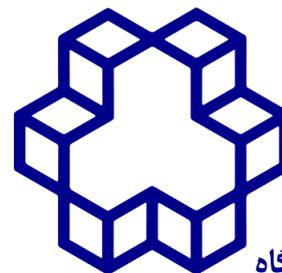


بسم الله الرحمن الرحيم



دانشگاه  
خواجہ نصیرالدین طوسی  
K. N. Toosi University  
of Technology

پاسخ مینی پروژه سوم یادگیری ماشین

Google Colab Parts 1.1 , 1.2  
Google Colab Parts 1.3 , 2  
GitHub

نگارش: علی شعبانیپور مقدم - هدیه شوشیان

شماره دانشجویی: ۴۰۲۰۷۳۰۴-۴۰۳۰۸۰۵۴

استاد درس: دکتر مهدی علیاری شوره دلی

## فهرست مطالب

۳	۱ پرسش اول
۳	۱.۱
۳	۲.۱
۳	۱.۲.۱
۴	۲.۲.۱
۷	۳.۲.۱
۸	۴.۲.۱
۹	۵.۲.۱
۹	۶.۲.۱
۱۰	۷.۲.۱
۱۶	۸.۲.۱
۱۸	۹.۲.۱
۲۲	۱۰.۲.۱
۲۲	۱۱.۲.۱
۲۲	۱۲.۲.۱
۳۲	۱۳.۲.۱
۳۳	۱۴.۲.۱
۳۴	۱۵.۲.۱
۳۵	۳.۱
۳۵	۱.۳.۱
۳۵	۲.۳.۱
۳۵	۳.۳.۱
۳۶	۴.۳.۱
۳۶	۵.۳.۱
۳۹	۶.۳.۱
۴۱	۷.۳.۱
۴۳	۲ پرسش دوم
۴۳	۱.۲
۴۰	۲.۲



## ۱ پرسش اول

### ۱.۱

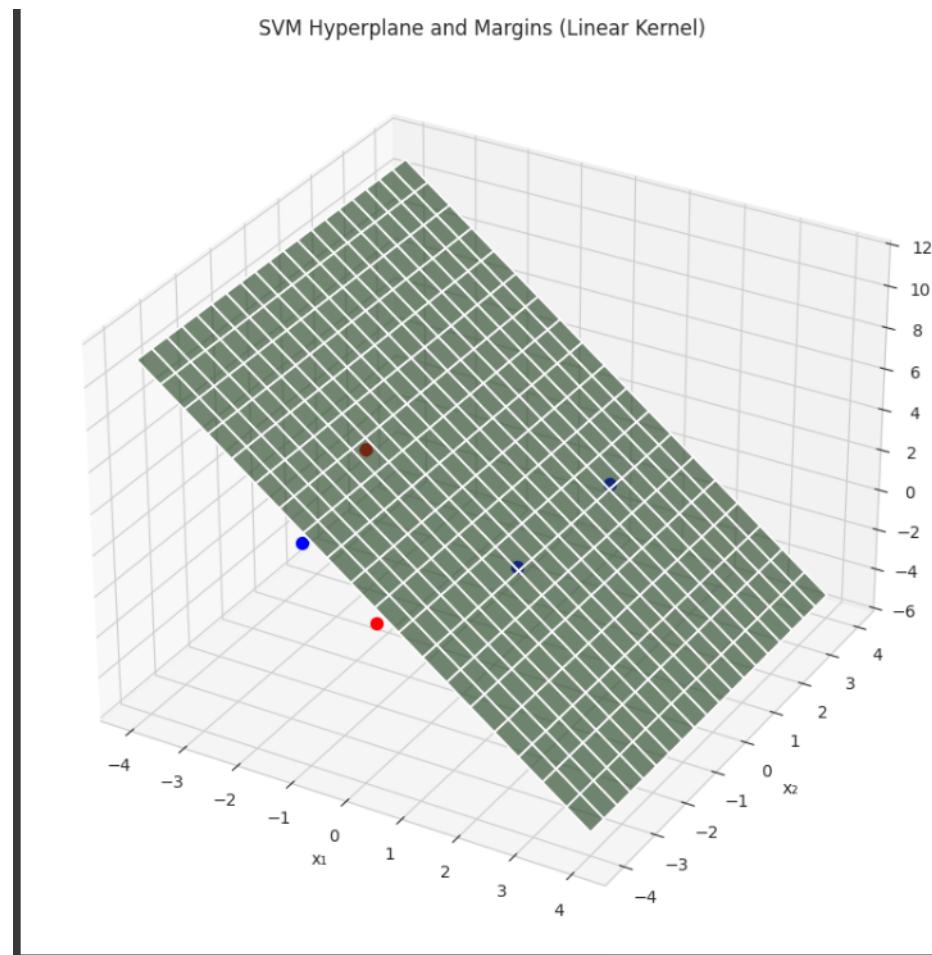
پس از اضافه کردن کتابخانه های مربوطه و اجرای کد شکل صفحه خروجی و پارامترها و معادله صفحه بصورت زیر است:

```

Indices of Support Vectors (1-based): [1 2 3 4 5]
Alphas: [5.05770117e+19 3.03462070e+20 4.55193106e+20 1.01154023e+20
          3.03462070e+20]
Weight vector w: [-262144.          0. -131072.]
Bias b: 393215.0

Hyperplane Equation:
-262144.00·x1 + 0.00·x2 - 131072.00·x3 + 393215.00 = 0

```



### ۲.۱

#### ۱.۲.۱

چهار ویژگی اول تاریخ تشکیل دیتا است که به ترتیب شماره ستون سال ماه و روز و ساعت هستند. ویژگی های دیگر میزان غلظت ذرات معلق با واحد متر مکعب هستند. نقطه شبیم ذمای نشان میدهد که در آن هوا به حالت اشباع میرسد و بخار اب نقطیر میکند و شاخص رطوبت است. میتواند باعث نزدیک نهگداشت رطوبت در اطراف زمین گردد. دما با واحد درجه سانتی گراد. فشار اتمسفر با واحد هکتوپاسکال. جهت باد ترکیبی بدون واحد نماینگر جهت است. سرعت باد در هر لحظه با واحد متر بر ثانیه.  $ls$  برای معیاری از زمان باریدن برف بر حسب ساعت.  $lr$  میزان زمان بارش باران بر حسب ساعت

ویژگی های مرتبط با میزان غلظت ذرات با قطر ۰.۵ میکرومتر عبارت است از: دما بر روی واکنس ها شیمیایی و پراکنده‌گی الایده ها اثر می‌گذارد. نقطه شبنم معیار رطوبت و درنتیجه نزدیک نگه داشتن ذرات به سطح زمین اثر دارد. فشار باعث سکون هوا و انباسته‌اینده ها می‌گردد. جهت ورودی و خروجی الودگی را جهت باد تعیین می‌کند. سرعت باد روی پراکنده‌گی الایnde ها اثر گذار است. میزان ساعت بارش برف و باران در شست و شوی ذرات معلق اثر دارد. الودگی ها شهری و عواملی که به ساعت روزمره اثر دارد نیز در غلظت الایnde ها موثر است.

از جمله موارد دیگه ای که به عنوان دیتا میتواند برای ما مفید باشد میتوان به جزیات دقیق تر جهت باد و وجود اطلاعات ماهواره ای، مهندسی ویژگی ها یعنی برای مثال بررسی نقش انسانی ویژگی های دیگر اب و هوایی مانند رطوبت نسبی و وارونگی دما اشاره کرد. همچنین عوامل مهمی مانند ارتفاع میزان تابش نور خورشید نیز بسیار مهم و موثر است. تمامی موارد ذکر شده دارای ارتباط ریاضی نسبت به همدیگر میباشند که میتوان در کد پایتون خود انها را به عنوان عوامل و ویژگی های دیگر اضافه کرده و اثر ان ها را مورد بررسی برای بدست اوردن نتایج بهتر استفاده کنیم.

## ۲.۲.۱

پس از بارگذاری دیتا بصورت خواسته شده داده خواهیم داشت:

File 'PRSA_data_2010.1.1-2014.12.31.csv' already exists locally.													
No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	
0	1	2010	1	1	0	NaN	-21	-11.0	1021.0	NW	1.79	0	0
1	2	2010	1	1	1	NaN	-21	-12.0	1020.0	NW	4.92	0	0
2	3	2010	1	1	2	NaN	-21	-11.0	1019.0	NW	6.71	0	0
3	4	2010	1	1	3	NaN	-21	-14.0	1019.0	NW	9.84	0	0
4	5	2010	1	1	4	NaN	-20	-12.0	1018.0	NW	12.97	0	0

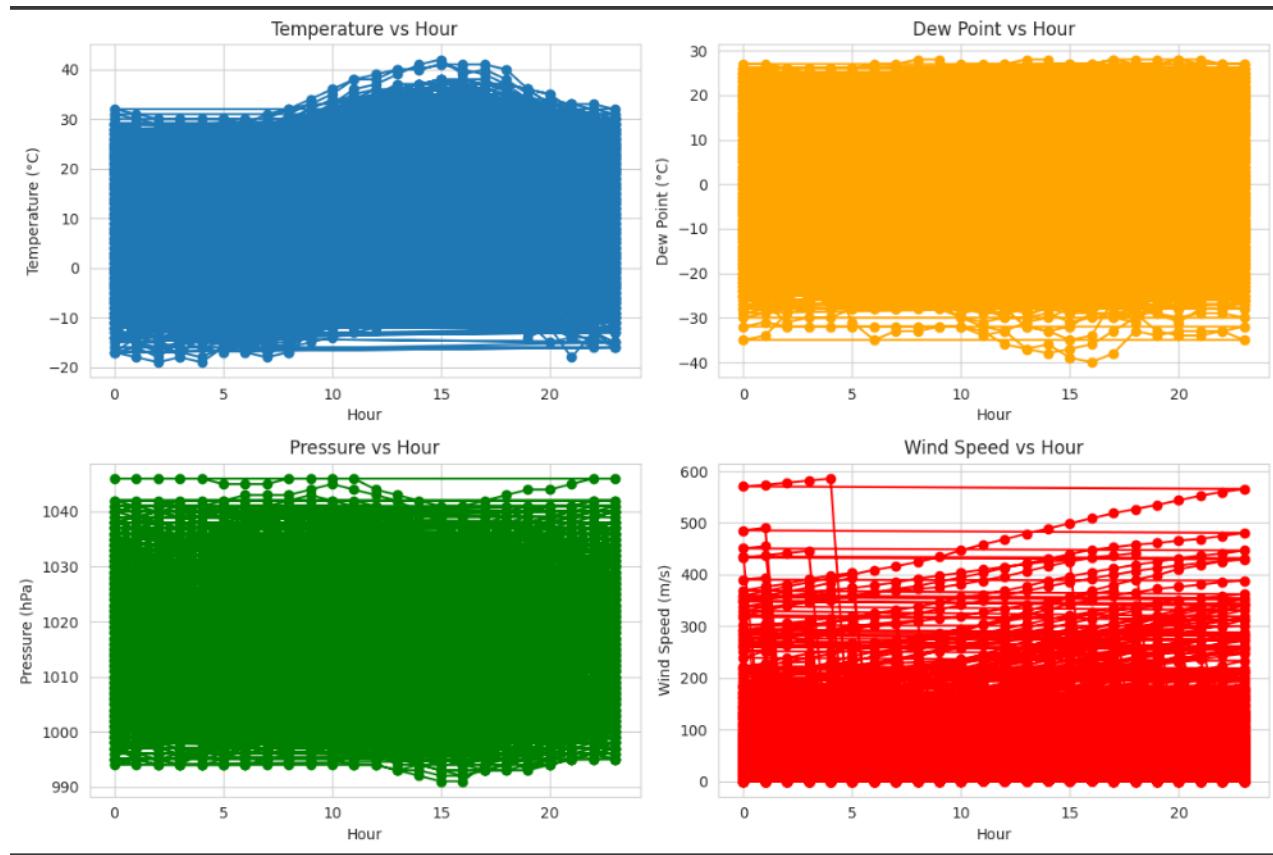
Features sample:												
No	year	month	day	hour	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	
0	1	2010	1	1	0	-21	-11.0	1021.0	NW	1.79	0	0
1	2	2010	1	1	1	-21	-12.0	1020.0	NW	4.92	0	0
2	3	2010	1	1	2	-21	-11.0	1019.0	NW	6.71	0	0
3	4	2010	1	1	3	-21	-14.0	1019.0	NW	9.84	0	0
4	5	2010	1	1	4	-20	-12.0	1018.0	NW	12.97	0	0

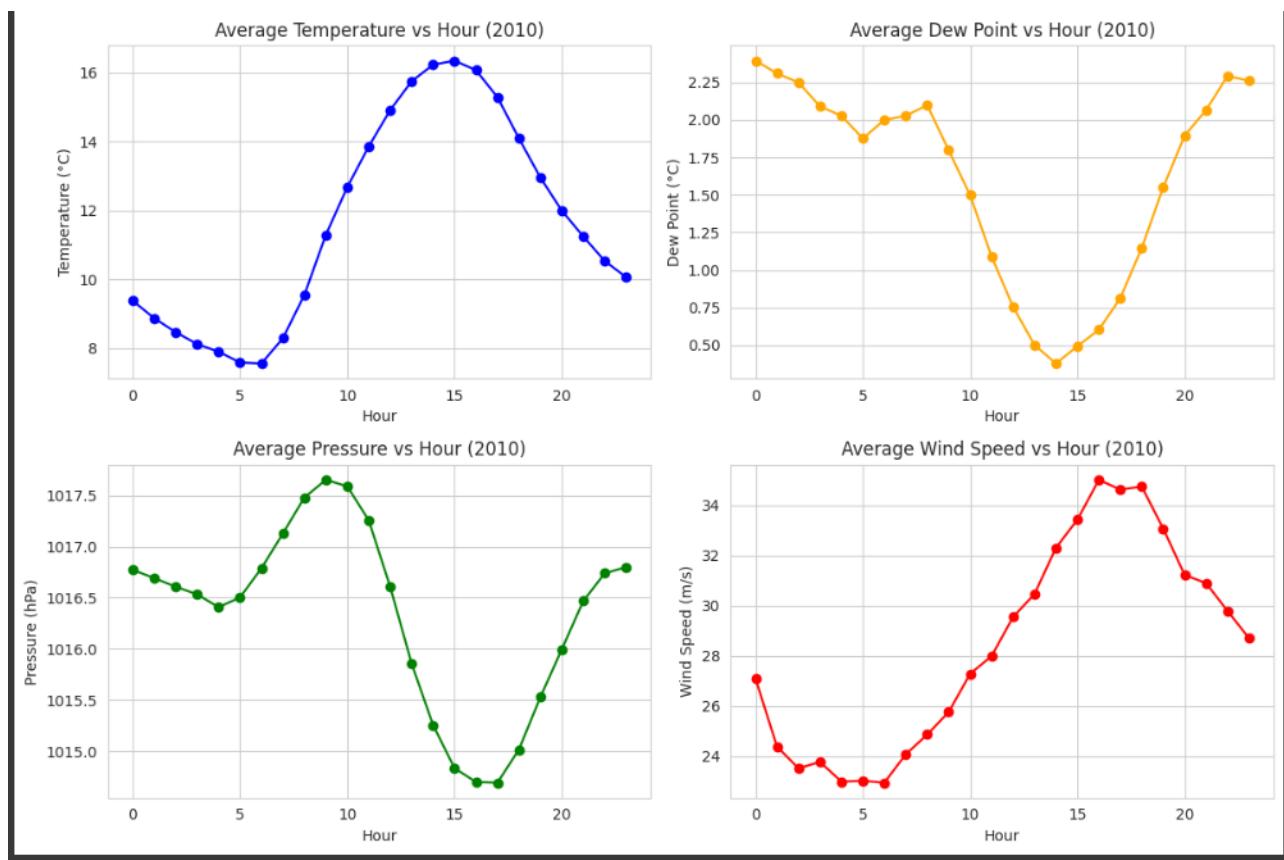
Target sample:												
0	NaN											
1	NaN											
2	NaN											
3	NaN											
4	NaN											

Name: pm2.5, dtype: float64

همانطور که مطابق تصویر مشاهده میکنیم داده ها از سال ۲۰۱۰ تا ۲۰۱۴ جمع اوری شده اند و بارش باران و برف نداشته ایم. جهت باد ثابت بوده و دما پایین فشار بالا و سرعت باد نیز پایین بوده است و این باعث تجمع و افزایش غلظت الایnde ها می‌گردد. میزان تغییرات چهار ویژگی در نمودار زیر نمایش داده شده است.



در یک سال و بصورت میانگین مقادیر این ویژگی‌ها بصورت زیر خواهد بود:



مطابق شکل فوق روند دما از هشت درجه برای نیمه شب تا هفت درجه میرسد و پس از ساعت دو بعد ظهر تا شانزده درجه افزایش میابد و مجدداً رو به کاهش میرود. همینطور برای روند نقطه شبنم میتوان متوجه شد از الگوی مشابه دما بهره میبرد اما رطوبت نسبی باعث تغییراتی رو ان نیز میگردد. برای فشار عموماً شاهد تغییرات شدید نیستم مگر شرایط جوی بر روی آن اثر بگذارد. سرعت باد اما تغییراتی بر حسب گرمایش زمین و عوامل اب و هوایی دیگر دارد. هر سطح نمایانگر یک مشاهده در روز و ساعت خاص و هر ستون یک شاخصه مثلاً ویژگی های اب هوایی است.

