«*In The Name Of GOD*»

[HW-01-Report]

[MACHINE LEARNING]

Hasan Masroor | [403131030] | February 5, 2025

"فهرست مطالب تمرين 01"

Preprocessing Job Experience Data for Predictive Modeling

.A

در بخش اول باید دیتاست exp.csv را آپلود کنیم و بعد از آن 5 نمونه از دیتاست را نمایش دهیم. (برای این کار می‌توانیم از ()head. برای نمایش 5 نمونه اول، ()tail. برای نمایش 5 نمونه آخر و یا iloc[i:j]. برای نمایش نمونه‌های بین i و j-1 استفاده نماییم.)

به عنوان مثال برای نمایش 5 نمونه اول داریم:

	Unnamed: 0	id	title	time_type_full_or_part	duration_of_cooperation	location_hybrid	company_url	company_name
0	0	1	Search Engine Optimization Specialist	Pixune · Freelance	Jul 2022 - Present · 2 yrs 1 mo	Poland · Remote	https://www.linkedin.com/search/results/all/?k...	NaN
1	1	2	Search Engine Optimization Manager	Cofinfo · Self-employed	Dec 2022 - Present · 1 yr 8 mos	NaN	https://www.linkedin.com/search/results/all/?k...	NaN
2	2	3	Search Engine Optimization Specialist	Karnakon · Full-time	Dec 2020 - Aug 2022 · 1 yr 9 mos	Tehran, Iran	https://www.linkedin.com/search/results/all/?k...	NaN
3	3	4	Trader	Forex Trading Online · Apprenticeship	Apr 2020 - Present · 4 yrs 4 mos	NaN	https://www.linkedin.com/company/18372839/	Forex Trading Online
4	4	5	In personal marketing	Home Design Decoration ideas · Part-time	Jan 2018 - Present · 6 yrs 7 mos	Shiraz County, Fars, Iran	https://www.linkedin.com/company/13660967/	Home Design Decoration ideas

.B

در پارت دوم باید داده‌های گمشده را هندل کنیم. ابتدا ستون‌های دیتاست را به همراه مقادیر گمشده مربوط به هر ستون در خروجی چاپ می‌کنیم که بعد برای هندل‌کردن بهتر این داده‌های گمشده تصمیم بهتری بگیریم و خروجی این بخش به صورت زیر است:

```
Missing values in each column:
Unnamed: 0                    0
id                            0
title                         5
time_type_full_or_part       19
duration_of_cooperation     861
location_hybrid            3221
company_url                  41
company_name               3282
dtype: int64
```

می‌بینیم که در شش ستون مقادیر گمشده داریم و در ستون‌های title، time_type_full_or_part و company_url تعداد مقادیر گمشده کم‌تری نسبت به کل داده‌ها داریم و به نظر می‌رسد حذف سطرهای این ستون‌ها که شامل مقادیر گمشده هستند راهکار خوبی باشد. ستون duration_of_cooperation مقادیر گمشده بیشتری دارد و به جای حذف بهتر است از روش‌های بهتری مثل جایگزین‌کردن با میانگین، میانه و ... برای این ستون استفاده کرد؛ اما با کمی دقت و بررسی این ستون در دیتاست متوجه می‌شویم که شامل مقادیر متنی است و برای استفاده از میانگین و ... ابتدا باید داده‌های متنی این ستون را به داده‌های عددی (تبدیل به ماه) تبدیل کنیم و به عبارتی پارت C را کمی زودتر و در این بخش باید انجام دهیم که بعد بتوانیم مقادیر گمشده این ستون را بهتر هندل کنیم.

