافی مانع یادگیری الگوهای تصادفی می‌شود.
- افزایش سرعت پردازش: مدل‌های کوچک‌تر سریع‌تر اجرا می‌شوند.

### G. داده‌های تکراری چگونه در پایگاه داده‌ها حذف می‌شود؟

حذف داده‌های تکراری باعث بهبود کارایی و دقیقت تحلیل داده می‌شود.

روش‌های حذف داده‌های تکراری

استفاده از DROP DUPLICATES در پایتون(Pandas)

```
df.drop_duplicates(inplace=True)
```

- استفاده از SQL

```
DELETE FROM table_name WHERE rowid NOT IN (SELECT MIN(rowid) FROM table_name GROUP BY column_name);
```

- بررسی داده‌های تکراری بر اساس چندین ستون

```
df.drop_duplicates(subset=['column1', 'column2'], keep='first', inplace=True)
```

## H. داده های نامرتب چه مشکلاتی را در پیش‌بینی‌های Machine Learning ایجاد می‌کند؟

داده‌های نامرتب مشکلاتی از جمله موارد زیر ایجاد می‌کنند:

- **کاهش دقت مدل**: ویژگی‌های نامرتب باعث یادگیری اطلاعات غیرضروری توسط مدل می‌شوند.
- **افزایش پیچیدگی مدل**: پردازش ویژگی‌های اضافی باعث کندی محاسبات می‌شود.
- **افزایش Overfitting**: داده‌های نامرتب ممکن است مدل را به سمت یادگیری الگوهای نادرست سوق دهند.

### I. چرا برای پر کردن Missing Values Data Imputation کاربرد دارد؟

تکنیکی برای پر کردن داده‌های گمشده است که به دلایل زیر اهمیت دارد:

- **حفظ حداقل داده‌ها**: حذف داده‌ها می‌تواند اطلاعات ارزشمندی را از بین ببرد.
- **کاهش نویز در داده‌ها**: روش‌هایی مانند میانگین، میانه یا مدل‌سازی از داده‌های موجود برای جایگزینی مقادیر گمشده استفاده می‌کنند.
- **بهبود عملکرد مدل‌های یادگیری ماشین**: داده‌های کامل‌تر باعث عملکرد بهتر مدل‌ها می‌شوند.

### J. چگونه می‌توانید Normality را در داده‌های عددی بررسی کنید؟

برای بررسی نرمال بودن داده‌ها، روش‌های زیر استفاده می‌شوند:

#### 1. آزمون Shapiro-Wilk

```
from scipy.stats import shapiro  
stat, p = shapiro(data)  
print(f'Statistics={stat}, p={p}')
```

- 
- گر مقدار **p-value < 0.05** باشد، داده‌ها نرمال نیستند.

#### آزمون Kolmogorov-Smirnov

- روش دیگری برای بررسی توزیع داده‌ها است.

#### Box Plot و Histogram

- نمایش چشمی توزیع داده‌ها برای بررسی نرمال بودن

## نمودار Q-Q Plot

```
import scipy.stats as stats  
import matplotlib.pyplot as plt  
stats.probplot(data, dist="norm", plot=plt)  
plt.show()
```

اگر نقاط به طور تقریبی روی یک خط قرار گیرند، داده‌ها نرمال هستند.

---

## : Visualization

[چرا Seaborn برای تجسم داده‌های پیشرفته استفاده می‌شود؟]

**ChatGPT said:**

چرا Line Chart برای نمایش رابطه‌های خطی استفاده می‌شود؟

نمودار خطی (برای نمایش روند تغییرات متوالی یک متغیر در طول زمان یا در یک دامنه خاص بسیار مناسب است.