همانطور که بررسی شد دما فشار و نقطه شبنم دارای مقادیر پیوسته و منظمی و باباتی هستند. در اوایل هر ماه مشاهده میشود غلظت الاینده ها ثبت نگردیده است در ادامه ای حل سوال باید با پیش بروز اینها مناسب آنها را حذف کنیم یا میانگین گیری یا پیش بینی انجام دهیم و مقادیری را برای آنها ثبت کنیم. تغییرات جهت باد را باید بصورت کیفی کد گذاری نماییم. روش *trendanalysis* میتواند در این ساختار داده زمانی ما را مساعدت دهد. ساختار داده اجازه میدهد از ویژگیهای ترکیبی مانند اختلاف دما یا رطوبت نسبی بهره ببریم.

تراکم داده ها نشان میدهد در حدود ۴۳۰۰۰ داده و سطر داریم. پس میتوان از مدلی ها *lstm*, *arima* بهره برد.



```

File 'PRSA_data_2010.1.1-2014.12.31.csv' already exists locally.
== Dataset Metadata (extracted) ==
name: Beijing PM2.5
num_instances: 43824
num_features: 12
target_col: ['pm2.5']
has_missing_values: yes
missing_values_count: 2067
feature_types: ['int64', 'float64', 'object']
date_range: {'year': 2010, 'month': 1, 'day': 1} to {'year': 2014, 'month': 12, 'day': 31}

== Variables Summary ==
    name      role   type missing_values
0      No       ID    Integer      no
1    year     Feature  Integer      no
2   month    Feature  Integer      no
3     day     Feature  Integer      no
4    hour    Feature  Integer      no
5  pm2.5    Target    Real      yes
6   DEMP    Feature  Integer      no
7   TEMP    Feature    Real      no
8   PRES    Feature    Real      no
9   cbwd  Feature Categorical      no
10   Iws    Feature    Real      no
11   Is     Feature  Integer      no
12   Ir     Feature  Integer      no

== Sample Data with datetime ==
        datetime  pm2.5
0 2010-01-01 00:00:00    NaN
1 2010-01-01 01:00:00    NaN
2 2010-01-01 02:00:00    NaN
3 2010-01-01 03:00:00    NaN
4 2010-01-01 04:00:00    NaN

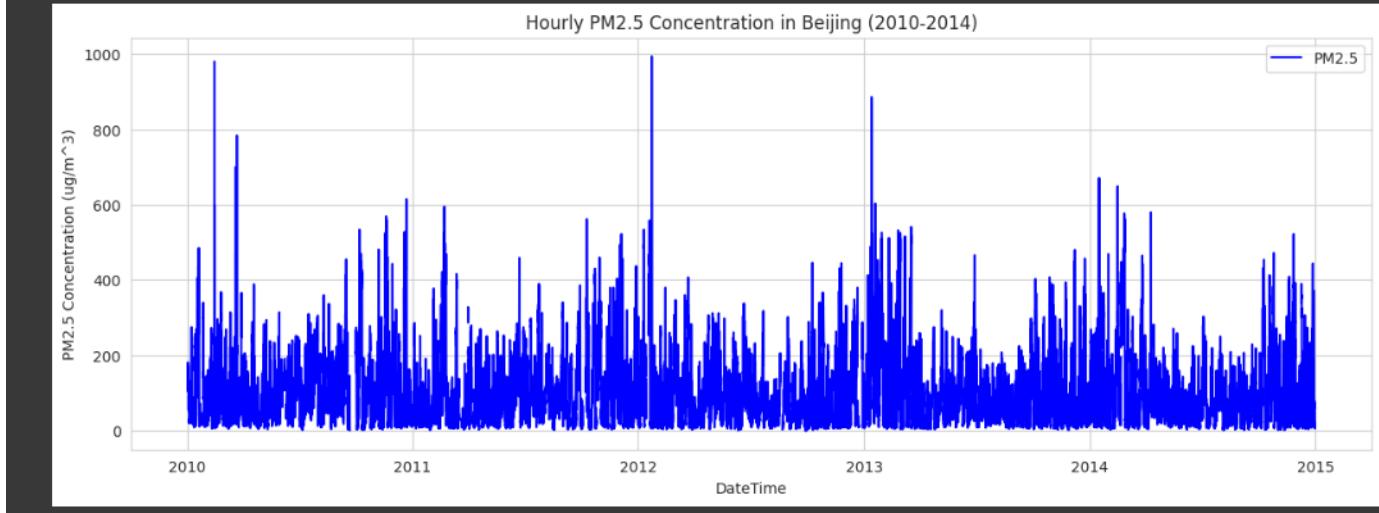
```

میزان تغییرات غلظت الاینده ها در سال های ۲۰۱۰ تا ۲۰۱۴ را مشاهده میکنیم:

```

File 'PRSA_data_2010.1.1-2014.12.31.csv' already exists locally.
        datetime  pm2.5
0 2010-01-01 00:00:00    NaN
1 2010-01-01 01:00:00    NaN
2 2010-01-01 02:00:00    NaN
3 2010-01-01 03:00:00    NaN
4 2010-01-01 04:00:00    NaN

```



بیشترین افزایش ایالنده برای سال ۲۰۱۲ میباشد.

### ۳.۲.۱

فقط ستون غلظت دارای گمشدگی دیتا میباشد و این سطر خا را حذف مینماییم. برای ۲۴ سطر اول حذف و برای مابقی نزدیک ترین مقدار را جایگزین مینماییم.



八

```
↓ نتایج مقادیر گفته شده در هر سوتون  
No          0  
year        0  
month       0  
day         0  
hour        0  
pm2.5       2067  
DEWP        0  
TEMP        0  
PRES        0  
cbwd        0  
Iws         0  
Is          0  
Ir          0  
datetime    0  
dtype: int64
```

تنهای ۴ درصد از کل داده ها باید محوذف گردند.

```

No          0.000000
year        0.000000
month       0.000000
day         0.000000
hour        0.000000
pm2.5       4.716594
DEWP        0.000000
TEMP        0.000000
PRES        0.000000
cbwd        0.000000
Iws         0.000000
Is          0.000000
Ir          0.000000
datetime    0.000000
dtype: float64

```

۴۰۲۱

داده های مربوط به *cbwd* از نوع *categorical* است و با *labelencoding* این داده ها را به عدد تبدیل مینماییم. به شکل ها زیر توجه کنیم:

```
['SE' 'SE' 'SE' ... 'NW' 'NW' 'NW']
```

[2	2	2	...	1	1	1]	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	\
0				25	2010		1		2	0	129.0	-16	-4.0	1020.0	SE	1.79		
1				26	2010		1		2	1	148.0	-15	-4.0	1020.0	SE	2.68		
2				27	2010		1		2	2	159.0	-11	-5.0	1021.0	SE	3.57		
3				28	2010		1		2	3	181.0	-7	-5.0	1022.0	SE	5.36		
4				29	2010		1		2	4	138.0	-7	-5.0	1022.0	SE	6.25		
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...		
43795				43820	2014		12		31	19	8.0	-23	-2.0	1034.0	NN	231.97		
43796				43821	2014		12		31	20	10.0	-22	-3.0	1034.0	NN	237.78		
43797				43822	2014		12		31	21	10.0	-22	-3.0	1034.0	NN	242.70		
43798				43823	2014		12		31	22	8.0	-22	-4.0	1034.0	NN	246.72		
43799				43824	2014		12		31	23	12.0	-21	-3.0	1034.0	NN	249.85		
Is	Ir						datetime							cbwd_numerical				
0	0	0		2010-01-02	00:00:00									2				
1	0	0		2010-01-02	01:00:00									2				
2	0	0		2010-01-02	02:00:00									2				
3	1	0		2010-01-02	03:00:00									2				
4	2	0		2010-01-02	04:00:00									2				
...	..	..					...							...				
43795	0	0		2014-12-31	19:00:00									1				
43796	0	0		2014-12-31	20:00:00									1				
43797	0	0		2014-12-31	21:00:00									1				
43798	0	0		2014-12-31	22:00:00									1				
43799	0	0		2014-12-31	23:00:00									1				

تصویر فوق دسته بندی را انجام میدهیم.

#### ۵.۲.۱

وجود داده ها پر باعث نتیجه گیری اشتباہ میشود بخصوص اگر به ان بسیار دقت کنیم و وزن دهیم. مثلاً می تواند باعث اورفیت شدن در رگرسیون گردد یا در شبکه عصبی باعث واگرایی شود. ستون ها را با روش  $IQR$  بدست می اوریم و با روش *capping* ادامه میدهیم. خروجی بصورت زیر است بطوریکه داده های بیرون بازه قبول قابل قبول محدود میشوند و تغییر میابند.

```
pm2.5: no. outliers = 1857
TEMP: no. outliers = 0
DEWP: no. outliers = 0
Iws: no. outliers = 5101
```

قبل از اعمال الگوریتم داده های پر بصورت زیر هستند و با روش چارک و تعیین معیاری بر اساس ان داریم:

تقطیع بررسی ناچاهاتی بیت:	
pm2.5:	ستون تعداد بیت ها: 1857 برصد بیت ها: 4.24% محدوده مجاز: (296.50, 131.50)- نمونه بیت ها: [ .407 .349 .303 .313 .317 ]
TEMP:	ستون تعداد بیت ها: 0 برصد بیت ها: 0.00% محدوده مجاز: (-54.50, 29.50)- نمونه بیت ها: بیت وجود ندارد
DEWP:	ستون تعداد بیت ها: 0 برصد بیت ها: 0.00% محدوده مجاز: (-52.50, 47.50)- نمونه بیت ها: بیت وجود ندارد
Iws:	ستون تعداد بیت ها: 5101 برصد بیت ها: 11.65% محدوده مجاز: (52.09, 28.39)- نمونه بیت ها: [ 65.71 61.69 58.56 55.43 52.3 ]

با اعمال الگوریتم توضیح داده شده داده پرتری نخواهیم داشت:

```
تعداد نقطه بیت = 0
TEMP: 0 = تعداد نقطه بیت
DEWP: 0 = تعداد نقطه بیت
Iws: 0 = تعداد نقطه بیت
```

#### ۶.۲.۱

با استفاده از جدول استانداردهای کیفیتی هوا زیر سطوح مختلف سلامتی را تعیین کردیم:

AQI Category	Index Values	Revised Breakpoints ( $\mu\text{g}/\text{m}^3$ , 24-hour average)
Good	0 - 50	0.0 - 12.0
Moderate	51 - 100	12.1 - 35.4
Unhealthy for Sensitive Groups	101 - 150	35.5 - 55.4
Unhealthy	151 - 200	55.5 - 150.4
Very Unhealthy	201 - 300	150.5 - 250.4
	301 - 400	250.5 - 350.4
Hazardous	401 - 500	350.5 - 500



ستونی جدید به دیتا است اضافه می‌نماییم و نتیجه بصورت زیر خواهد بود:

No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	datetime	AQI_Category	
24	25	2010	1	2	0	129.0	-16	-4.0	1020.0	SE	1.79	0	0	2010-01-02 00:00:00	Unhealthy
25	26	2010	1	2	1	148.0	-15	-4.0	1020.0	SE	2.68	0	0	2010-01-02 01:00:00	Unhealthy
26	27	2010	1	2	2	159.0	-11	-5.0	1021.0	SE	3.57	0	0	2010-01-02 02:00:00	Very Unhealthy
27	28	2010	1	2	3	181.0	-7	-5.0	1022.0	SE	5.36	1	0	2010-01-02 03:00:00	Very Unhealthy
28	29	2010	1	2	4	138.0	-7	-5.0	1022.0	SE	6.25	2	0	2010-01-02 04:00:00	Unhealthy

Next steps: [Generate code with df\\_with\\_AQI](#) [View recommended plots](#) [New interactive sheet](#)

۷.۲.۱

### ویزگی lag

زمانی که سری های زمانی با وابستگی همراه هستند و مقدار فعلی یا اینده تحت تاثیر گذشته‌ی خود است مدل‌های یادگیری ماشین بطور مستقیم و کار نمی‌کنند از این فیچر لک بهره می‌بریم. برای اینکار ابتدا داده‌ها را شیفت داده و در یک ستون جدید میریزیم و برای داده‌های ۲۴ ساعت قبل ۲۴ سطر اول خالی را حذف می‌کنیم همین کار را برای دو ساعت نیز انجام میدهیم. یعنی از شکل زیر

No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	datetime	AQI_Category	pm2.5_lag_2h	pm2.5_lag_24h	
24	25	2010	1	2	0	129.0	-16	-4.0	1020.0	SE	1.79	0	0	2010-01-02 00:00:00	Unhealthy	NaN	NaN
25	26	2010	1	2	1	148.0	-15	-4.0	1020.0	SE	2.68	0	0	2010-01-02 01:00:00	Unhealthy	NaN	NaN
26	27	2010	1	2	2	159.0	-11	-5.0	1021.0	SE	3.57	0	0	2010-01-02 02:00:00	Very Unhealthy	129.0	NaN
27	28	2010	1	2	3	181.0	-7	-5.0	1022.0	SE	5.36	1	0	2010-01-02 03:00:00	Very Unhealthy	148.0	NaN
28	29	2010	1	2	4	138.0	-7	-5.0	1022.0	SE	6.25	2	0	2010-01-02 04:00:00	Unhealthy	159.0	NaN

Next steps: [Generate code with df\\_with\\_lags](#) [View recommended plots](#) [New interactive sheet](#)

به این شکل از دادگان میرسیم:

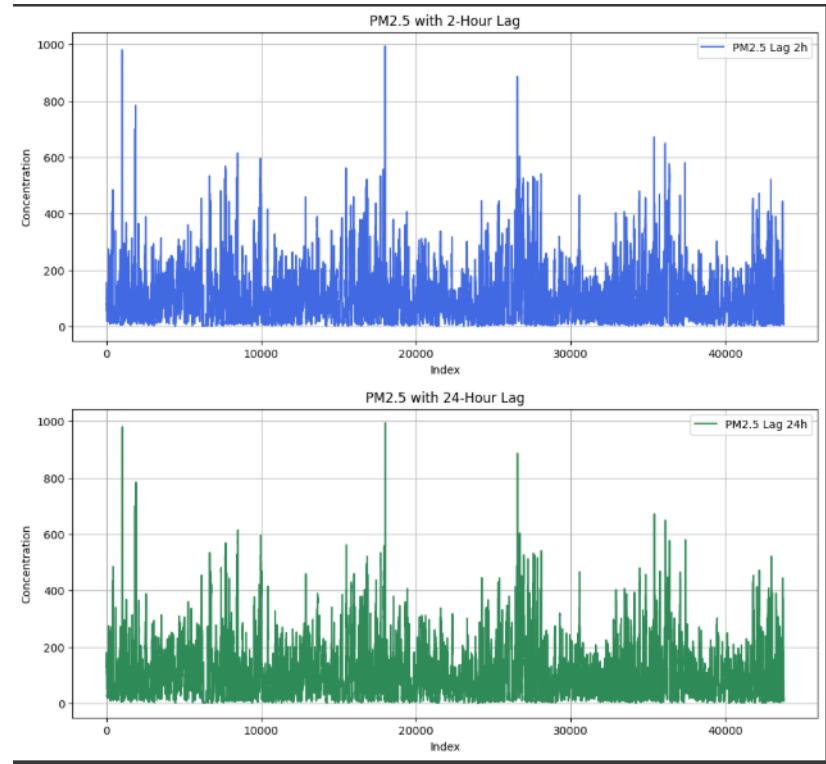
No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	datetime	AQI_Category	pm2.5_lag_2h	pm2.5_lag_24h	
0	49	2010	1	3	0	90.0	-7	-6.0	1027.0	SE	58.56	4	0	2010-01-03 00:00:00	Unhealthy	156.0	129.0
1	50	2010	1	3	1	63.0	-8	-6.0	1026.0	SE	61.69	5	0	2010-01-03 01:00:00	Unhealthy	126.0	148.0
2	51	2010	1	3	2	65.0	-8	-7.0	1026.0	SE	65.71	6	0	2010-01-03 02:00:00	Unhealthy	90.0	159.0
3	52	2010	1	3	3	55.0	-8	-7.0	1025.0	SE	68.84	7	0	2010-01-03 03:00:00	Unhealthy for Sensitive Groups	63.0	181.0
4	53	2010	1	3	4	65.0	-8	-7.0	1024.0	SE	72.86	8	0	2010-01-03 04:00:00	Unhealthy	65.0	138.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
43771	43820	2014	12	31	19	8.0	-23	-2.0	1034.0	NW	231.97	0	0	2014-12-31 19:00:00	Good	9.0	35.0
43772	43821	2014	12	31	20	10.0	-22	-3.0	1034.0	NW	237.78	0	0	2014-12-31 20:00:00	Good	10.0	26.0
43773	43822	2014	12	31	21	10.0	-22	-3.0	1034.0	NW	242.70	0	0	2014-12-31 21:00:00	Good	8.0	20.0
43774	43823	2014	12	31	22	8.0	-22	-4.0	1034.0	NW	246.72	0	0	2014-12-31 22:00:00	Good	10.0	8.0
43775	43824	2014	12	31	23	12.0	-21	-3.0	1034.0	NW	249.85	0	0	2014-12-31 23:00:00	Good	10.0	16.0