با استفاده از عبارات منظم حالات مختلف عبارات این ستون را در نظر می‌گیریم و بررسی می‌کنیم، ابتدا تابع extract_duration_in_months با استفاده از عبارات منظم، مدت زمان همکاری را از فرمت‌های مختلف مانند سال و ماه استخراج کرده و آن‌ها را به تعداد ماه‌ها تبدیل می‌کند؛ سپس به کمک تابع apply این تابع در ستون cooperation_duration_months ذخیره می‌شود و هر مقدار متنی به تعداد ماه‌ها تبدیل می‌شود:

	duration_of_cooperation	cooperation_duration_months
0	Jul 2022 - Present · 2 yrs 1 mo	25.0
1	Dec 2022 - Present · 1 yr 8 mos	20.0
2	Dec 2020 - Aug 2022 · 1 yr 9 mos	21.0
3	Apr 2020 - Present · 4 yrs 4 mos	52.0
4	Jan 2018 - Present · 6 yrs 7 mos	79.0

بعد از اینکه بخش بالا را که همان پارت C نیز می‌باشد را انجام دادیم، حالا به سراغ هندل‌کردن مقادیر گمشده ستون duration_of_cooperation می‌رویم. میانگین ماه‌ها را به کمک بخش قبل بدست می‌آوریم و سپس جایگزین مقادیر گمشده می‌کنیم؛ همچنین برای یکدست‌شدن داده‌های این ستون که در دیتاست به صورت متنی قرار داشتند می‌توانیم این مقادیر گمشده که حالا با میانگین جایگزین شده‌اند را به فرم متنی اولیه تبدیل کنیم و در نهایت به عنوان مثال iloc[10:15]. را که سطرهای 10 تا 14 می‌شود را نمایش می‌دهیم (به این خاطر این بخش از دیتاست را برای نمایش انتخاب کردیم که مقدار 11 قبلا جزو مقادیر گمشده بوده است و حالا با مقدار میانگین پر شده است و به خوبی می‌توانیم از کارمان اطمینان حاصل کنیم):

	duration_of_cooperation	cooperation_duration_months
10	1 yrs 0 mos	12
11	2 yrs 7 mos	31
12	1 yrs 6 mos	18
13	2 yrs 1 mos	25
14	1 yrs 8 mos	20

یک نکته مهم در این قسمت این بود که در ابتدا که بخش بالا را پیاده‌سازی کردیم به دلیل اینکه میانگین این ستون اعشاری در آمد و مقادیر گمشده که با این مقدار جایگزین شدند ظاهری خوبی نداشتند، از تابع round استفاده کردیم که مقادیر را گرد کنیم و تصویر بالا حاصل شود.

حالا به سراغ دو ستون location_hybrid و company_name می‌رویم که شامل مقادیر گمشده زیادی بودند و مثل قبل یک راه این است که این موارد را حذف کنیم اما راه منطقی و درستی نیست و بهتر است که مقادیر گمشده این دو ستون را با یکی از مواردی که بالا توضیح دادیم جایگزین کنیم و در اینجا ما از مد استفاده کردیم و این مقادیر را با داده‌ای که در بیشترین فراوانی را در این دو ستون دارد جایگزین کردیم و در نهایت هم پس از هندل‌کردن مقادیر گمشده، برای اطمینان خروجی را نمایش می‌دهیم و بررسی می‌کنیم:

```
Missing values after handling:
Unnamed: 0                    0
id                            0
title                         0
time_type_full_or_part        0
duration_of_cooperation       0
location_hybrid               0
company_url                   0
company_name                  0
cooperation_duration_months   0
dtype: int64
```

C.

در این بخش هم باید ویژگی‌های عددی را از ستون Duration of Cooperation استخراج کنیم که در پارت قبل برای مقادیر گمشده این ستون مجبور شدیم این قسمت را زودتر پیاده‌سازی کنیم.

با استفاده از عبارات منظم حالات مختلف عبارات این ستون را در نظر می‌گیریم و بررسی می‌کنیم، ابتدا تابع extract_duration_in_months با استفاده از عبارات منظم، مدت زمان همکاری را از فرمت‌های مختلف مانند سال و ماه استخراج کرده و آن‌ها را به تعداد ماه‌ها تبدیل می‌کند؛ سپس به کمک تابع apply این تابع در ستون cooperation_duration_months ذخیره می‌شود و هر مقدار متنی به تعداد ماه‌ها تبدیل می‌شود:

	duration_of_cooperation	cooperation_duration_months
0	Jul 2022 - Present · 2 yrs 1 mo	25.0
1	Dec 2022 - Present · 1 yr 8 mos	20.0
2	Dec 2020 - Aug 2022 · 1 yr 9 mos	21.0
3	Apr 2020 - Present · 4 yrs 4 mos	52.0
4	Jan 2018 - Present · 6 yrs 7 mos	79.0