♦ دلایل استفاده از:

- نمایش روندهای پیوسته و تغییرات تدریجی
- مناسب برای تحلیل سری‌های زمانی
- قابلیت تشخیص الگوهای خطی و غیرخطی
- امکان مقایسه چندین روند مختلف در یک نمودار

مثال در پایتون:

```
import matplotlib.pyplot as plt
```

```
x = [1, 2, 3, 4, 5]
```

```
y = [2, 3, 5, 7, 11]
```

```
plt.plot(x, y, marker='o', linestyle='-', color='b')
```

```
plt.xlabel('X-axis')
```

```
plt.ylabel('Y-axis')
```

```
plt.title('Line Chart Example')
```

```
plt.show()
```

**جرا Bar Chart B.** برای مقایسه داده‌های گروهی کاربرد دارد؟

**Bar Chart (نمودار میله‌ای)** (برای مقایسه مقادیر بین گروه‌های مختلف استفاده می‌شود).

♦ دلایل استفاده از **Bar Chart:**

- نمایش داده‌های دسته‌بندی شده (**Categorical Data**)
- مقایسه مستقیم بین گروه‌ها
- امکان نمایش داده‌ها به صورت عمودی یا افقی

مثال در پایتون:

```
import seaborn as sns  
import matplotlib.pyplot as plt  
categories = ['A', 'B', 'C', 'D']  
values = [10, 20, 15, 25]  
sns.barplot(x=categories, y=values)  
plt.xlabel('Categories')  
plt.ylabel('Values')  
plt.title('Bar Chart Example')  
plt.show()
```

**جرا Scatter Plot C.** برای نمایش رابطه‌های غیرخطی استفاده می‌شود؟

**Scatter Plot (نمودار پراکندگی)** (برای نمایش رابطه بین دو متغیر عددی استفاده می‌شود).

♦ دلایل استفاده از **Scatter Plot:**

- نمایش همبستگی (**Correlation**) بین دو متغیر
- مناسب برای تحلیل رابطه‌های غیرخطی و الگوهای پراکنده
- تشخیص **Outliers** در داده‌ها

مثال در پایتون:

```
import numpy as np  
x = np.random.rand(50)  
y = x**2 + np.random.rand(50) * 0.1 # یک رابطه غیرخطی  
plt.scatter(x, y, color='r')  
plt.xlabel('X-axis')  
plt.ylabel('Y-axis')  
plt.title('Scatter Plot Example')  
plt.show()
```

---

چرا Bubble Chart برای نمایش سه متغیر استفاده می‌شود؟

Bubble Chart مشابه Scatter Plot است اما از اندازه حباب‌ها (Bubbles) برای نمایش متغیر سوم استفاده می‌کند.

#### ◆ دلایل استفاده از Bubble Chart:

- نمایش سه بعد اطلاعات در یک نمودار دو بعدی
- مناسب برای تحلیل‌های چند متغیره
- امکان اندازه‌گیری تأثیر متغیر سوم بر دو متغیر دیگر

مثال در پایتون:

```
x = [1, 2, 3, 4, 5]  
y = [10, 20, 25, 30, 50]  
size = [100, 200, 300, 400, 500] # متغیر سوم
```

```
plt.scatter(x, y, s=size, alpha=0.5, color='g')  
plt.xlabel('X-axis')  
plt.ylabel('Y-axis')  
plt.title('Bubble Chart Example')  
plt.show()
```

---

**E.** چرا Heatmap برای نمایش رابطه‌های بین متغیرها کاربرد دارد؟

نmodار حرارتی (به صورت ماتریسی از رنگ‌ها برای نشان دادن قدرت رابطه Heatmap (Correlation) بین متغیرها استفاده می‌شود.

♦ دلایل استفاده از Heatmap:

- نمایش ماتریس همبستگی (Correlation Matrix)
- تشخیص رابطه‌های مثبت و منفی بین متغیرها
- نمایش مقادیر داده‌های ماتریسی به صورت تصویری

مثال در پایتون:

```
import seaborn as sns  
import pandas as pd  
  
data = pd.DataFrame({  
    'A': [1, 2, 3, 4, 5],  
    'B': [2, 4, 6, 8, 10],  
    'C': [5, 3, 8, 6, 2]  
})  
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')  
plt.title('Heatmap Example')  
plt.show()
```

**F.** چرا Pairplot برای تحلیل روابط بین متغیرها کاربرد دارد؟

نmodارهای پراکندگی (Scatter Plot) بین تمامی جفت‌های متغیرهای عددی را Pairplot نمایش می‌دهد.