43776 rows x 17 columns

ماتریس هم بستگی نتیج ملموس تری را برای تحلیل به ما ارائه خواهد داد اما فعلاً بنظر می‌اید در زمان‌های یکسان مقادیر نزدیک بهم اختیار شده‌اند. و داده‌های دو ساعت قبل به داده‌های فعلی هم بستگی شدیدی دارند و با فاطله گرفتن داده‌ها از یکدیگر هم بخشگی انها کاهش می‌یابد. در حقیقت بجای دو ساعت می‌توان از سه ساعت بهره بردن نکته مهمن در کوتاه مدت بودن آن و برای ۲۴ ساعت میان مدت بودن است. زمانی که از کوتاه مدت برای پیش‌بینی بهره میریم الگوهای ساعتی و را می‌باییم اما وقتی از میان مدت بهره میریم الگوی روزانه را بررسی می‌کنیم همچنین می‌توان از الگوی بلندت

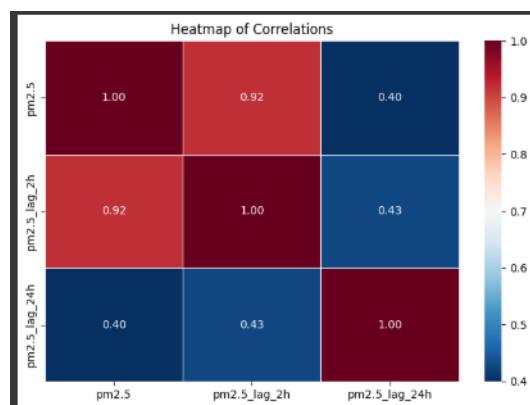


مدت مثلا هفته بهره برد. به همین ترتیب با افزایش این زمان میتوان حتی روندهای ماهانه یا فصلی را مورد بررسی قرار داد.  
خروجی برای دو و ۲۴ ساعت بصورت زیر است:



با بررسی الگوی دو ساعت قبل مشاهده میکنیم در سطحهای ۲۸ تا ۲۸ شاهد تغییرات تدریجی و نوسانات کوچک هستیم و این همان همبستگی بالا را نشان میدهد. پس در کوتاه مدت دارای تغییرات سریع و ساعتی است. مناسب برای پیش بینی حداکثر تا دو ساعت است.  
 از طرف دیگر برای تغییر میان مدت یا ۲۴ ساعته شاهد الگوی تکراری روزانه هستیم که میتواند علت ان شرایط جوی یا انسانی باشد. این ویژگی برای برنامه ریزی روزانه مطمئن تر است.

ماتریس همبستگی به شکل زیر است:



مشاهده می کنیم این ماتریس نیز میزان همبستگی ساعتی را بیشتر میداند.

*rollingstatics* ویژگی

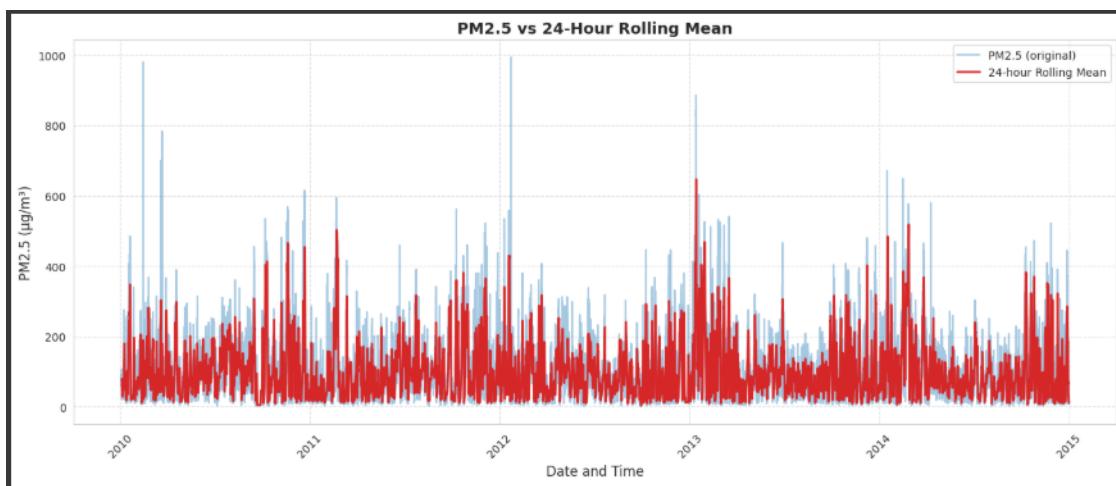
برای تحلیل سری های زمانی شامل محاسباتی اماری از قبیل میانگین واریانس انحراف معیار و سایر مقادیر روی یک باز یا پنجه محرك از داده هاست. برای *smooth* کردن سری زمانی بررسی ترندهای محلی و ایجاد ویژگی هایی که عملکرد پیش بینی را بهبود میبخشد کاربرد دارد. کمک میکند روند ها الگوهای نوسانات داده ها را در طول زمان بصورت پویا بدست اوریم.

از شرایط استفاده از این ویژگی میتوان به مواردی اشاره کرد از قبیل: وقتی دارای نویز و نوسانات زیاد سهتیمو روند را بطور نرم میخواهیم بررسی کنیم. مثلاً از میانگین برای نوسانات ساعتی بالا بهره ببریم. یا زمانی که الگوهای تکراری روزانه یا هفتگی را شناسایی و پیش بینی میکنیم مانند میانگین الودگی در یک روز. یا در مقایسه تغییرات ناگهانی و ناپایداری که در این حالت از واریانس یا انحراف معیار برای شناسایی نوسان های غیرعادی مثلاً در ساعات خاص استفاده میشود. میتوان برای داده های پرت از میانگین بهره ببریم. در بهبود و بهینه سازی مدل پیش بینی زمانی که ویژگی جدید به عنوان وروردی به مدل داده میشود.

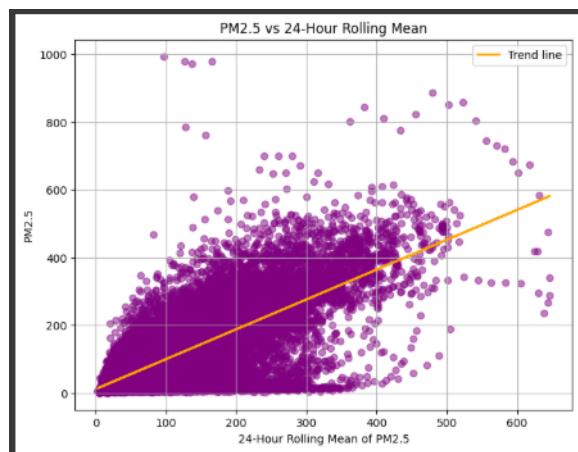
در مجموع موارد فوق دارای مزیت های بهبود در ک روند هاست، تشخیص نوسانات و ناپایداری ها، پیش بینی دقیق تر، تصمیم گیری بهتر و...

	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Tws	Is	Ir	datetime	AQI_Category	pm2.5_lag_2h	pm2.5_lag_24h	pm2.5_roll_mean_24h
0	49	2010	1	3	0	90.0	-7	-6.0	1027.0	SE	58.56	4	0	2010-01-03 00:00:00	Unhealthy	156.0	129.0	NaN
1	50	2010	1	3	1	63.0	-8	-6.0	1026.0	SE	61.69	5	0	2010-01-03 01:00:00	Unhealthy	126.0	148.0	NaN
2	51	2010	1	3	2	65.0	-8	-7.0	1026.0	SE	65.71	6	0	2010-01-03 02:00:00	Unhealthy	90.0	159.0	NaN
3	52	2010	1	3	3	55.0	-8	-7.0	1025.0	SE	68.84	7	0	2010-01-03 03:00:00	Unhealthy for Sensitive Groups	63.0	181.0	NaN
4	53	2010	1	3	4	65.0	-8	-7.0	1024.0	SE	72.86	8	0	2010-01-03 04:00:00	Unhealthy	65.0	138.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
43771	43820	2014	12	31	19	8.0	-23	-2.0	1034.0	NW	231.97	0	0	2014-12-31 19:00:00	Good	9.0	35.0	11.291667
43772	43821	2014	12	31	20	10.0	-22	-3.0	1034.0	NW	237.78	0	0	2014-12-31 20:00:00	Good	10.0	26.0	10.625000
43773	43822	2014	12	31	21	10.0	-22	-3.0	1034.0	NW	242.70	0	0	2014-12-31 21:00:00	Good	8.0	20.0	10.208333
43774	43823	2014	12	31	22	8.0	-22	-4.0	1034.0	NW	246.72	0	0	2014-12-31 22:00:00	Good	10.0	8.0	10.208333
43775	43824	2014	12	31	23	12.0	-21	-3.0	1034.0	NW	249.85	0	0	2014-12-31 23:00:00	Good	10.0	16.0	10.041667
43776 rows × 18 columns																		

برای ۲۴ سطر اول که مقدار *nan* است علت آن این است که پنجه زمانی ۲۴ ساعته است پس داده های زیر ۲۴ است داده ندارند. بدیهی است در این حالت میانگین ثابت میماند. سه روش برای مقادیر گمشده وجود دارد یکی حذف کامل این سطر هاست دیگری پر کردن با میانه و مد است و اخیر پر کردن با قبلی و بعدی است. که پر کردن با بلی بعدی برای سری ها زمانی که به یکدیگر وابسته هستند بسیار مهم است. البته روش پر کردن با داده های قبلی هم میتواند خطرناک باشد و اطلاعات مصنوعی وارد کند. درنتیجه رشد مد هم انتخاب خوبی است. دقت داریم که چون لگ داریم ۲۴ سطر اول خالی است و حذف باید بشود.



شکل فوق پنجه ۲۴ ساعته را برای حالت *rolling* و نبود ان ترسیم میکند. با دقت کردن به شکل زیر متوجه میشویم هرچه داده ها از لحظه زمانی از یکدیگر فاصله میگیرند نرا کم داده ها از نیم ساز صفحه کمتر میشود. و داده ها در روز اول بسیار نزدیک هستند.

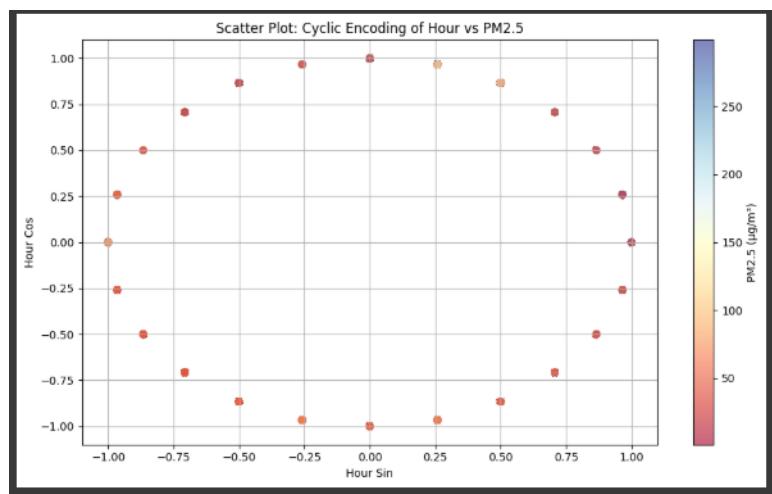


### انکود کردن ویژگی های تناوبی

تبدیل ویژگی های دوره ای تکرار شونده به شکل عددی که چرخش وار است. این ویژگی های تناوبی را به عدد تبدیل می کنیم تا درک ان برای ماشین اسان تر گردد یک نمایش دایره ای برای می تواند با استفاده از سینوس و کسینوس ما را برای انکور کردن ساعات کمک بخشد. برای مثال کسینوس نیمه شب صفر و سینوس ان یک است. بدیهی است باید ساعات را به درجه تبدیل نماییم. به شکل های زیر دقت شود:

No	year	month	day	hour	pm2.5	Dtemp	Temp	Pres	Cloud	Iws	Is	Tr	datetime	AQI_Catagory	pm2.5_lag_2h	pm2.5_lag_24h	pm2.5_rolling_mean_24h	hour_sincos	hour_sinsin	hour_cossin	day_sincos	day_sinsin	year_norm	year_sincos	
0	49	2010	1	3	0	90.25	-7	-5.0	1027.0	SE	58.56	4	0	2010-01-03 00:00:00	Unhealthy	150.0	129.0	Nan	0.000000	1.000000	0.05162	0.999957	0	0.0	1.0
1	50	2010	1	3	1	63.0	-8	-6.0	1026.0	SE	61.69	5	0	2010-01-03 01:00:00	Unhealthy	120.0	148.0	Nan	0.23819	0.95926	0.05162	0.999957	0	0.0	1.0
2	51	2010	1	3	2	65.0	-8	-7.0	1026.0	SE	65.71	6	0	2010-01-03 02:00:00	Unhealthy	90.0	159.0	Nan	0.290000	0.866025	0.05162	0.999957	0	0.0	1.0
3	52	2010	1	3	3	53.0	-8	-7.0	1025.0	SE	68.84	7	0	2010-01-03 03:00:00	Unhealthy for Sensitive Groups	63.0	181.0	Nan	0.707107	0.707107	0.05162	0.999957	0	0.0	1.0
4	53	2010	1	3	4	65.0	-8	-7.0	1024.0	SE	72.86	8	0	2010-01-03 04:00:00	Unhealthy	65.0	138.0	Nan	0.866025	0.500000	0.05162	0.999957	0	0.0	1.0

مزایای انکود کردن حفظ تناوب و نزدیکی مقادیر و بهبود عملکرد مدل ماشین و شناسایی الگوهای پنهان و جلوگیری از سوگیری مدل است.



مطابق شکل فوق مشاهده می کنیم تمرکز ناحیه ابی در ساعت ۶ صبح و عصر است که الودگی بشت بالاست و رنگ های قرمز و نارنجی در ساعت میانی روز و اوایل صبح حدود نیمه شب الودگی کمتر است.

### ویژگی زمان پیشرفت

با تعریف دیکشنری فضول را به یکی از ستون های دیتا فریم اضافه نمودیم. ایکدا دقت شود که ممکن است بنظر بیاید کلا تمام فصل ها زمستان بوده اما دادیه ۲۰۰۶۸ بررسی گردید و فصل بهار بود. شکل دیتا فریم بصورت زیر است: مشاهده می شود ستون آخر فصل است. برای درک هر چه بهتر شد

الودگی در فصول مختلف به شکل زیر توجه فرمایید:

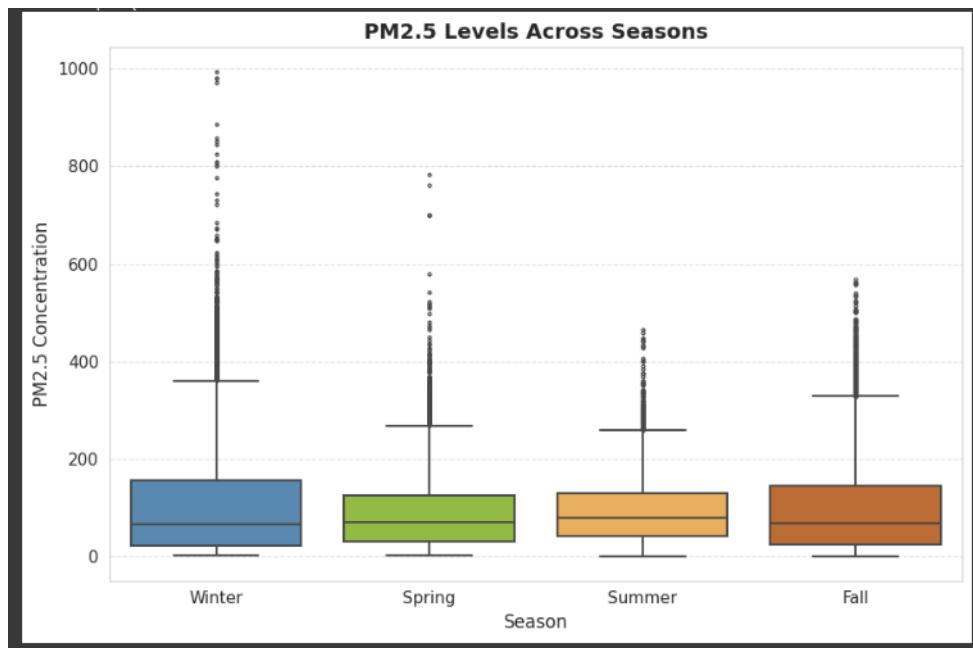
No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir	datetime	AQI_Catagory	pm2.5_lag_2h	pm2.5_lag_24h	pm2.5_roll_mean_24h	hour_sin	hour_cos	day_sin	day_cos	year_norm	year_sin	year_cos	season	
0	49	2010	1	3	0	90.0	7	-6.0	1027.0	SE	68.56	4	0	2010-01-03 00:00:00	Unhealthy	156.0	129.0	NaN	0.000000	1.000000	0.051620	0.998667	0	0.000000	1.000000	Winter
1	50	2010	1	3	1	63.0	-8	-6.0	1026.0	SE	61.69	5	0	2010-01-03 01:00:00	Unhealthy	126.0	148.0	NaN	0.258819	0.965926	0.051620	0.998667	0	0.000000	1.000000	Winter
2	51	2010	1	3	2	65.0	-8	-7.0	1026.0	SE	65.71	6	0	2010-01-03 02:00:00	Unhealthy	90.0	159.0	NaN	0.500000	0.866025	0.051620	0.998667	0	0.000000	1.000000	Winter
3	52	2010	1	3	3	55.0	-8	-7.0	1025.0	SE	68.84	7	0	2010-01-03 03:00:00	Unhealthy for Sensitive Groups	63.0	181.0	NaN	0.707107	0.707107	0.051620	0.998667	0	0.000000	1.000000	Winter
4	53	2010	1	3	4	65.0	-8	-7.0	1024.0	SE	72.86	8	0	2010-01-03 04:00:00	Unhealthy	65.0	138.0	NaN	0.866025	0.500000	0.051620	0.998667	0	0.000000	1.000000	Winter
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...			
43771	43820	2014	12	31	19	8.0	-23	-2.0	1034.0	NW	201.97	0	0	2014-12-31 19:00:00	Good	9.0	35.0	11.291667	-0.965926	0.258819	0.508671	0.860961	4	-0.951057	0.309017	Winter
43772	43821	2014	12	31	20	10.0	-22	-3.0	1034.0	NW	237.78	0	0	2014-12-31 20:00:00	Good	10.0	26.0	10.625000	-0.866025	0.500000	0.508671	0.860961	4	-0.951057	0.309017	Winter
43773	43822	2014	12	31	21	10.0	-22	-3.0	1034.0	NW	242.70	0	0	2014-12-31 21:00:00	Good	8.0	20.0	10.208333	-0.707107	0.707107	0.508671	0.860961	4	-0.951057	0.309017	Winter
43774	43823	2014	12	31	22	8.0	-22	-4.0	1034.0	NW	246.72	0	0	2014-12-31 22:00:00	Good	10.0	8.0	10.208333	-0.500000	0.866025	0.508671	0.860961	4	-0.951057	0.309017	Winter
43775	43824	2014	12	31	23	12.0	-21	-3.0	1034.0	NW	249.85	0	0	2014-12-31 23:00:00	Good	10.0	16.0	10.041667	-0.258819	0.965926	0.508671	0.860961	4	-0.951057	0.309017	Winter