تفسیر برخی از این عبارت‌های منظم به‌کاررفته در کد نیز به شرح زیر می‌باشد:

- (+\d): این قسمت عدد مربوط به سال را پیدا می کند.
- +s\: این قسمت به بررسی اینکه حداقل یک فاصله بعد از عدد مربوط به سال وجود داشته باشد می پردازد.
- ?yrs: این قسمت به یافتن شکل صحیح سال می پردازد و ?s برای انتخاب هم yr و هم yrs است.
- (*\d): این قسمت بررسی می کند که عدد مربوط به ماه وجود داشته باشد.
- ?mos: مانند ?yrs عمل می کند و اینجا هم ?s برای تشخیص هم mo و هم mos است.

D.

در این پارت هم با حذف کاراکترهای ناخواسته، نرمال‌سازی متن و شناسایی نقش‌های کلیدی به پیش‌پردازش عناوین شغلی و شرکت‌ها می‌پردازیم. ابتدا یک لیست از نقش‌های کلیدی و مهم مثل manager، engineer، analyst و ... ایجاد می‌کنیم؛ سپس یک تابع به نام remove_punctuation برای حذف علائم نگارشی و کاراکترهای غیرضروری از داده‌ها نوشتیم. (این تابع از()isalnum برای نگه‌داری فقط حروف و اعداد استفاده می‌کند و از ()isspace نیز برای حفظ اسپیس‌ها بهره می‌برد)

در مرحله بعد تابع process_row را تعریف می‌کنیم که به ازای هر سطر در داده‌ها، ابتدا عنوان شغلی و نام شرکت را به حروف کوچک تبدیل می‌کند (برای جلوگیری از رخ‌دادن خطاهای احتمالی) و همچنین تمام فضای اضافی با استفاده از ()strip حذف می‌شود؛ سپس در داخل هر عنوان شغلی به دنبال نقش‌های کلیدی می‌گردیم و اگر یکی از این نقش‌ها در عنوان شغلی پیدا شود، آن نقش به عنوان key_role اضافه می‌شود و در غیر این صورت مقدار 'other' برای نقش شغلی قرار می‌دهیم. در نهایت سه ستون جدید **title_cleaned**، **company_name_cleaned** و **key_role** به دیتافریم اضافه می‌شود و خروجی به شکل زیر حاصل می‌شود:

	title	title_cleaned	company_name	company_name_cleaned	key_role
0	Search Engine Optimization Specialist	search engine optimization specialist	TAPSI	tapsi	specialist
1	Search Engine Optimization Manager	search engine optimization manager	TAPSI	tapsi	manager
2	Search Engine Optimization Specialist	search engine optimization specialist	TAPSI	tapsi	specialist
3	Trader	trader	Forex Trading Online	forex trading online	other
4	In personal marketing	in personal marketing	Home Design Decoration ideas	home design decoration ideas	other

.E

در این قسمت باید با استفاده از تکنیک‌های One-Hot Encoding یا Label Encoding ستون location_hybrid را به مقادیر عددی تبدیل کنیم، ابتدا به توضیح کوتاهی در رابطه با این دو کدگذاری و تاثیر آنها بر مدل می‌پردازیم و سپس پیاده‌سازی کد را توضیح می‌دهیم.

One-Hot Encoding: این تکنیک برای داده‌هایی که هیچ ترتیب یا رابطه‌ای بین دسته‌ها ندارند استفاده می‌شود. به ازای هر دسته یک ستون جدید ایجاد می‌شود و اگر آن دسته در آن ردیف موجود باشد مقدار آن ستون 1 می‌شود و در غیر این صورت 0 می‌شود. این تکنیک می‌تواند اطلاعات هر دسته را حفظ کند و هیچ ترتیبی بین دسته‌ها در نظر نمی‌گیرد اما از آن طرف اگر تعداد دسته‌ها زیاد باشد، تعداد ستون‌ها افزایش قابل‌توجهی پیدا می‌کند و با بزرگ‌شدن ابعاد داده‌ها می‌تواند منجر به معضل ابعاد، هزینه‌های محاسباتی زیاد و ... شود.