♦ دلایل استفاده از Pairplot:

- نمایش همبستگی متقابل بین تمامی متغیرها
- تشخیص الگوهای خوش‌ای و داده‌های پرت
- تحلیل سریع روابط بین متغیرها در داده‌های چندبعدی

مثال در پایتون:

```
import seaborn as sns

df = sns.load_dataset('iris')
sns.pairplot(df, hue='species')
plt.show()
```

G. چرا Boxplot برای تشخیص Outliers استفاده می‌شود؟

نمودار جعبه‌ای (پراکنده‌گی داده‌ها و Outliers را نمایش می‌دهد.

♦ دلایل استفاده از Boxplot:

- نمایش میانه (Median)، چارک‌ها و دامنه داده‌ها
- شناسایی Outliers با استفاده از IQR (Interquartile Range)
- مناسب برای مقایسه توزیع داده‌ها در گروه‌های مختلف

مثال در پایتون:

```
sns.boxplot(x=df['sepal_width'])
plt.title('Boxplot Example')
plt.show()
```

H. چرا Histogram برای نمایش توزیع داده‌ها کاربرد دارد؟

تعداد داده‌های موجود در هر بازه را نمایش می‌دهد و برای تحلیل توزیع داده‌ها استفاده می‌شود.

♦ دلایل استفاده از Histogram:

- نمایش توزیع فراوانی داده‌ها
- تشخیص نرمال یا غیرنرمال بودن داده‌ها
- نمایش چگالی داده‌ها در محدوده‌های خاص

مثال در پایتون:

```
sns.histplot(df['sepal_length'], bins=20, kde=True)
plt.title('Histogram Example')
plt.show()
```

---

چگونه می‌توانید یک 3D Plot را در Python ایجاد کنید؟

مثال ایجاد نمودار سه‌بعدی در Python با Matplotlib:

```
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x = np.random.rand(50)
y = np.random.rand(50)
z = np.random.rand(50)
ax.scatter(x, y, z, c='r', marker='o')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
plt.title('3D Scatter Plot Example')
plt.show()
```

---

J. **Seaborn** برای تجسم داده‌های پیشرفته استفاده می‌شود؟

یک کتابخانه تجسم داده‌ها در Python است که بر پایه **Matplotlib** ساخته شده و **Seaborn** ویژگی‌های قدرتمندی دارد.

## ♦ دلایل استفاده از Seaborn:

- سادگی در ایجاد نمودارهای پیچیده
- امکان سفارشی‌سازی بیشتر فته
- پشتیبانی از داده‌های دسته‌بندی‌شده و چندبعدی
- توابع آماده برای ...Heatmap, Pairplot, Violin Plot

مثال در پایتون:

```
sns.set(style="darkgrid")
sns.histplot(df['petal_length'], bins=30, kde=True)
plt.title('Seaborn Example')
plt.show()
```

---

### چرا A. در مدیریت داده‌ها کاربرد دارد؟ Distributed Ledger Technology (DLT)

( فناوری DLT دفتر کل توزیع شده (یک سیستم غیرمت مرکز است که به چندین کاربر اجازه می‌دهد داده‌ها را بدون نیاز به یک مرجع مرکزی ذخیره و مدیریت کنند.