مقادیر بدست امده برای میزان الاینده ها نشان میدهد که فصل زمستان الوده ترین فصل است.

```
(برای هر فصل PM2.5 (µg/m³):
season
Winter 109.792131
Fall 101.582509
Summer 91.739764
Spring 88.245380
Name: pm2.5, dtype: float64)
```

در ادامه به بررسی دلایل الوده تر بودن زمستان میپردازیم. اورنگی هوا را میتوان مهمترین علت دانست. همچنین افزایش استفاده از وسایل گرمایشی منجر به مصرف سوخت فسیلی بیشتر و کاهش وزش باد منجر به عدم تهویه هوا را نیز میتوان نام برد. همچنین رخداد پدیده مه دود که باعث معلق ماندن ذرات الاینده در هوا میشود نیز بی تاثیر نیست. همچنین شاهد کاهش بارندگی هستیم و درنتیجه هوا پاک نمیگردد.

همچنین برای روز های نیز شکل نمودار بصورت زیر است:



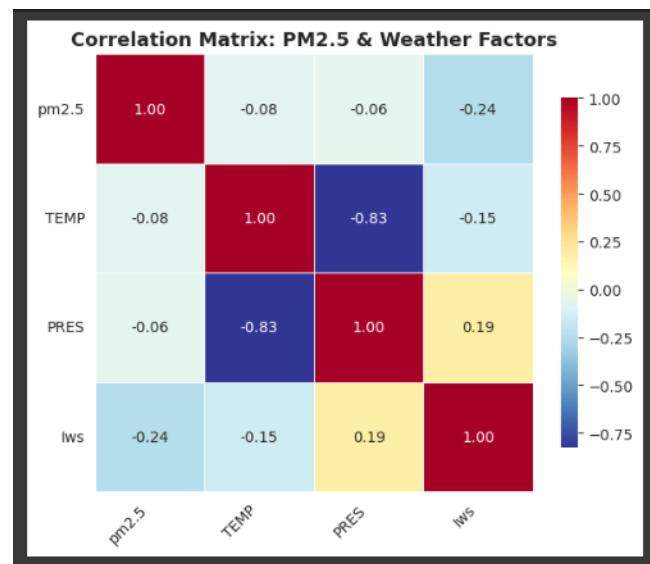
و مقادیر بدست میابید:



برای هر روز هنگام میتوان:	
weekday	Monday
	92.443008
	Tuesday
	96.249840
	Wednesday
	97.284483
	Thursday
	96.920353
	Friday
	99.845353
	Saturday
	102.989103
	Sunday
	98.595945
Name:	pm2.5, dtype: float64

که نشان میدهد الوده ترین روزها *friday, saturday* است. که میتواند به علت افزایش فعالیت های انسانی باشد.

ماتریس هم بستگی و مقادیر ان بصورت زیر است:



با عوامل PM2.5 همبستگی:	
pm2.5	1.000000
TEMP	-0.077505
PRES	-0.057698
Iws	-0.243117
Name:	pm2.5, dtype: float64

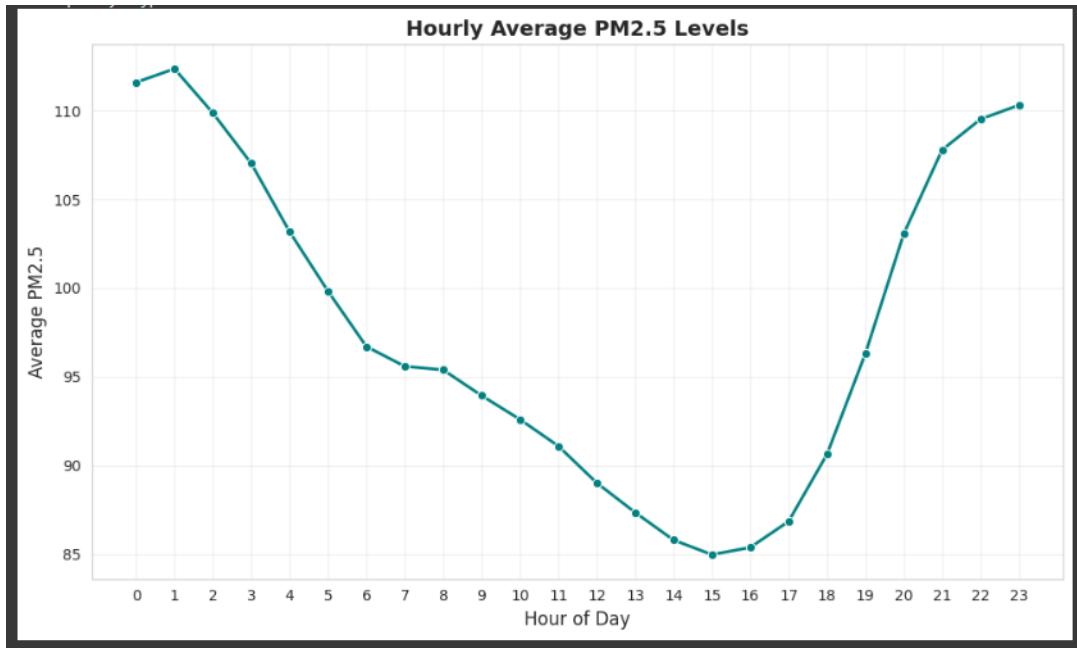
مقادیر بالا نشان میدهد فشار هوا هم بستگی منفی دارد پس تاثیر ان بر غلظت الاینده ها کم است همچنین دما نیز هم بستگی نیز همینطور است و با افزایش دما غلظت الاینده ها کاهش میابد و برای سرعت باد همبستگی بسیار قوی و منفی است بطوريکه با افزایش سرعت باد الاینده ها بطور قابل توجهی کاهش میابد. پس تمام این موارد رابطه عکس با غلظت الاینده ها دارد اما موثرترین انها سرعت باد است.

برای شبانه روز نیز داریم:

PM2.5 (برای هر ساعت $\mu\text{g}/\text{m}^3$ ): hour	
0	111.590461
1	112.354167
2	109.865132
3	107.024671
4	103.173246
5	99.823465
6	96.706140
7	95.590461
8	95.390899
9	93.944079
10	92.601425
11	91.101425
12	89.026864
13	87.361842
14	85.807018
15	84.983553
16	85.383224
17	86.853618
18	90.632127
19	96.304276
20	103.058114
21	107.785088
22	109.515351
23	110.305921

Name: pm2.5, dtype: float64

و نمودار روند تغییرات آن بصورت زیر است:



نشان میدهد که در ساعت نیمه شب و حدود ساعت یک بامداد بیشترین میزان الینده را داریم. همچنین بنظر میاید در ساعات ترافیک غلظت الینده ها افزایش یافته است. از جمله دلایل دیگر میتوان به فعالیت صنعتی شبانه کاهش سرعت باد در شب رخ داد پدیده وارونگی دما غالبا در شب که منجر به تجمع الیندگی های روز میشود اشاره کرد.

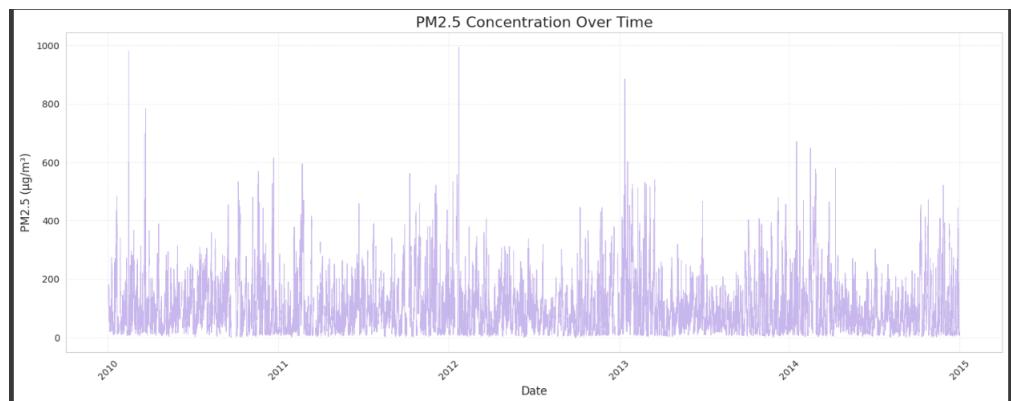
#### ۸.۲.۱

چهار و بیزگی سال ماه روز و ساعت را بصورت زیر ادغام کرده و در یک ستون از دیتا فریم اضافه مینمایم



I	s	Ir	datetime
4	0	2010-01-03 00:00:00	
5	0	2010-01-03 01:00:00	
6	0	2010-01-03 02:00:00	
7	0	2010-01-03 03:00:00	Unheal
8	0	2010-01-03 04:00:00	
...	...	...	...
0	0	2014-12-31 19:00:00	
0	0	2014-12-31 20:00:00	
0	0	2014-12-31 21:00:00	
0	0	2014-12-31 22:00:00	
0	0	2014-12-31 23:00:00	

سپس نمودار غلظت را ترسیم مینماییم برحسب زمان:



و مقادیر عددی بصورت میانگین یا مаксیمم بصورت زیر است:



```
month_avg = df_added_season.groupby('month')['pm2.5'].mean()
print("میانگین PM2.5 برای هر ماه ( $\mu\text{g}/\text{m}^3$ ):")
print(month_avg)

[Out[8]: میانگین PM2.5 برای هر ماه ( $\mu\text{g}/\text{m}^3$ ):
month
1    109.730664
2    126.092790
3     99.629832
4     84.046389
5     88.925269
6     96.887778
7     94.425538
8     84.072043
9     88.456389
10   118.242473
11   105.493333
12   95.024462
Name: pm2.5, dtype: float64]

[93]: max_row = df.loc[df['pm2.5'].idxmax()]
print(f"On {max_row['datetime']}, the highest PM2.5 level of {max_row['pm2.5']}  $\mu\text{g}/\text{m}^3$  was observed.")

[Out[93]: On 2012-01-23 01:00:00, the highest PM2.5 level of 994.0  $\mu\text{g}/\text{m}^3$  was observed.

[94]: df = df_added_rolling
max_row = df.loc[df['pm2.5'].idxmax()]
print(f"On {max_row['datetime']}, the PM2.5 level reached its maximum value of {max_row['pm2.5']}  $\mu\text{g}/\text{m}^3$ .")

[Out[94]: On 2012-01-23 01:00:00, the PM2.5 level reached its maximum value of 994.0  $\mu\text{g}/\text{m}^3$ .
```

اگر مقدار ماکزیمم را فرض کنیم بنظر می‌آید که شاید پارامتر مناسب را در نظر نگرفتیم زیرا اینکه روی از ماه بسیار الوده است اما با این حال بالاترین پیک برای زانویه است. اما مقدار ماکزیمم لزوماً منجر به الوده ترین بودن ماه نمی‌شود پس میانگین الودگی هر ماه را بررسی کردیم از روی میانگین نشان داده شده است که زانویه الوده تر است میتواند به علت فعالیت‌های انسانی اب و هوایی جوی و... باشد. با کمی تحقیق متوجه میشویم سرعتی در پکن زانویه است و احتمال وقوع وارونگی دارد از طرف دیگر برای زانویه ممکن است فعالیت‌های انسانی بیشتر باشد یا هوا خشک تر یا مصرف شدید سوخت فسیلی برای شروع فصل سرما یا شرایط جوی از قبیل باد کمتری باشد. یا ممکن است الودگی زانویه در حقیقت الودگی تجمیعی حاصل از دسامبر سرعت باد یا بارش سرما یا شرایط جوی از ان شود که الوده ترین ماه گردد. یا مصرف سوخت به مقداری پایدار رسیده باشد یا بارش همراه با رطوبت بیشتری باشد یا مساعد با تعطیلات باشد.

### ۹.۲.۱

برای بررسی توازن داده‌ها در اینجا از دو روش بهره می‌بریم یکی روش *undersampling* دیگری با استفاده از *smote* است. بر اساس شکل زیر بهوضوح می‌بینیم توازن داده‌ها برقرار نیست.



تعداد داده‌ها در هر ماه			
Month Number	Month Name	Count	
0	1	January	3672
1	2	February	3384
2	3	March	3720
3	4	April	3600
4	5	May	3720
5	6	June	3600
6	7	July	3720
7	8	August	3720
8	9	September	3600
9	10	October	3720
10	11	November	3600
11	12	December	3720

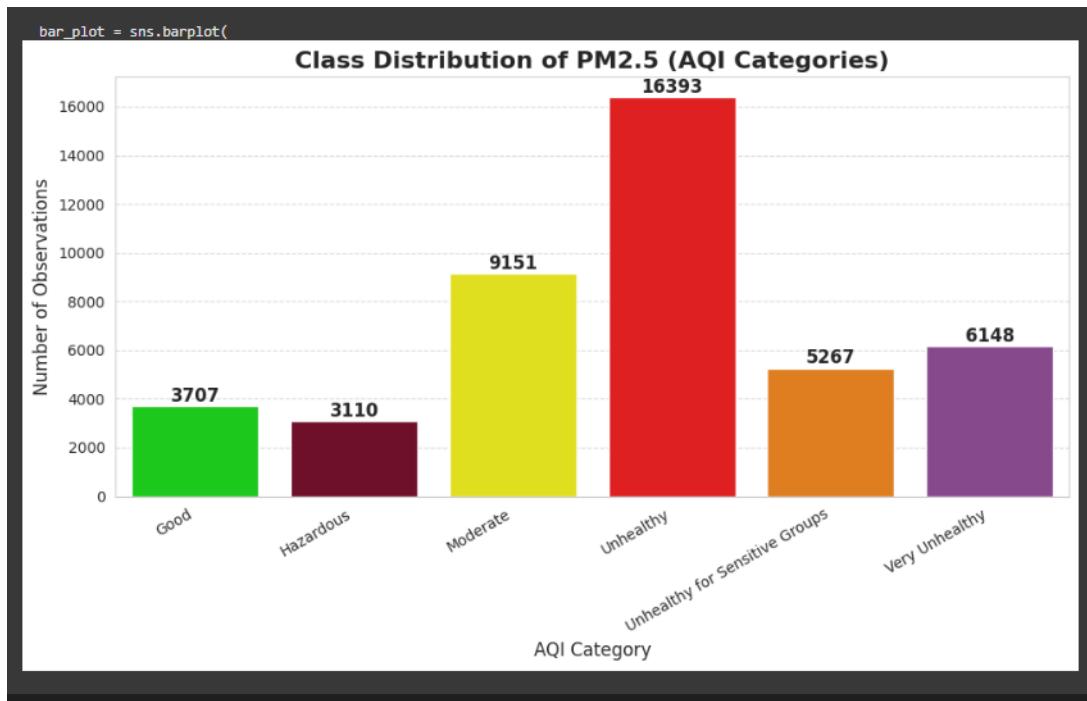
Day of Month	Count	
0	1	1416
1	2	1416
2	3	1440
3	4	1440
4	5	1440
5	6	1440
6	7	1440
7	8	1440
8	9	1440
9	10	1440
10	11	1440
11	12	1440
12	13	1440
13	14	1440
14	15	1440
15	16	1440
16	17	1440
17	18	1440
18	19	1440
19	20	1440
20	21	1440
21	22	1440
22	23	1440
23	24	1440
24	25	1440
25	26	1440
26	27	1440
27	28	1440
28	29	1344
29	30	1320
30	31	840