Label Encoding: در این تکنیک به هر دسته یک عدد صحیح اختصاص داده می‌شود و این روش بیشتر برای داده‌هایی که دارای ترتیب خاصی هستند استفاده می‌شود. برخلاف تکنیک قبلی حافظه و زمان کم‌تری را مصرف می‌کند و برای مدل‌هایی که وابستگی و یک ترتیبی دارند می‌تواند مناسب باشد اما اگر داده‌ها ترتیبی نباشند می‌تواند به فرضیات اشتباهی ختم شود و مثلا ممکن است فرض کند که مقداری که در لیبل 2 قرار دارد از مقدار لیبل 1 بزرگتر است و مقایسات اشتباهی انجام دهد.

با توجه به توضیحات گفته‌شده برای این مدل استفاده از تکنیک One-Hot Encoding مناسب‌تر است. ابتدا از pd.get_dummies() برای ایجاد متغیرهای باینری برای هر موقعیت جغرافیایی استفاده می‌کنیم. هر موقعیت جغرافیایی منحصربه‌فرد یک ستون جدید به نام location_ ... می‌گیرد و بعد از آن با استفاده از applymap()، مقادیر True به 1 و مقادیر False به 0 تبدیل می‌شوند.

بخشی از ستون‌های خروجی به شکل زیر نمایش داده شده است:

company_name_cleaned	...	location_ایران	location_برج اوران	location_تهران	location_تهران - On-site	location_تهران مجتمع شهید حدادی
tapsi	...	0	0	0	0	0
tapsi	...	0	0	0	0	0
tapsi	...	0	0	0	0	0
forex trading online	...	0	0	0	0	0
home design decoration ideas	...	0	0	0	0	0

.F

در این قسمت باید نقاط پرت را با استفاده از تکنیک های z-score یا IQR شناسایی و در صورت لزوم حذف کنیم. ابتدا از ()quantile برای محاسبه Q1 (25 درصد) و Q3 (75 درصد) استفاده می‌شود تا نقاط مرزی داده‌ها مشخص شود، سپس IQR (بازه بین Q1 وQ3) محاسبه می‌شود که نشان‌دهنده محدوده‌ای است که داده‌های معتبر باید در آن قرار داشته باشند. برای شناسایی نقاط پرت، حد پایین (lower_bound) و حد بالا (upper_bound) تعریف می‌شوند که از 1.5 برابر IQR فاصله دارند و هر داده‌ای که خارج از این محدوده باشد به عنوان نقطه پرت شناسایی می‌شود؛ در نهایت داده‌های تمیز شده که شامل نقاط پرت نیستند در exp_data_cleaned_iqr ذخیره می‌شوند و خروجی را نمایش می‌دهیم:

```
number of outliers: 953
       cooperation_duration_months
count                  9005.000000
mean                     21.820544
std                      14.472795
min                       1.000000
25%                      10.000000
50%                      19.000000
75%                      31.000000
max                      66.000000
```

در این خروجی، تعداد 953 نقطه پرت شناسایی شده است که خارج از محدوده IQR قرار دارند. پس از حذف این نقاط، تعداد داده‌های باقی‌مانده 9005 است که شامل میانگین 21.82 ماه، انحراف معیار 14.47 ماه و ...

می‌باشد. نقاط پرت می‌توانند تاثیرات جدی و منفی بر مدل‌های پیش‌بینی داشته باشند و می‌توانند دقت پیش‌بینی را کاهش دهند، زیرا باعث می‌شوند مدل از مقادیر واقعی فاصله بگیرد و نتواند الگوی دقیق و درست را شبیه‌سازی کند. علاوه بر این نقاط پرت می‌توانند روی پارامترهایی مانند میانگین، انحراف معیار و ... تاثیر بگذارند و آن‌ها را از مقادیر واقعی منحرف کنند که در نتیجه خطاها افزایش پیدا می‌کند و دقت و درستی مدل ما کاهش پیدا می‌کند؛ بنابراین شناسایی و حذف این نقاط برای بهبود دقت مدل و شبیه‌سازی بهتر الگوهای واقعی داده‌ها ضروری است.