#### ♦ دلایل کاربرد DLT در مدیریت داده‌ها:

- افزایش امنیت: جلوگیری از تغییر غیرمجاز داده‌ها
- شفافیت و اعتماد: هر تغییری در دفتر کل برای همه اعضا قابل مشاهده است
- کاهش هزینه‌ها: حذف واسطه‌ها و مدیریت غیرمت مرکز داده‌ها
- مقیاس‌پذیری و دسترس‌پذیری: داده‌ها بین نودهای مختلف توزیع می‌شود

مثال: در سیستم‌های بانکی، زنجیره تأمین، رأی‌گیری الکترونیکی، و هویت دیجیتال استفاده می‌شود.

---

### چرا B. برای ذخیره‌سازی داده‌های امن استفاده می‌شود؟ Blockchain

یک نوع DLT است که داده‌ها را در قالب **بلوک‌های رمزگاری شده** ذخیره می‌کند و به هم متصل می‌کند.

#### ♦ چرا امنیت بالایی دارد؟

- تغییرناپذیری (Immutability): بعد از ثبت، داده‌ها قابل تغییر نیستند
- رمزگاری قوی: استفاده از هشینگ (SHA-256) برای امنیت
- اجماع توزیع شده: تأیید تراکنش‌ها از طریق الگوریتم‌های اجماع مانند PoS و PoW
- مقاومت در برابر حملات: هیچ نقطه مرکزی برای حمله وجود ندارد

مثال: بیت‌کوین و اتریوم از بلاکچین برای ذخیره‌سازی امن تراکنش‌ها استفاده می‌کنند.

### چرا C. در علم داده پیشرفته کاربرد دارند؟ GANs (Generative Adversarial Networks)

شامل دو شبکه عصبی است: GANs

1. داده‌های جعلی تولید می‌کند Generator:

2. داده‌های جعلی را از واقعی تشخیص می‌دهد Discriminator:

## ◆ کاربردهای GANs:

- تولید تصاویر مصنوعی Deepfake
- افزایش کیفیت تصاویر و ویدیوها
- ساخت داده‌های مصنوعی برای آموزش مدل‌های یادگیری ماشین
- بازسازی داده‌های از دست رفته

مثال: تولید چهره‌های غیرواقعی با StyleGAN یا افزایش کیفیت تصاویر پزشکی.

PCA و t-SNE چه تفاوتی دارند؟

هر دو روش برای کاهش ابعاد داده‌ها استفاده می‌شوند اما تفاوت‌های مهمی دارند:

PCA	t-SNE	ویژگی
خطی	غیرخطی	نوع کاهش ابعاد
سریع‌تر	کندر	سرعت
ضعیفتر	بهتر	حفظ ساختار خوبه‌بندی
آسان‌تر	سخت‌تر	تفسیر داده‌ها
داده‌های پیچیده و غیرخطی	داده‌های پیچیده و غیرخطی	مناسب برای

مثال در پایتون:

```
from sklearn.manifold import TSNE  
from sklearn.decomposition import PCA
```

---

چرا UMAP برای Dimensionality Reduction استفاده می‌شود؟

UMAP (Uniform Manifold Approximation and Projection) یک روش پیشرفته برای کاهش ابعاد داده‌ها است که از نظریه توپولوژی استفاده می‌کند.

## ◆ مزایای UMAP:

- سریع‌تر از t-SNE
- حفظ بهتر ساختار داده‌ها
- کارایی بالا در داده‌های بزرگ و پیچیده

مثال در پایتون:

```
import umap  
reducer = umap.UMAP()  
embedding = reducer.fit_transform(data)
```

---

چرا NetworkX برای تحلیل گراف‌ها کاربرد دارد؟

یک کتابخانه پایتون برای مدلسازی و تحلیل گراف‌های شبکه‌ای است.

♦ کاربردهای NetworkX:

- تحلیل شبکه‌های اجتماعی (Instagram, Twitter)
- بررسی مسیرهای بهینه (Google Maps)
- مدیریت داده‌های ارتباطی در زیست‌شناسی و فیزیک

مثال در پایتون:

```
import networkx as nx  
G = nx.Graph()  
G.add_edges_from([(1, 2), (2, 3), (3, 4)])  
nx.draw(G, with_labels=True)
```

---

چرا NLTK برای پردازش زبان طبیعی (NLP) استفاده می‌شود؟

یکی از محبوب‌ترین کتابخانه‌های NLP در پایتون است.