Next steps: [Generate code with month v\\_if](#)



Hour	Count
0	1824
1	1824
2	1824
3	1824
4	1824
5	1824
6	1824
7	1824
8	1824
9	1824
10	1824
11	1824
12	1824
13	1824
14	1824
15	1824
16	1824
17	1824
18	1824
19	1824
20	1824
21	1824
22	1824
23	1824

AQI Category	Count
Good	3707
Hazardous	3110
Moderate	9151
Unhealthy	16393
Unhealthy for Sensitive Groups	5267
Very Unhealthy	6148

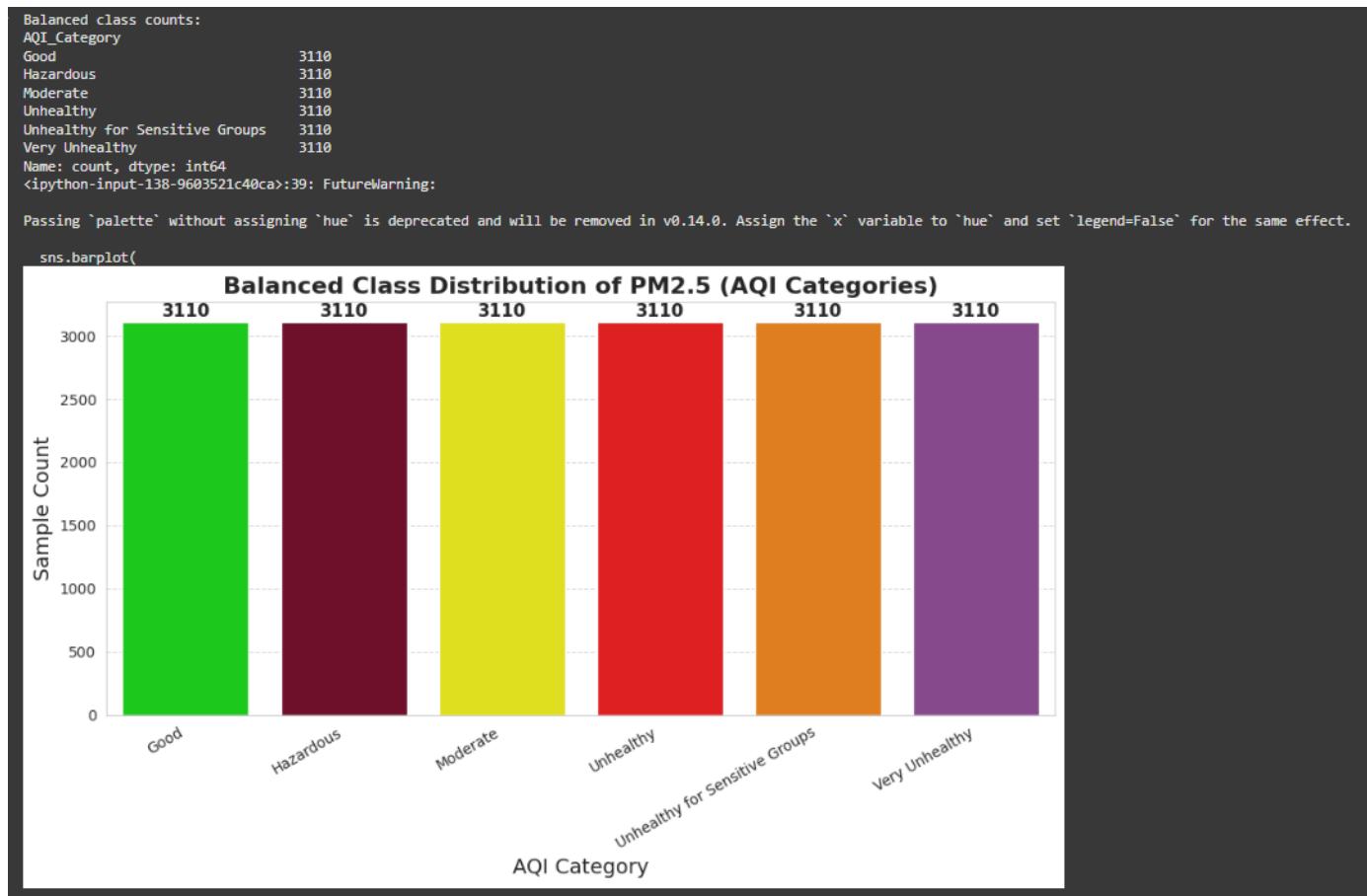


بهوضوح دیده میشود که داده ها توزان ندارند با توجه به تمامی شکل های فوق. پس از برقراری توزان دیتا فریم به شکل زیر خواهد بود:

	DEWP	TEMP	PRES	Iws	Ir	Is	cbwd_numeric	hour	month	weekday	pm2.5	AQI_Category	
0	-7	-6.0	1027.0	58.56	0	4		2	0	1	Sunday	90.0	Unhealthy
1	-8	-6.0	1026.0	61.69	0	5		2	1	1	Sunday	63.0	Unhealthy
2	-8	-7.0	1026.0	65.71	0	6		2	2	1	Sunday	65.0	Unhealthy
3	-8	-7.0	1025.0	68.84	0	7		2	3	1	Sunday	55.0	Unhealthy for Sensitive Groups
4	-8	-7.0	1024.0	72.86	0	8		2	4	1	Sunday	65.0	Unhealthy
...	...	...	...	...	...	...	...	...	...	...	...	...	
43771	-23	-2.0	1034.0	231.97	0	0		1	19	12	Wednesday	8.0	Good
43772	-22	-3.0	1034.0	237.78	0	0		1	20	12	Wednesday	10.0	Good
43773	-22	-3.0	1034.0	242.70	0	0		1	21	12	Wednesday	10.0	Good
43774	-22	-4.0	1034.0	246.72	0	0		1	22	12	Wednesday	8.0	Good
43775	-21	-3.0	1034.0	249.85	0	0		1	23	12	Wednesday	12.0	Good

43776 rows x 12 columns

پس از توازن برقرار کردن بین کلاس ها تعداد داده هی هر کلاس و شکل خروجی بصورت زیر است:



#### نکته مهم

بدیهی است این توزین کلاس ها فقط روی داده های آموزش صورت میگیرد. توجه داریم که باید پس از جداسازی توزین کلاس ها صورت گیرد. ارگر در اینجا خروجی ها نمایش داده شده است اما در دیتا فرمی با نامی دیگر ذخیره شده و در ادامه کد استفاده نشده صرفا برای نمایش خروجی بوده است. در ادامه مدل



ابتدا جداسازی سپس با *smote* توزین شده است.

#### ۱۰.۲.۱

از روش *minmaxscalar* استفاده میکنیم. اما دقت داریم داده ها همه عددی باشند.

	DEWP	TEMP	PRES	Iws	Ir	Is	cbwd_numeric	hour	month	weekday	pm2.5	AQI_Category	Actions
0	0.485294	0.213115	0.654545	0.099308	0.0	0.148148	0.666667	0.000000	0.0	1.000000	0.090543	Unhealthy	
1	0.470588	0.213115	0.636364	0.104657	0.0	0.185185	0.666667	0.043478	0.0	1.000000	0.063380	Unhealthy	
2	0.470588	0.196721	0.636364	0.111527	0.0	0.222222	0.666667	0.086957	0.0	1.000000	0.065392	Unhealthy	
3	0.470588	0.196721	0.618182	0.116876	0.0	0.259259	0.666667	0.130435	0.0	1.000000	0.055332	Unhealthy for Sensitive Groups	
4	0.470588	0.196721	0.600000	0.123746	0.0	0.296296	0.666667	0.173913	0.0	1.000000	0.065392	Unhealthy	
...	...	...	...	...	...	...	...	...	...	...	...	...	
43771	0.250000	0.278689	0.781818	0.395659	0.0	0.000000	0.333333	0.826087	1.0	0.333333	0.008048	Good	
43772	0.264706	0.262295	0.781818	0.405588	0.0	0.000000	0.333333	0.869565	1.0	0.333333	0.010060	Good	
43773	0.264706	0.262295	0.781818	0.413996	0.0	0.000000	0.333333	0.913043	1.0	0.333333	0.010060	Good	
43774	0.264706	0.245902	0.781818	0.420866	0.0	0.000000	0.333333	0.956522	1.0	0.333333	0.008048	Good	
43775	0.279412	0.262295	0.781818	0.426216	0.0	0.000000	0.333333	1.000000	1.0	0.333333	0.012072	Good	

دیتا فریم پس از نرمال سازی به شکل فوق است. علت استفاده از این روش مقایس بندی کردن داده ها بین صفر تا یک است که باعث میشود ستون ها با بازه های متفاوت با یک مقایس قیاس گردند. از طرفی برای مدل سای یادگیری ماشینی برای مدل هایی که به مقایس حساسند مانند گرادیان نزولی مفید است. سایر ستون ها را نیز حذف کردیم.

#### ۱۱.۲.۱

در کولب به طور کامل انجام شده است.

#### ۱۲.۲.۱

پس از تقسیم داده ها به داده یها اموزش و اعتبار سنجی و ازمون داریم:

```
SMOTE: Counter({'Unhealthy': 11474, 'Unhealthy for Sensitive Groups': 11474, 'Moderate': 11474, 'Hazardous': 11474, 'Very Unhealthy': 11474, 'Good': 11474})
Train: (68844, 10) (68844, )
Validation: (6566, 10) (6566, )
Test: (6567, 10) (6567, )
```

نیازی به انکود کردن خروجی نیست زیرا *svm* خودش ان را لیبل در نظر میگیرد.

پارامترهای مختلفی بر اساس نوع *svm* برای بهینه سازی استفاده میگردد. این هایپر پارامتر ها در کرنل اهمیت ویژه ای دارند زیر به شکل مرز تصمیم توانایی تعیین مدل و درنتیجه عملکرد الگوریتم اثر مستقیم دارند. هر کرنل *svm* با هایپر پارامترهای مخصوص خود اگر به درستی تعیین گردد دقت مدل افزایش قابل توجهی خواهد داشت. و *svm* بین کلاس ها مرزی بهینه را تعیین میکند. اگر هم داده ها توسط خط جدای ناپذیر بودند با کرنل داده ها را به فضای ویژگی بالاتری نگاشت کرده و باعث جدای پذیری انها میگردد. هایپر پارامترها تعیین میکند این نگاشت چقدر پیچیه باشد یا چ میزان اهمیت به داده های نزدیک و پرت تخصیص داده شود.



جدول جامع مقایسه کرnel‌ها در یادگیری ماشین					
کرnel	هایپرپارامترها	مزایا	معایب	بهترین شرایط استفاده	
- ندارد (در SVM فقط C مهم)	- درجه چندجمله‌ای : <code>degree</code> - ضریب ثابت : <code>coef0</code> - بیجیدگی مدل : <code>C</code> -	سریع، ساده، قابل تفسیر، مناسب داده‌های پرباعاد	مدل ساده‌ای دارد، روابط غیرخطی را پوشش نمی‌دهد	وقتی داده‌ها به صورت تقریبی خطی قابل تفکیک باشند یا ویژگی‌ها زیاد باشند (مثل داده‌های متنی)	
<b>Polynomial</b> (چندجمله‌ای)	کنترل وسعت : <code>gamma</code> - تأثیر نقاط : <code>C</code> -	انعطاف‌پذیر برای روابط غیرخطی، حساس به پارامترها، ممکن است overfit	کندر از کرnel خطی، ساختار خاص (درجات پایین) در داده‌ها وجود دارد	وقتی روابط غیرخطی با ساختار خاص (درجات پایین) در داده‌ها برای داده‌های با مرز تضمیم پیچیده و غیردقیق، زمانی که همچیزی از شکل مرز نرم دانم	
<b>RBF / Gaussian</b>	کنترل وسعت : <code>gamma</code> - کنترل نقاط : <code>C</code> -	پرقدرت، پوشش‌دهنده الگوهای پیچیده، استفاده رایج	بسیار حساس به <code>gamma</code> و <code>C</code> ، نیاز به تنظیم دقیق	برای داده‌های با مرز تضمیم پیچیده و غیردقیق، زمانی که همچیزی از شکل مرز نرم دانم	
<b>Sigmoid</b>	مشابه وزن ورودی : <code>gamma</code> - ضریب بایاس : <code>coef0</code> -	الهام‌گرفته از شبکه عصبی، گاهی مؤثر نیست	معمولًا دقت کمتری دارد، خروجی کرnel همیشه مثبت	برای شبیه‌سازی رفتار نورون‌ها یا وقتی مدل شبیه شبکه عصبی مدنظر باشد	
<b>Precomputed / Custom</b>	- جدول شباهت از قبل محاسبه شده - بدون پارامتر خاص در مدل	امکان استفاده از داشن تخصصی، قابل سفارش‌سازی بالا	نیاز به تعریف یا محاسبه شباهت‌های ساختاری، ژنتیکی، یا شیمیایی	برای مسائل خاص مثل شباهت‌های خاص، حافظه بر	

جدول فوق مقایسه عملکرد کرnel‌ها هایپرپارامترهای آنها و شرایط استفاده از انها را توضیح داده است. این هایپرپارامترها را با دو تکنیک *gridsearch*, *randomsearch* میتوان بهینه نمود.

مطابق شکل نوع کرnel را کاربرد و داده‌ها تعیین میکند و رایج‌ترین مدل‌های آن *linear*, *poly*, *rbf*, *sigmoid* است.

برای مثال اگر هایپرپارامتر *C* در کرnel خطی کم باشد مدل انعطاف‌پذیری تر است اما امکان رخداد اندرفیتینگ هست و بر عکس یعنی اگر مقدارش زیاد باشد احتمال وقوع اورفیتینگ هست و مدل انعطاف‌کمتری دارد.

یا هایپرپارامتر گاما برای تعیین تأثیر هر نمونه اموزش استفاده می‌شود و اگر مقدار آن زیاد باشد باعث اورفیتینگ می‌شود.

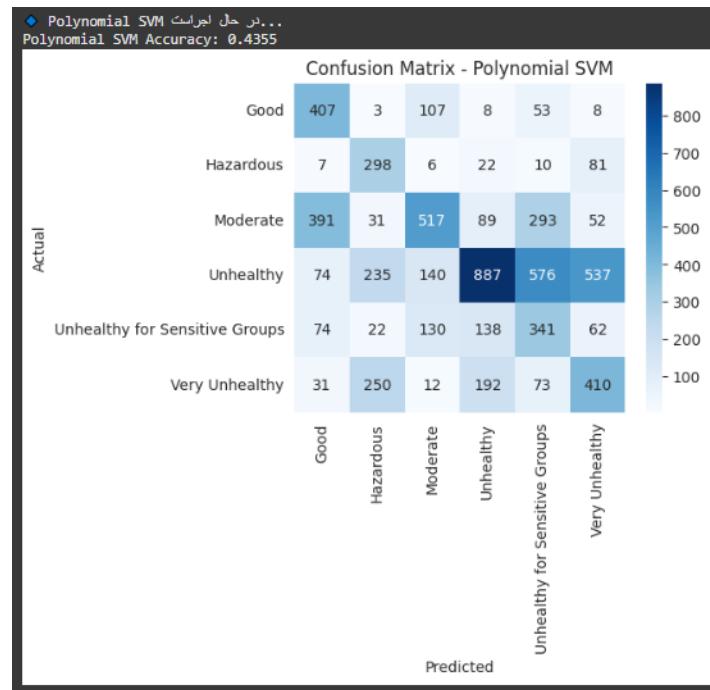
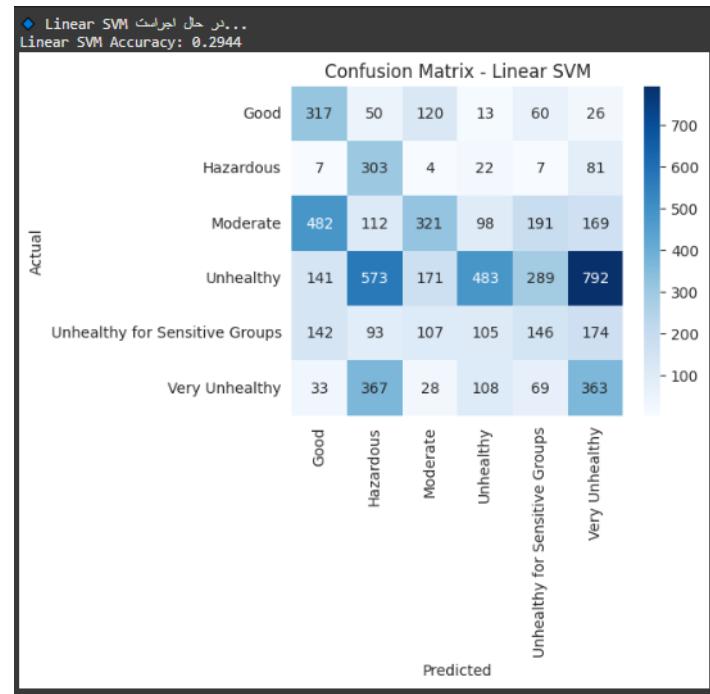
هایپرپارامتر درجه برای تعیین درجه چند جمله‌ای استفاده می‌شود و بدیهی است برای کرnel چند جمله‌ای تعریف می‌گردد.