.G

در این پارت باید ویژگی cooperation_duration_months را نرمال‌سازی کنیم. همانطور که از درس به خاطر داریم از روش‌های مختلفی برای نرمال‌سازی می‌توانیم استفاده می‌کنیم. یکی از این روش‌ها MinMax است که مقادیر را در بازه 0 تا 1 می‌برد؛ همچنین روش‌های دیگری مثل Z-Score یا decimal scaling نیز داریم و در ادامه از روش MinMax استفاده کرده‌ایم. ستون cooperation_duration_months که مدت زمان همکاری را به صورت عددی نشان می‌دهد به مقیاس [0,1] تبدیل می‌شود تا مقیاس‌های داده‌ها یکسان شوند و برای مدل‌سازی آماده باشند. این کار باعث می‌شود که تفاوت‌های مقیاس در داده‌ها اثر منفی بر عملکرد مدل نداشته باشد و در نهایت نیز داده‌های نرمال‌شده در ستون جدید cooperation_duration_normalized ذخیره می‌شود و مقادیر اولیه و نرمال‌شده را در خروجی نمایش می‌دهیم:

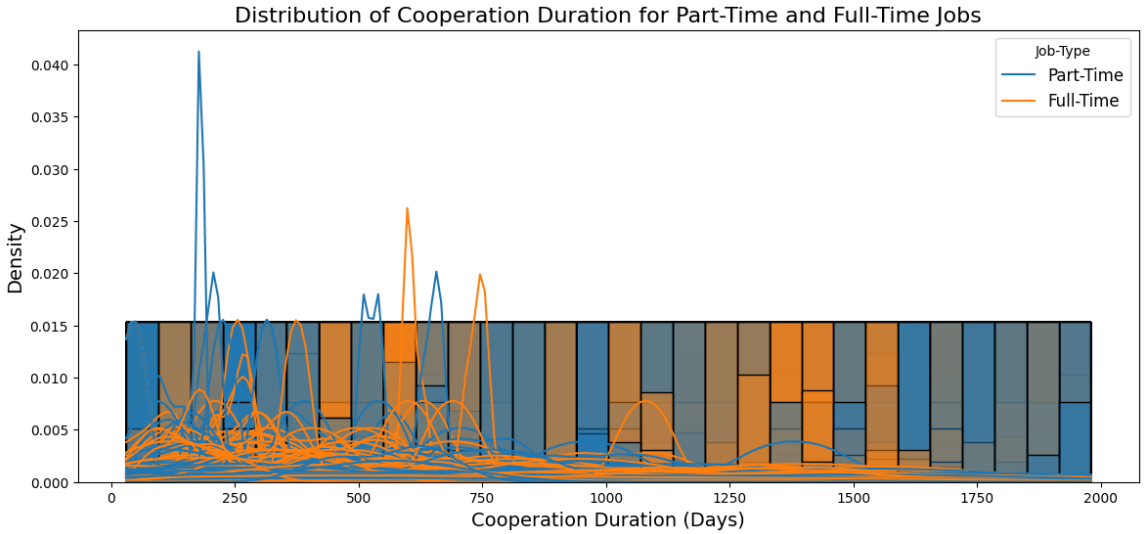
	cooperation_duration_months	cooperation_duration_normalized
0	25	0.369231
1	20	0.292308
2	21	0.307692
3	52	0.784615
6	19	0.276923

نرمال‌سازی داده‌ها می‌تواند تأثیر زیادی بر عملکرد مدل‌های یادگیری ماشین داشته باشد. زمانی که داده‌ها توزیع نرمال ندارند یا در مقیاس‌های مختلف قرار دارند، نرمال‌سازی به یکسان‌سازی مقیاس‌ها کمک می‌کند و از اثرات منفی اختلاف مقیاس‌ها جلوگیری می‌کند. این تکنیک می‌تواند به مدل کمک کند تا بهتر و سریع‌تر یاد بگیرد و همچنین به بهبود سرعت همگرایی کمک شایانی می‌کند؛ در نتیجه نرمال‌سازی می‌تواند دقت مدل را بهبود بخشد و زمان لازم برای آموزش مدل را نیز کاهش دهد.