♦ چرا استفاده می‌شود؟

- پیش‌پردازش متن: توکن‌سازی، حذف توقف‌کلمات، Lemmatization
- تحلیل زبان‌شناسی: استخراج ریشه واژه‌ها
- مدل‌سازی زبان: برچسب‌گذاری گفتار (POS Tagging), Named Entity Recognition

مثال در پایتون:

```
import nltk  
from nltk.tokenize import word_tokenize  
  
text = "Hello, how are you?"  
tokens = word_tokenize(text)  
print(tokens)
```

---

## چرا Scrapy برای Web Scraping کاربرد دارد؟

یک فریمورک قدرتمند برای خریدن (Crawling) و جمعآوری داده‌ها از وب است.

### • چرا Scrapy ؟

- سرعت بالا در استخراج داده‌ها
- قابلیت خریدن چندین صفحه به صورت خودکار
- ذخیره داده‌ها در JSON, CSV, SQL و ...

مثال در پایتون:

```
import scrapy  
  
class MySpider(scrapy.Spider):  
    name = "example"  
    start_urls = ['https://example.com']  
  
    def parse(self, response):  
        title = response.css('title::text').get()  
        print(title)
```

---

**چرا استفاده می‌شود؟** Web Scraping برای BeautifulSoup استفاده می‌شود؟  
یک کتابخانه ساده برای استخراج اطلاعات از HTML و XML است.

◆ **چرا استفاده می‌شود؟**

- مناسب برای آنالیز صفحات HTML ساده
- سادگی در استفاده نسبت به Scrapy
- پردازش ساختارهای پیچیده HTML

مثال در پایتون:

```
from bs4 import BeautifulSoup
import requests

url = "https://example.com"
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

print(soup.title.text) # استخراج عنوان صفحه
```

---

• منابع

• مرتبط با Blockchaing DLT

• کتاب "Mastering Blockchain" - Imran Bashir

• مقاله‌های ACM و IEEE درباره فناوری دفتر کل توزیع شده

• مستندات Ethereum و Hyperledger

2. منابع مرتبط با GANs و کاهش ابعاد داده‌ها (PCA, t-SNE, UMAP)

• کتاب "Deep Learning" - Ian Goodfellow, Yoshua Bengio, Aaron Courville

• مقاله "Visualizing Data using t-SNE" - Laurens van der Maaten & Geoffrey Hinton

• مستندات رسمی UMAP-learn و scikit-learn

3. منابع مرتبط با NetworkX و تحلیل شبکه‌ها

• کتاب "Network Science" - Albert-László Barabási

• مستندات رسمی NetworkX (<https://networkx.github.io/>)

• مقالات مرتبط با تحلیل شبکه‌های اجتماعی و داده‌های گراف

4. منابع مرتبط با NLTK و پردازش زبان طبیعی (NLP)

• کتاب "Speech and Language Processing" - Daniel Jurafsky & James H. Martin

• مستندات NLTK (<https://www.nltk.org/>)

• مقاله‌های EMNLP و ACL در زمینه پردازش متن

5. منابع مرتبط با BeautifulSoup و Scrapy و Web Scraping

• مستندات رسمی Scrapy (<https://docs.scrapy.org/>)

• مستندات BeautifulSoup

(<https://www.crummy.com/software/BeautifulSoup/>)

• کتاب "Web Scraping with Python" - Ryan Mitchell

6. منابع عمومی و مستندات کتابخانه‌های پایتون

• scikit-learn: <https://scikit-learn.org>

• <https://pytorch.org> و PyTorch: <https://www.tensorflow.org> و TensorFlow

• و Seaborn: <https://matplotlib.org> و Matplotlib

<https://seaborn.pydata.org>