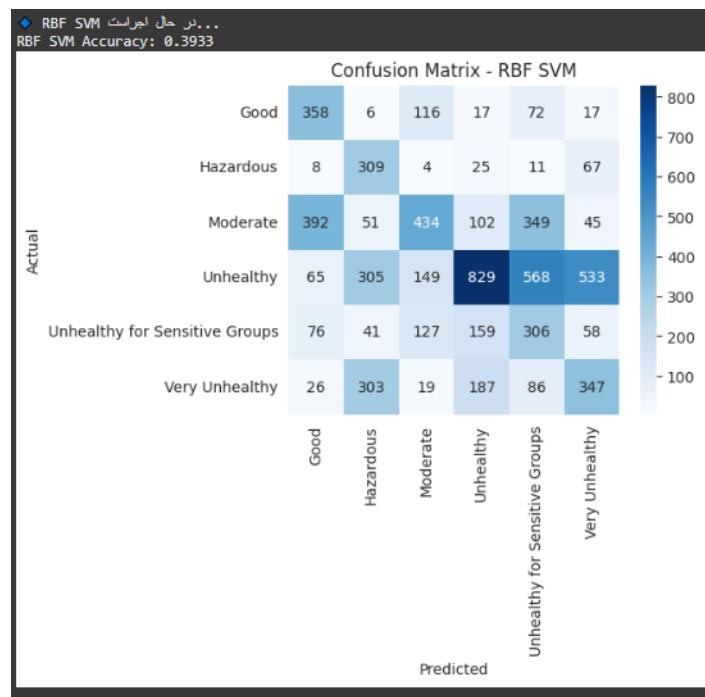
حالا سه نوع از کرnel‌ها را اعمال می‌کنیم و نتایج را بررسی می‌کنیم.

```
Linear SVM Accuracy: 1.0000
Polynomial SVM Accuracy: 0.9939
RBF SVM Accuracy: 0.8914
```

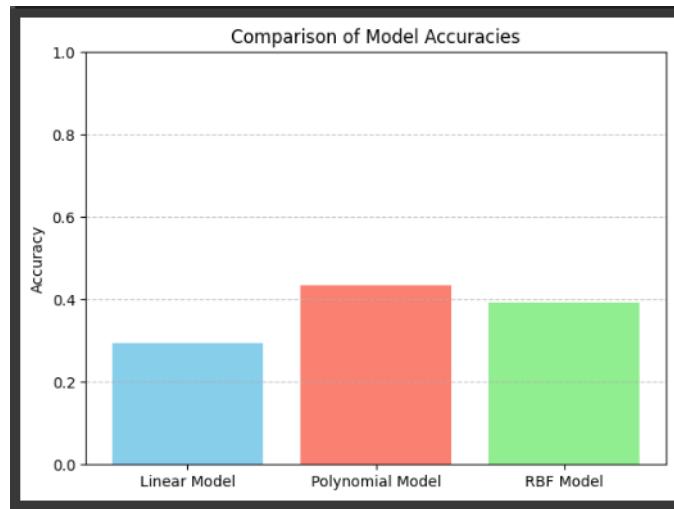
نتایج در ابتدا بسیار راضی کننده بود اما با دقت بیشتر متوجه می‌شویم که اگر دقت یک شده است لزوماً خبر خوبی نیست و باید به دنبال بررسی اورفیت یا نشتی داده یا وابستگی ویژگی‌ها به یکدیگر و مواردی دیگر باشیم. ابتدا بررسی کردیم *smote* پس از *split* صورت گرفته باشد.

نکته بسیار مهم این است که باید دقت داشته باشیم که ترتیب بسیار مهم است ترتیب انجام عملیات‌ها عبارت است از ابتدا *split* را انجام میدهیم سپس *minmaxscaler* و بعد از آن *smote*. پس از بررسی دقیق آنها و اصلاح کد نتایج بصورت زیر قابل مشاهده است:





از مقایسه میزان دقت مشاهده میکنیم چند جمله‌ای موثر تر عملکرد :



از روی شکل مشاهده میکنیم برای حالت چند جمله‌ای موثر تر بوده است : در ادامه برای سرعت بخشیدن به کار تعداد ۲۰۰ داده را انتخاب میکنیم

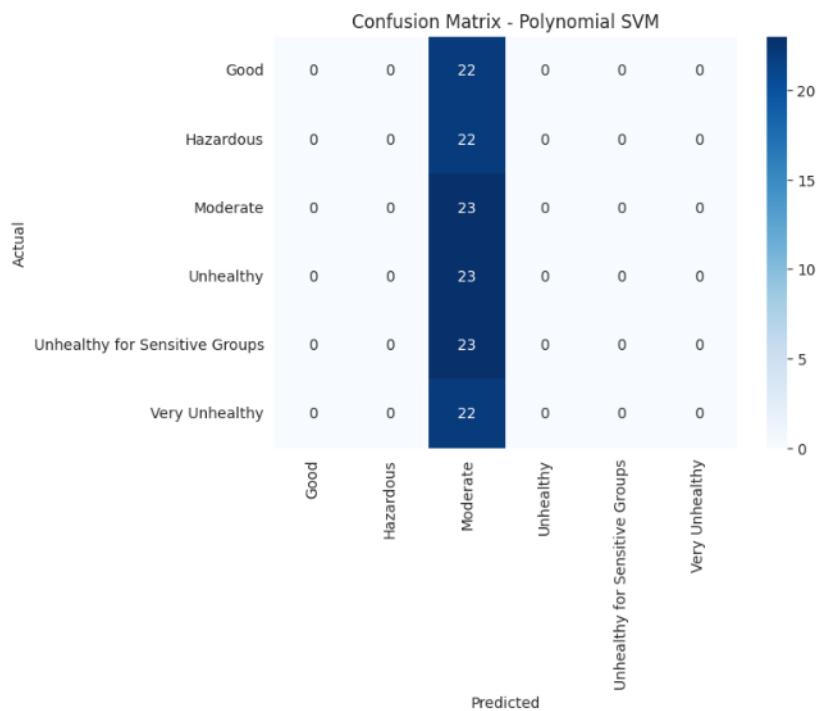
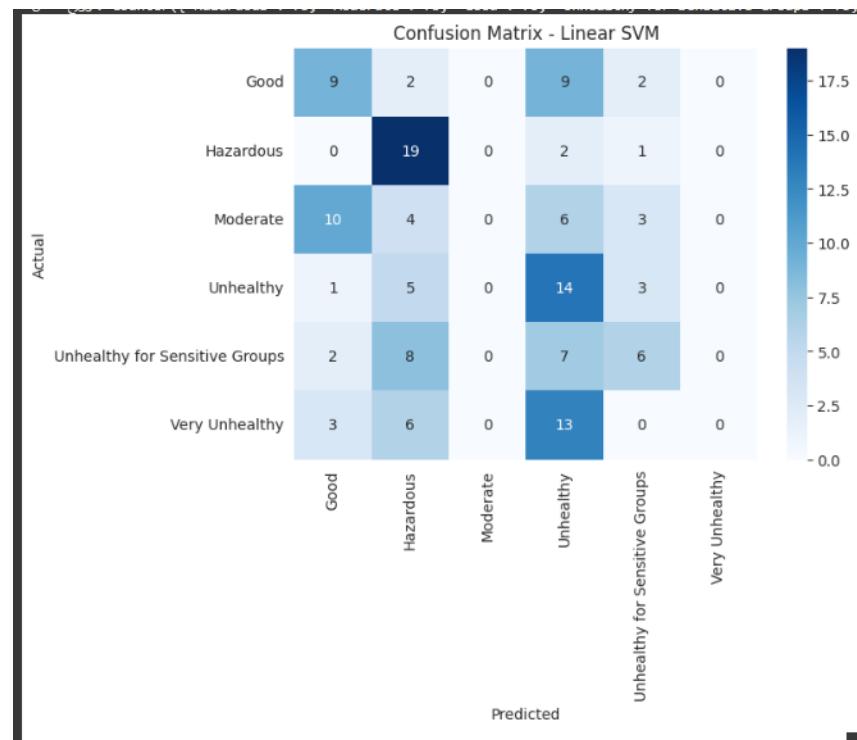
```

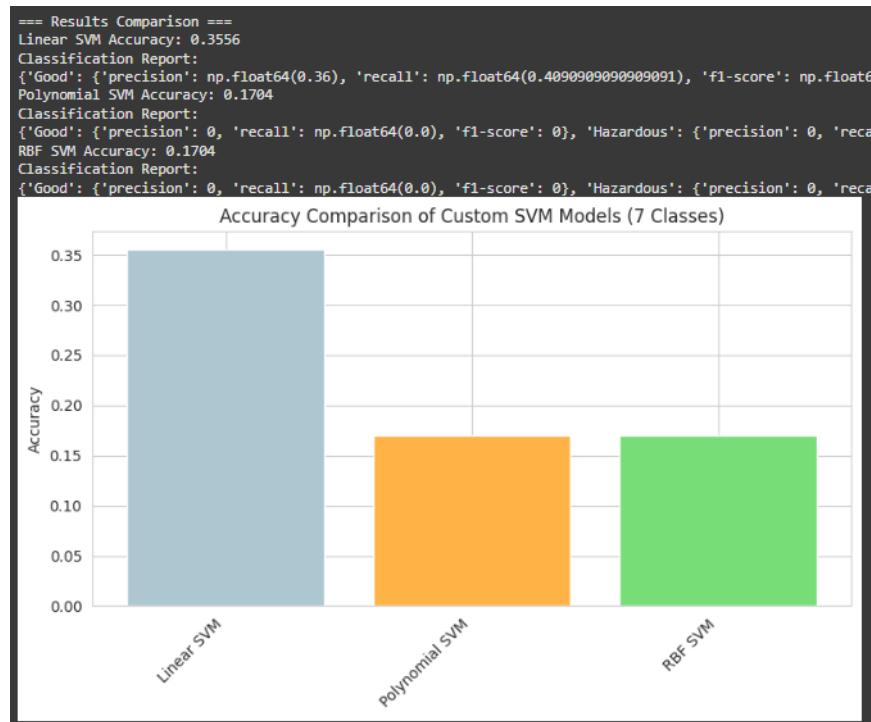
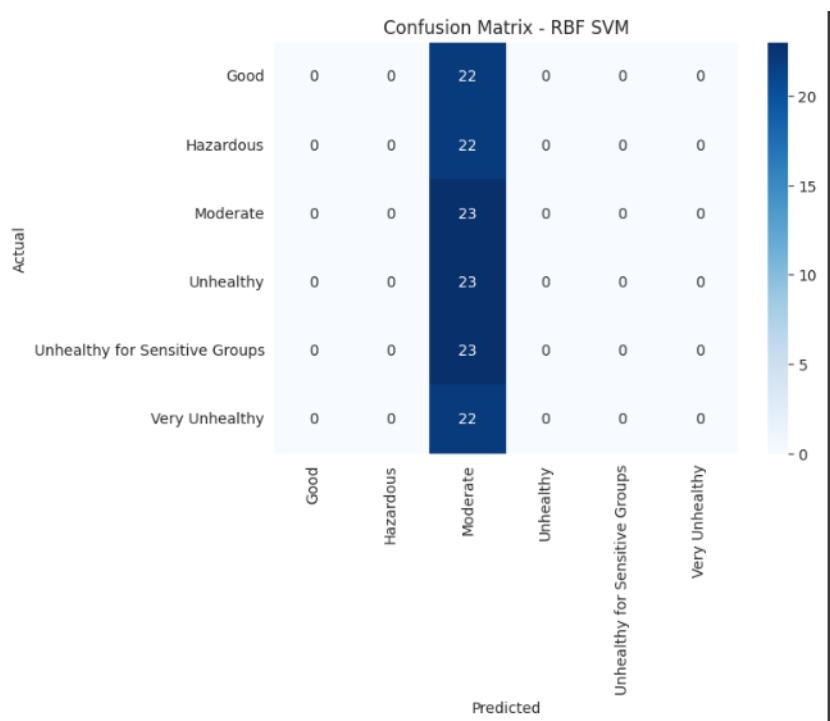
قبل از SMOTE:
تعداد نمونه‌ها: 200
توزیع کلاس‌ها: Counter({'Unhealthy': 75, 'Moderate': 42, 'Very Unhealthy': 28, 'Unhealthy for Sensitive Groups': 24, 'Good': 17, 'Hazardous': 14})

بعد از SMOTE:
تعداد نمونه‌ها: 450
توزیع کلاس‌ها: Counter({'Hazardous': 75, 'Moderate': 75, 'Good': 75, 'Unhealthy for Sensitive Groups': 75, 'Very Unhealthy': 75, 'Unhealthy': 75})

```

و ادامه میدهیم این بار برای کد دستی خروجی‌ها به شکل زیر است:





از روی شکل ها متوجه میشویم که خطی موفق تر عمل کرده است.

در مقایسه دقت بین مدل های آمده در کتابخانه های استاندارد و مدل های پیاده سازی شده دستی، عموماً مدل های موجود در کتابخانه هایی مانند اس کی لرن دقت بالاتری دارند، بهویژه در استفاده از کرنل های غیرخطی. دلیل این امر استفاده از روش های پیشرفته بهینه سازی مانند روش بهینه سازی دوتایی حداقل دنباله ای است که به طور خاص برای حل مسائل دسته بندی با ماشین بردار پشتیبان طراحی شده اند. همچنین، مقادیر پیش فرض بسیاری از پارامترها مانند

گاما در اس کی لرن به صورت بهینه و استاندارد در نظر گرفته شده‌اند که به بهبود عملکرد مدل کمک می‌کند.

در مقابل، مدل‌های دستی که توسط کاربر پیاده‌سازی می‌شوند، معمولاً دقت پایین‌تری دارند، بعویظه در شرایطی که از کرنل‌های غیرخطی استفاده می‌شود.

یکی از دلایل اصلی این تفاوت، استفاده از روش‌های ساده‌تری مانند گرادیان نزولی به جای روش‌های پیچیده بهینه‌سازی است که ممکن است تنها به یک مینیمم محلی برسند و جواب بهینه کامل را پیدا نکنند. همچنین، در مسائل چندکلاسه، پیاده‌سازی استراتژی‌هایی مانند «یکی در برابر بقیه» به صورت ساده‌تر ممکن است باعث عدم تعادل در مدیریت کلاس‌ها شود.

از نظر ماتریس سردرگمی، مدل‌های آماده در کتابخانه‌های معتبر معمولاً ماتریس‌های دقیق‌تری را نمایش می‌دهند که نشانه توزیع نسبتاً متداول کلاس‌ها و دسته‌بندی دقیق‌تر آن‌هاست. علت این دقت بیشتر، تنظیم بهینه پارامترها و پیاده‌سازی دقیق کرنل‌هاست که باعث کاهش خطأ در طبقه‌بندی می‌شود.

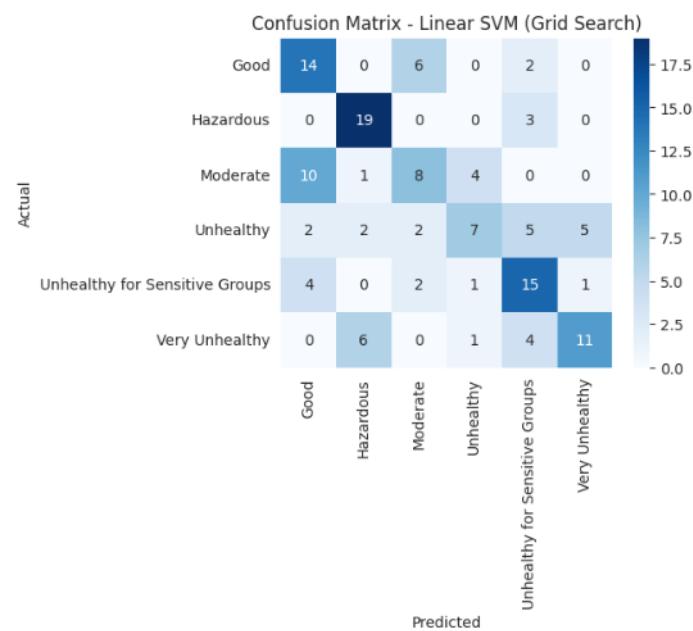
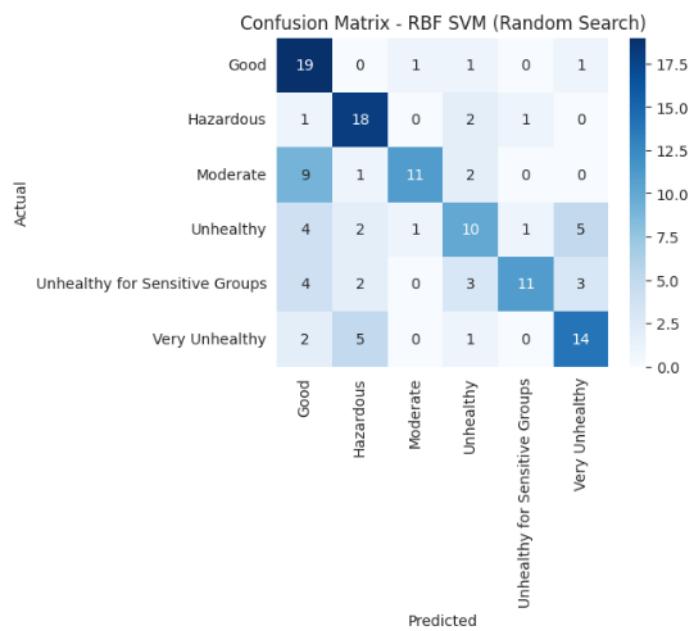
در حالی که در مدل‌های دستی، احتمال بروز خطاهای بیشتر در برخی کلاس‌ها، بعویظه کلاس‌هایی با تعداد نمونه کمتر یا کلاس‌هایی که مرز آن‌ها به خوبی از سایر کلاس‌ها جدا نشده، بیشتر است. این امر به دلیل محدودیت در تنظیم دقیق پارامترهایی مانند گاما و  $\gamma$  و عدم استفاده از روش‌های پیشرفت‌بهینه‌سازی است.