.H

در این بخش هدف ما تحلیل مدت زمان همکاری در شغل‌های تمام‌وقت و پاره‌وقت است. ابتدا داده‌های مدت زمان همکاری را به timestamp تبدیل می‌کنیم تا به صورت دقیق‌تری آن‌ها را اندازه‌گیری کنیم. در این قسمت از تمرین ما مدت زمان همکاری را از ماه به روز تبدیل کردیم تا بتوانیم آن را به صورت دقیق‌تری تحلیل کنیم؛ سپس با استفاده از کتابخانه‌های matplotlib و seaborn نمودار هیستوگرام مربوطه را رسم کردیم تا توزیع مدت زمان همکاری را برای شغل‌های تمام‌وقت و پاره‌وقت نشان دهیم.

خروجی حاصل از این قسمت را در ادامه نمایش می‌دهیم:



در این نمودار، تفاوت‌های قابل توجهی بین مدت زمان همکاری برای شغل‌های تمام‌وقت و پاره‌وقت قابل مشاهده است. شغل‌های تمام‌وقت (رنگ نارنجی) عمدتاً در محدوده‌های زمانی طولانی‌تری متمرکز هستند که نشان می‌دهد افراد شاغل در این نوع شغل‌ها معمولاً همکاری‌های بلندمدت‌تری دارند؛ این در حالی است که شغل‌های پاره‌وقت (رنگ آبی) بیشتر در محدوده زمانی کوتاه‌تری قرار دارند، به این معنی که همکاری‌ها در این دسته بیشتر به صورت مقطعی و کوتاه‌مدت است.

این تحلیل نشان می‌دهد که مدت زمان همکاری به نوع شغل وابسته است. معمولا در شغل‌های تمام‌وقت افراد زمان طولانی‌تری را در آن شغل می‌مانند، در حالی که شغل‌های پاره‌وقت ممکن است برای دوره‌های زمانی کوتاه‌مدت‌تر به کار گرفته شوند.

I.

در ابتدا به توضیح مختصری از این دو مفهوم می‌پردازیم.

Model Preprocessing: در دنیای واقعی اصطلاحا گفته می‌شود داده‌ها کثیف هستند؛ زیرا همانطور که در درس هم به آن اشاره شد ممکن است دیتاستی که داریم شامل مقادیر گمشده، ناسازگار، نقاط پرت و ... باشد و یکی از مهم‌ترین مراحل که تاثیر به سزایی در بهبود دقت مدل دارد پیش‌پردازش داده است و جزئی از آماده‌سازی داده است.

Example #	Price	Engine Power	Family Car
1	7000	310	no
2	8000	180	no
3	14000	200	no
4	15000	280	yes
5	20000 020	250	yes
6	20000	340	no
7	21000	—	no
8	22000	300	no
9	-25000 33	260	no
10	27000	285	yes
11	29000	340 8039	no
12	30000	210	no
13	39000	260	no
14	—	245	no
15	41000	285	no

در ادامه به چند مورد از مراحل پیش‌پردازش اشاره خواهیم کرد:

- **Data Cleaning:** یکی از مشکلاتی که در دیتاست‌های دنیای واقعی ممکن است برای ما پیش بیاید، وجود داده‌های گمشده است. ساده‌ترین راه می‌تواند حذف سطرهایی که دارای مقادیر گمشده هستند می‌باشد؛ اما روش هوشمندانه‌تر این است که به جای حذف این سطرها که ممکن است باعث حذف اطلاعات مهم شود، این سطرها را با مقدار میانگین، میانه، مد یا ... مربوط به آن ستون جایگزین نماییم؛ همچنین در این مرحله اگر ناسازگاری یا نویز داشته باشیم رفع می‌کنیم. نقاط پرت یا outliers هم می‌تواند تاثیر زیادی روی مدل و دقتش بگذارد و باید این موارد را هم شناسایی و در صورت نیاز حذف کنیم.
- **Data Integration:** در این قسمت، داده‌های مختلف از منابع مختلف را به یک مجموعه داده یکپارچه تبدیل می‌کنیم و اصطلاحا یکپارچه‌سازی انجام می‌شود.
- **Encoding:** برای تبدیل داده‌های متنی به مقادیر عددی از تکنیک‌هایی مثل One-Hot Encoding یا Label Encoding استفاده می‌شود؛ مثلا برای یکی از پارت‌های همین تمرین نیاز بود که ستون location_hybrid که یک ستون حاوی مقادیر دسته‌ای بود را به مقادیر عددی تبدیل کنیم و از این روش‌ها استفاده کردیم.
- **Data Reduction:** در این مرحله برای کاهش پیچیدگی محاسباتی تعداد ویژگی‌ها یا ابعاد داده‌ها را کاهش می‌دهیم و تکنیک‌های مختلفی مثل PCA، AutoEncoder و ... است که PCA یک روش خطی

برای این مسئله است و اتوانکدر هم روشی غیرخطی است و برای داده‌های پیچیده‌تر استفاده می‌شود. همچنین با کاهش تعداد نمونه‌ها (Numerosity Reduction) می‌توانیم سرعت مدل را افزایش دهیم.

- **Scaling and Normalization:** همانطور که از درس به خاطر داریم از روش‌های مختلفی برای نرمال‌سازی استفاده می‌کنیم. یکی از این روش‌ها MinMax است که مقادیر را در بازه 0 تا 1 می‌برد؛ همچنین روش‌های دیگری مثل Z-Score یا decimal scaling نیز داریم و در پیش‌پردازش میتوانیم از آنها بهره ببریم.

Feature Engineering: یکی دیگر از موارد مهم در یادگیری ماشین و بطور کلی هرجا که با داده سر و کار داریم مهندسی ویژگی است. مهندسی ویژگی به فرایند ایجاد، ترکیب یا استخراج ویژگی های مهم از داده های خام برای افزایش بهبود عملکرد مدل گفته می‌شود و به عنوان مثال می‌توان با ترکیب چند ویژگی یک ویژگی جدید ایجاد کرد. مثلا ما یک فروشگاه آنلاین داریم و می‌خواهیم بررسی کنیم که یک مشتری یک محصول مورد نظر را خریداری می‌کند یا خیر و یکسری ویژگی هم داریم، بعد از مدتی پی می‌بریم که مثلا علاوه بر فیچرهایی که داریم تعداد خریدهای قبلی و سن فرد یک ارتباطی با هم دارند و می‌تواند برای بررسی میزان فروش محصول x یک ویژگی مفید باشد؛ پس این ویژگی رو می‌توانیم ایجاد کنیم و یا در یکی از جلسات که در رابطه با استفاده از دست خط انسان بود، به کمک مهندسی ویژگی برای تشخیص و تمایز هر داده یکسری فیچر ایجاد کردیم:

همانطور که توضیح دادیم مهندسی ویژگی برای بهبود و یافتن اطلاعات بیشتر کاربرد دارد و به ساخت ویژگی های جدید و یا اصلاح برخی ویژگی می پردازد و کمک می‌کند تا مدل بتواند نتیجه‌های دقیق‌تری بگیرد اما پیش‌پردازش داده‌ها جزئی از آماده‌سازی داده‌ها است و اهمیت قابل توجهی روی دقت مدل دارد. معمولا اول پیش‌پردازش داده‌ها انجام می‌شود و بعد از آن از مهندسی ویژگی انجام استفاده می‌کنیم؛ علی‌رغم تفاوت‌های مهندسی ویژگی و پیش‌پردازش داده، هر دو برای بهبود عملکرد مدل‌های ما اهمیت دارند و باعث بهتر و بهینه‌تر شدن مدل ما خواهند شد.

«. . . بهمن‌ماه ۱۴۰۳ . . .»