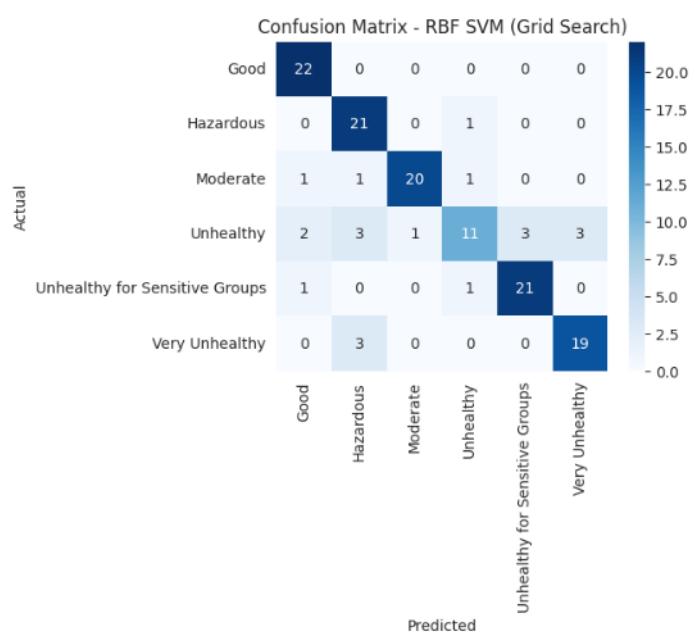
در سنجش شاخص‌های ارزیابی همچون دقت، پادآوری و امتیاز  $\text{F1}$  نیز معمولاً مدل‌های موجود در کتابخانه‌ها عملکرد بهتری دارند. این مدل‌ها می‌توانند به طور دقیق و معادل بین کلاس‌ها تمایز قائل شوند و به همین دلیل در مسائل چندکلاسه که نیاز به مدیریت متوازن کلاس‌ها وجود دارد، بهتر عمل می‌کنند. مدل‌های دستی، در صورتی که توزیع داده‌ها نامتعادل باشد یا مرز میان کلاس‌ها به خوبی مشخص نشده باشد، ممکن است در برخی کلاس‌ها مقدار دقت یا پادآوری پایین‌تری داشته باشند، زیرا نمی‌توانند مرز بهینه را به خوبی پیدا کرده و خطای طبقه‌بندی را کاهش دهند.

از نظر زمان اجرا نیز تفاوت قابل توجهی بین این دو نوع پیاده‌سازی وجود دارد. مدل‌های اس کی لرن به دلیل استفاده از کتابخانه‌هایی مانند `sklearn` و `scikit-learn` بهینه‌سازی‌های داخلی، هم در مرحله آموزش و هم در مرحله پیش‌بینی، زمان سیار کمتری صرف می‌کنند. در حالی که مدل‌های دستی، بعویظه در استفاده از الگوریتم‌های ساده مانند گرادیان نزولی و با تعداد تکرار زیاد، زمان اجرای بالاتری دارند. همچنین، در استفاده از کرنل‌های غیرخطی، زمان محاسبه ماتریس کرنل نیز می‌تواند بهشدت افزایش یابد، که بر عملکرد کلی تأثیر می‌گذارد.

واینبار محاسبات را با استفاده از رنک سرج و گرید سرج ادامه میدهیم:

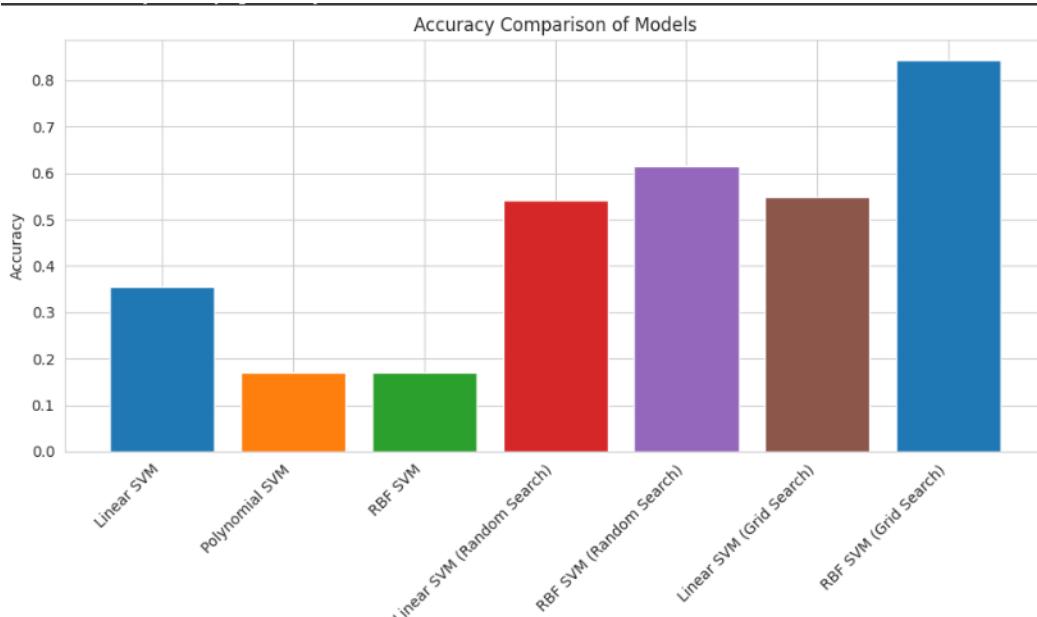
Confusion Matrix - Linear SVM (Random Search)						
Actual	Predicted					
	Good	Hazardous	Moderate	Unhealthy	Unhealthy for Sensitive Groups	Very Unhealthy
Good	13	0	7	0	2	0
Hazardous	0	19	0	0	3	0
Moderate	9	1	8	4	1	0
Unhealthy	2	2	4	7	3	5
Unhealthy for Sensitive Groups	3	1	3	1	15	0
Very Unhealthy	0	6	0	1	4	11





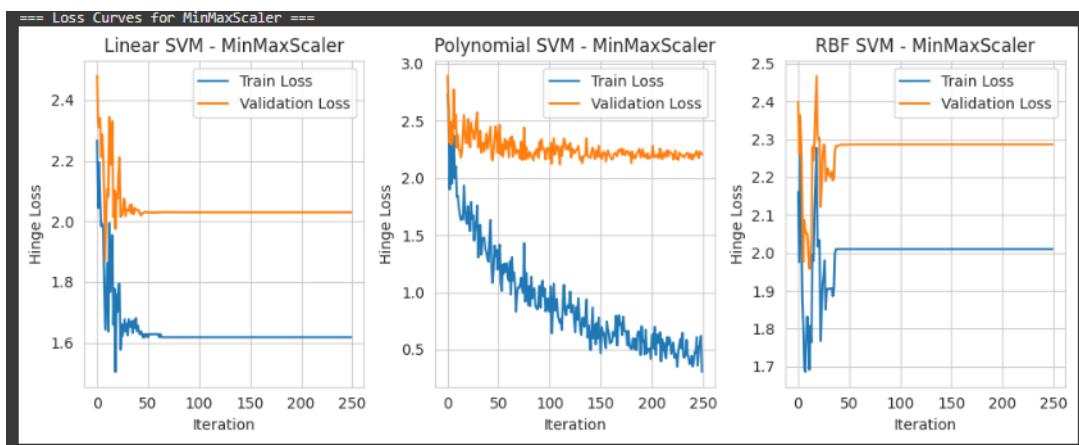
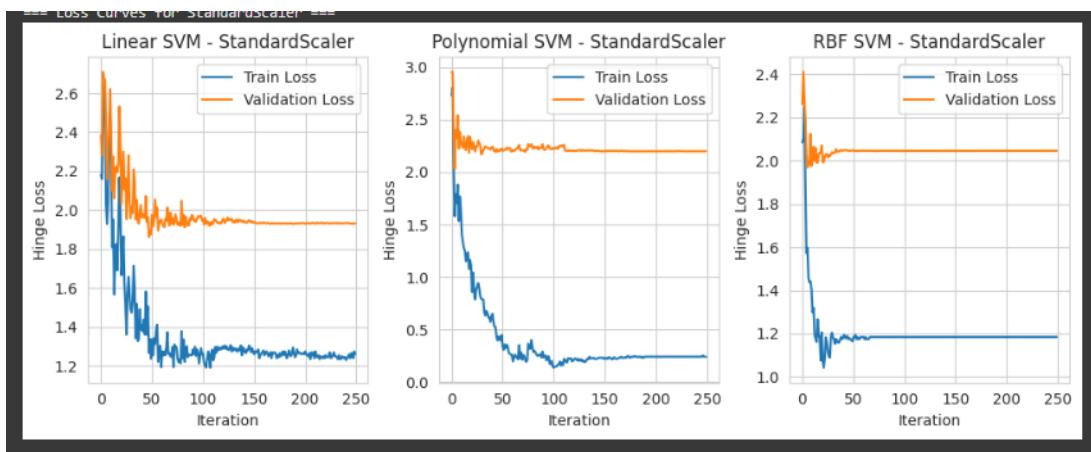
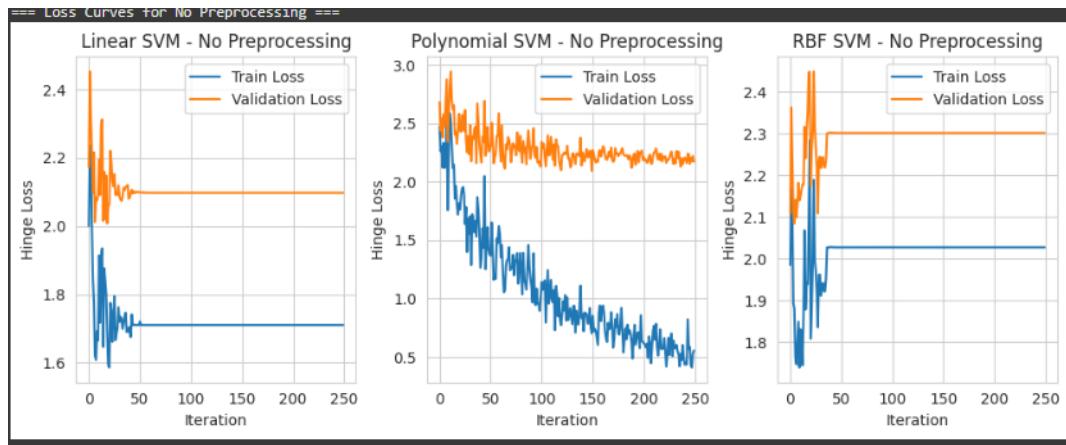
حال برای مقایسه حالات استفاده از رندم سرج و گرید سرج و دستی داریم:

```
== Results Comparison ==
Linear SVM Accuracy: 0.3556
Polynomial SVM Accuracy: 0.1704
RBF SVM Accuracy: 0.1704
Linear SVM (Random Search) Accuracy: 0.5407
RBF SVM (Random Search) Accuracy: 0.6148
Best Parameters: {'C': np.float64(157.41890047456639)}
Linear SVM (Grid Search) Accuracy: 0.5481
Best Parameters: {'C': 100}
RBF SVM (Grid Search) Accuracy: 0.8444
Best Parameters: {'C': 100, 'gamma': 1}
```





مشاهده میکنیم که استفاده از گرید و رندم سرج بسیار کمک کننده بوده است.  
در ادامه نمودار های اتلاف برای حالت با پیش پردازش و بدون آن را مشاهده میکنیم:



واضح است که اعمال پیش پردازش باعث کاهش خطأ می‌گردد.



## ۱۳.۲.۱

در روش بردار پشتیبان برای پیش‌بینی، اگر مسئله مربوط به مقداردهی باشد، یعنی بخواهیم عددی مانند قیمت یا دما را پیش‌بینی کنیم، مدل تلاش می‌کند تابعی پیدا کند که بتواند با خطای کمتر از یک میزان مشخص، مقدارها را تخمین بزند. در این حالت، مدل به جای اینکه داده‌ها را به دسته‌هایی جدا کند، به دنبال نزدیک شدن به مقدار درست است. ولی اگر هدف ما تشخیص دسته‌ی داده‌ها باشد، مثلاً بگوییم سالم است یا ناسالم، مدل با پیدا کردن مرزهایی میان دسته‌ها، داده‌ها را از هم جدا می‌کند. در حالت اول، کیفیت مدل با اندازه‌گیری میانگین خطای سنجیده می‌شود و در حالت دوم، با سنجش درصد درستی پیش‌بینی‌ها.

بله، این مدل دارای چند هایپرپارامتر است که برای گرفتن نتیجه بهتر باید به درستی انتخاب شوند. این مقدارها مشخص می‌کنند مدل چقدر سختگیر باشد، چقدر به داده‌های نزدیک توجه کند، یا تا چه اندازه اجازه دهد خطای داشته باشد. اگر این مقدارها به خوبی تنظیم نشوند، ممکن است مدل یا پیش از حد ساده شود و نتواند الگوها را یاد بگیرد، یا بیش از حد پیچیده شود و فقط داده‌های آموزش دیده را بشناسد. برای انتخاب درست این مقدارها، معمولاً آن‌ها را روی مجموعه‌ای جداگانه آزمایش می‌کنند تا بهترین نتیجه را بگیرند.

چون این مدل بر پایه‌ی جدا کردن داده‌ها از هم ساخته شده، هدفش اینه که بین دو گروه مختلف یه مرز مشخص ایجاد کنه که تا جای ممکن، از داده‌های هر دو گروه فاصله داشته باشه. برای اینکه بتونه این مرز رو دقیق‌تر و هدفمندتر بسازه، باید نمونه‌های کافی و متنوعی از هر گروه داشته باشه. وقتی داده‌ی بیشتری از گذشته یا از آزمایش‌های جدید دریافت می‌کنه، با بررسی تفاوت‌های بین گروه‌ها، می‌تونه مرزی پیدا کنه که بهتر دو گروه رو از هم جدا کنه. این داده‌های متنوع باعث می‌شن شکل‌ها و موقعیت‌های مختلفی از هر گروه دیده بشه و همین، در شناخت دقیق‌تر مرز کمک زیادی می‌کنه. در نتیجه، هر چقدر داده‌های بیشتر و متنوع‌تری داشته باشه، در برخورد با نمونه‌های تازه هم عملکرد بهتری نشون می‌ده، چون تجربه‌ی بیشتری از حالت‌های مختلف داره و می‌تونه تصمیم دقیق‌تری بگیره.

## جمع بندی کل سوالات :

این مدل چون بر اساس جدا کردن دو گروه مختلف ساخته شده، باید بین داده‌های اون‌ها یه مرز مشخص و مؤثر بکشه که از هر دو طرف تا جای ممکن فاصله داشته باشه. برای اینکه بتونه این مرز رو دقیق‌تر پیدا کنه، نیاز به نمونه‌های متنوع از هر گروه داره؛ یعنی نمونه‌هایی که از نظر موقعیت و شکل متفاوت باشند تا مرز رو به شکل هدفمندتر و واقعی‌تری بسازه. وقتی از داده‌های قبلی یا نتایج آزمایش‌های قبلی دوباره استفاده می‌کنه، در واقع داره دید خودش رو نسبت به انواع حالت‌هایی که ممکنه بین داده‌ها پیش بیاد، گسترش می‌ده. این باعث می‌شه بتونه تفاوت بین گروه‌ها رو بهتر درک کنه و در نهایت مرزی بسازه که فقط مخصوص همون حالت‌های دیده‌شده نیست، بلکه در برخورد با نمونه‌های جدید هم کارایی داشته باشه. بنابراین استفاده‌ی دوباره از داده‌ها نه تنها قدرت تشخیص مدل رو بالا می‌بره، بلکه کمک می‌کنه که در مواجهه با شرایط جدید، تصمیم‌های دقیق‌تری بگیره و کمتر دچار اشتباه بشه.

در مدل‌های ماشین بردار پشتیبان، یکی از مهم‌ترین پارامترها پارامتر "سی" است. این پارامتر نقش کنترل میزان خطای خارج از ناحیه مجاز را بر عهده دارد. در مدل طبقه‌بندی، داده‌هایی که خارج از ناحیه مرزی قرار می‌گیرند، به عنوان خطای در نظر گرفته می‌شوند، در حالی که در مدل رگرسیون، نقاطی که بیرون از لوله‌ای به شعاع اپسیلون قرار دارند، خطای محسوس می‌شوند. مقدار بزرگ "سی" باعث می‌شود مدل حساس‌تر به داده‌ها شود و احتمال بیش‌برازش افزایش باید در حالی که مقدار کوچک‌تر آن، مدلی با تعمیم بهتر ولی با دقت کمتر ایجاد می‌کند. معمولاً این پارامتر در بازه‌ای مانند صفر ممیز یک تا صد تنظیم می‌شود. این پارامتر در هر دو مدل نقش مشابه‌ی دارد.

در ادامه، نوع هسته نیز اهمیت زیادی دارد. این هسته‌ها، رابطه‌های غیرخطی بین ویژگی‌ها را به فضای ویژگی‌های جدید نگاشت می‌کنند تا مدل بتواند داده‌های پیچیده‌تر را تشخیص دهد. از پرکاربردترین هسته‌ها می‌توان به خطی، چندجمله‌ای و پایه شعاعی اشاره کرد. هسته خطی نیاز به هیچ پارامتر اضافه‌ای ندارد. هسته چندجمله‌ای دارای پارامترهای مانند درجه چندجمله‌ای و عدد ثابت است. هسته پایه شعاعی نیز دارای پارامتر گاما است که تعیین کننده گستره تأثیر هر داده روی بقیه داده‌است. پارامترهای مربوط به نوع هسته در هر دو مدل یکسان هستند و بسته به نوع داده و مسئله تنظیم می‌شوند.

پارامتر دیگری که در هر دو مدل وجود دارد، حداکثر تعداد تکرار الگوریتم برای رسیدن به حالت بهینه است. این پارامتر مشخص می‌کند که الگوریتم یادگیری تا چه میزان مجاز به تلاش برای بهینه‌سازی مدل است. در هر دو مدل طبقه‌بندی و رگرسیون از این پارامتر برای کنترل زمان آموزش استفاده می‌شود. یکی از تفاوت‌های اصلی بین مدل رگرسیون و طبقه‌بندی در وجود یا نبود پارامتر اپسیلون است. در مدل رگرسیون، اپسیلون تعیین می‌کند که چه میزان خطای در پیش‌بینی بدون حریمه قابل قبول است. مقدار بزرگ‌تر آن باعث می‌شود مدل نسبت به خطایها کمتر حساس باشد و خطای بیشتری را مجاز بداند، اما مقدار



کوچک آن موجب حساسیت بیشتر مدل و احتمال بیش برآش می‌شود. این پارامتر در مدل طبقه‌بندی وجود ندارد. هدف مدل‌ها نیز با هم تفاوت دارد. در مدل طبقه‌بندی، هدف یافتن مرز بیشینه‌ای است که بتواند دسته‌های مختلف داده را با کمترین خطا از هم جدا کند. در حالی که در مدل رگرسیون، هدف یافتن تابعی است که بیشترین تعداد نقاط را درون لوله‌ای به عرض اپسیلون قرار دهد و تنها نقاط خارج از این محدوده به عنوان خطا در نظر گرفته شوند.

از نظر شاخص‌های ارزیابی عملکرد نیز تفاوت‌های وجود دارد. در مدل طبقه‌بندی، دقت، یادآوری، صحبت و میانگین موزون شاخص‌ها برای ارزیابی مدل استفاده می‌شوند. در این مدل، توانایی جداسازی کلاس‌ها اهمیت زیادی دارد و دقت آن به شدت به تنظیم درست پارامترهایی مانند سی و گاما وابسته است. همچنین در مسائل چندکلاسه، از روش‌هایی مثل یکی در برابر بقیه یا یکی در صورت نامتوازن بودن دسته‌ها می‌تواند باعث کاهش دقت شود. از طرف دیگر، مدل رگرسیون معمولاً با معیارهایی مانند میانگین مربع خطا یا میانگین قدر مطلق خطا ارزیابی می‌شود. این مدل برای پیش‌بینی مقادیر عددی مانند دما، زمان یا ارتفاع استفاده می‌شود و عملکرد آن نیز به تنظیم پارامترهایی مانند سی، اپسیلون و گاما بستگی دارد. اپسیلون نقش مهمی در تعیین حساسیت مدل نسبت به خطا دارد و انتخاب درست آن به دقت پیش‌بینی کمک می‌کند.

از نظر زمان آموزش، مدل طبقه‌بندی ممکن است به زمان بیشتری نیاز داشته باشد، بهخصوص در مسائل چندکلاسه یا در استفاده از هسته‌های غیرخطی مثل چندجمله‌ای و پایه شعاعی. زمان پیش‌بینی در این مدل معمولاً کوتاه‌تر است، چون فقط نیاز به انجام محاسبات بر اساس بردارهای پشتیبان دارد. با افزایش تعداد داده‌ها و بردارهای پشتیبان، زمان پیش‌بینی می‌تواند بیشتر شود. در مقابل، مدل رگرسیون معمولاً زمان آموزش کوتاه‌تری دارد، چون فقط یک مدل برای خروجی عددی ساخته می‌شود. زمان پیش‌بینی آن هم مشابه مدل طبقه‌بندی به تعداد بردارهای پشتیبان وابسته است.

از نظر تعمیم‌پذیری نیز، مدل طبقه‌بندی در صورتی که به خوبی تنظیم شده باشد، توانایی خوبی در جداسازی دسته‌های جدید دارد، اما در صورت وجود کلاس‌های نامتوازن یا داده‌های پراکنده ممکن است چار بیش برآش یا کم برآش شود. در مدل رگرسیون، تعمیم‌پذیری بیشتر به پارامتر اپسیلون وابسته است؛ انتخاب درست آن می‌تواند باعث نادیده گرفتن خطاهای کوچک و بهبود عملکرد مدل روی داده‌های واقعی شود. در عین حال، گاما نیز در هسته‌های غیرخطی تأثیر زیادی بر عملکرد تعمیم‌پذیری دارد و انتخاب نادرست آن می‌تواند باعث بیش برآش مدل شود.

در نهایت، در کاربردهای مختلف نیز بسته به نوع مسئله از یکی از این دو مدل استفاده می‌شود. مدل طبقه‌بندی برای مسائلی که داده‌ها باید به دسته‌های مجزا تقسیم شوند مناسب‌تر است، مانند تشخیص بیماری، دسته‌بندی تصاویر یا تشخیص اعداد. در مقابل، مدل رگرسیون برای پیش‌بینی مقادیر عددی مانند زمان، دما، قیمت یا وزن مناسب‌تر است. مدل رگرسیون معمولاً در صورت آموزش صحیح، مقدار خطای کمتری نسبت به مدل طبقه‌بندی برای مسائل عددی ایجاد می‌کند، در حالی که مدل طبقه‌بندی در مسائل دسته‌بندی، عملکرد بهتری دارد.

## ۱۴.۲.۱

این الگوریتم یک الگوریتم هوش جمعی و تصادفی است که برای بهینه سازی مسائل پیوسته و گستته بکار می‌رود این الگوریتم از رفتار اجتماعی گروه‌های طبیعی مانند پرستو‌ها الهام گرفته شده است. در این روش مجموعه‌ای از نقاط یا ذرات در فضا جست و جو و حرکت می‌کنند. هر ذره جواب ممکنی از مسئله است که در فضای پارامترها موقعیت دارد و سرعتی که با نکوچیت خود را بروز می‌کند به همراه حافظه ای از بهترین موقعیت خود ذره و کل گروه. این الگوریتم با ترکیب دانش فردی و اجتماعی به سمت بهترین جواب مایل است. بهینه سازی ازدحام ذرات توسط کندی و ابرهارت معرفی شده است. موقعیت هر ذره نماینگر یکی از هایپرپارامترهاست. دقت داریم که هر ذره یک هایپرپارامتر است که نوع ان بر حسب نوع کرنل تعیین می‌شود.

از بهترین مزایا آن ساده و مناسب بودن برای مسائل غیرخطی و چندوجهی عدم نیاز به محاسبات گرادیان و انعطاف پذیری بالا برای بهینه سازی است. و از معايب این میتوان به مینیم محلی گیری و کند بودن برای مسائل ابعاد بالا اشاره نمود.

برای *svm* به هر ذره یک بازه تعريف می‌شود و مدل آموزش داده می‌شود تا دقت آن محاسبه گردد و به حرکت بهترین نتیجه برای موقعیت خود و کل گروه را به عنوان جواب خروجی میدهد این ذرات با سرعت اولیه کمی شروع به حرکت مینمایند و الگوریتم پس از رسیدن به دقت مطلوب پایان می‌ابد



```

Requirement already satisfied: gdown in /usr/local/lib/python3.11/dist-packages (5.2.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (from gdown) (4.13.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from gdown) (3.18.0)
Requirement already satisfied: requests[socks] in /usr/local/lib/python3.11/dist-packages (from gdown) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from gdown) (4.67.1)
Requirement already satisfied: soupsieve<1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4->gdown) (2.7)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4->gdown) (4.14.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (2025.4.26)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (1.7.1)
Downloading...
From: https://drive.google.com/uc?id=1jTKH15Fr0E03r0QDoxoGystT8oLP7fP
To: /content/PRSA_data_2010.1.1-2014.12.31.csv
100% 2.01M/2.01M [00:00<00:00, 186MB/s]
 لذت شنیدن Linear SVR:
Validation MSE: 12473.1485
Test MSE: 12405.2943
Test R2: -0.0157
/usr/local/lib/python3.11/dist-packages/sklearn/svm/_base.py:305: ConvergenceWarning: Solver terminated early (max_iter=500). Consider pre-processing your data with StandardScaler().fit_transform()
warnings.warn(

```

## ۱۵.۲.۱

پس از پیاده سازی باید نتایج عملکرد *svmrbf*, *pssoptimizedrbfsvm* را با یکدیگر مقایسه کنیم که نتیجه بصورت زیر است:

```

Collecting pyswarms
  Downloading pyswarms-1.3.0-py2.py3-none-any.whl.metadata (33 kB)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from pyswarms) (1.15.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from pyswarms) (2.0.2)
Requirement already satisfied: matplotlib>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from pyswarms) (3.10.0)
Requirement already satisfied: attrs in /usr/local/lib/python3.11/dist-packages (from pyswarms) (25.3.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from pyswarms) (4.67.1)
Requirement already satisfied: future in /usr/local/lib/python3.11/dist-packages (from pyswarms) (1.0.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/dist-packages (from pyswarms) (6.0.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=1.3.1->pyswarms) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=1.3.1->pyswarms) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=1.3.1->pyswarms) (4.58.1)
Requirement already satisfied: kimisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=1.3.1->pyswarms) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=1.3.1->pyswarms) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=1.3.1->pyswarms) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=1.3.1->pyswarms) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=1.3.1->pyswarms) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib>=1.3.1->pyswarms) (1.17.0)
Downloading pyswarms-1.3.0-py2.py3-none-any.whl (104 kB)
  104.1/104.1 kB 3.3 MB/s eta 0:00:00
Installing collected packages: pyswarms
Successfully installed pyswarms-1.3.0
2025-06-09 17:11:42,893 - pyswarms.single.global_best - INFO - Optimize for 20 iters with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|20/20, best_cost=77.4
2025-06-09 17:16:13,750 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 77.41647627472805, best pos: [81.04126013 0.99430893 0.40126255]
 Best Parameters Found:
C: 81.04126013299695
gamma: 0.9943089308855513
epsilon: 0.4012625537048098
Best RMSE: 77.41647627472805

```

لذت شنیدن PSO:

MAE: 48.91  
RMSE: 70.17  
R<sup>2</sup> Score: 0.287

پیاده سازی مقادیر را بصورت فوق نمایش میدهد.

۳.۱

۱.۲.۱

۲.۳.۱

در این مقاله از یک دیتاست مربوط به آلودگی هوای شهری استفاده شده است. این داده‌ها از پایگاه داده متن باز زیستمحیطی Qingyue Open و ایستگاه‌های نظارتی شهر Xi'an در سال ۲۰۲۲ جمع‌آوری شده‌اند.

نوع داده‌ها شامل مقادیر عددی مرتبط با غلظت آلاینده‌های هواست که به صورت زمانی و مکانی از ایستگاه‌های نظارت محیطی ثبت شده‌اند. ویژگی‌های ثبت شده در هر رکورد عبارت‌انداز: غلظت  $\text{NO}_2$  بر حسب  $\mu\text{g}/\text{m}^3$ , غلظت  $\text{CO}$  بر حسب  $\mu\text{g}/\text{m}^3$ , غلظت  $\text{O}_3$  بر حسب  $\mu\text{g}/\text{m}^3$ , مقدار  $\text{PM}_{10}$  (ذرات معلق با قطر کمتر از ۱۰ میکرومتر) و  $\text{PM}_{2.5}$  (ذرات معلق با قطر کمتر از ۵.۲ میکرومتر)، شاخص کیفیت هوای Quality که به صورت طبقه‌بندی شده و سطح‌بندی مانند I, II, III, IV و ... نمایش داده شده، و همچنین زمان ثبت داده به صورت Pubtime مقدار هدف (برچسب کلاس) براساس شاخص کیفیت هوای AQI تعریف شده است که شامل سطح‌بندی‌هایی از عالی تا بسیار آلوده می‌باشد و به صورت عدد گسسته (سطح ۱ تا ۶ یا بیشتر) دسته‌بندی شده است.

در مرحله پیش‌پردازش، داده‌ها به صورت میانگین‌گیری ساعتی تجمیع شده‌اند تا نوسانات لحظه‌ای حذف شوند و سپس نرم‌افزاری ویژگی‌ها انجام شده است تا مقیاس تمام ویژگی‌ها یکنواخت شود.

در ادامه، این داده‌ها به یک مدل بهینه‌شده SVM داده شده‌اند که با استفاده از الگوریتم Differential Gravitational Fireworks Algorithm آموزش دیده و به دسته‌بندی دقیق سطوح آلودگی هوای کمک کرده است.

۳.۳.۱

### مدل استفاده شده و نوآوری‌های اعمال شده در ساختار SVM:

در این مقاله برای بهبود عملکرد مدل SVM در طبقه‌بندی سطوح کیفیت هوای از یک روش بهینه‌سازی ترکیبی جدید با عنوان Differential Gravitational Fireworks Optimization (DGFO) استفاده شده است. این الگوریتم حاصل ترکیب سه الگوریتم هوشمند فراابتکاری است که هر یک به تهیی دارای ویژگی‌ها و توانایی‌های خاص در جست‌وجوهی فضای پارامترها می‌باشند: الگوریتم Differential Evolution (DE), الگوریتم Fireworks Algorithm (FWA) و الگوریتم Gravitational Search Algorithm (GSA).

الگوریتم DE یکی از روش‌های پرکاربرد در بهینه‌سازی سراسری است که توسط Storn و Price معرفی شد. این الگوریتم بهدلیل ساختار ساده و مبتنی بر بردار تفاضلی، توانایی بالایی در جست‌وجوهی جهانی دارد و در تعیین مقادیر مناسب پارامترهایی مانند  $C$  و  $\gamma$  برای SVM بسیار مؤثر است. الگوریتم GSA توسط Rashedi و همکاران در سال ۲۰۰۹ توسعه داده شد. این الگوریتم با استفاده از مفاهیم فیزیکی گرانش و جرم، برای هر ذره نیروی تعزیزی کند که باعث جابه‌جایی آن به سوی ذرات سنگین‌تر (مناسب‌تر) می‌شود. این ساختار می‌تواند در همگرایی سریع به نواحی بهینه بسیار مؤثر باشد. الگوریتم Tan Ying FWA معرفی شد و با الهام از پخش ذرات آتش‌بازی در فضا، سعی دارد تنوع جمعیت و جهش‌های موضعی را حفظ کند. این ویژگی‌ها باعث می‌شوند الگوریتم از گیر افتادن در مینیمم‌های محلی جلوگیری کرده و بتواند به راه حل‌های بهتر دست یابد.

ترکیب این سه الگوریتم در ساختار DGFO باعث می‌شود از مزایای هر یک به طور همزمان بهره گرفته شود. برای مثال، DE در جست‌وجوهی جهانی قدرتمند است، GSA در تقویت همگرایی مؤثر عمل می‌کند، و FWA باعث افزایش تنوع و پایداری الگوریتم در فضاهای پیچیده می‌شود. در مجموع، DGFO می‌تواند ساختارهای جدیدی در فضای ویژگی تولید کرده و باعث بهبود عملکرد مدل یادگیری شود.

در این مدل، بهمنظور جلوگیری از بیش‌پرازش (Overfitting)، تابع هدفی مبتنی بر معکوس میانگین مربعات خطای  $(1/\text{MSE})$  تعریف شده است که ذراتی با دقت بالاتر وزن و احتمال انتخاب بیشتری دارند. همچنین، از استراتژی بهینه‌سازی مبتنی بر حریمه نیز استفاده شده است که در آن پارامترهای  $C$  و  $\gamma$  به طور همزمان تنظیم و بهینه‌سازی می‌شوند.

نتایج ارزیابی تجربی نشان داده‌اند که مدل DGFO-SVM در مقایسه با مدل‌های کلاسیک SVM و روش‌های بهینه‌سازی تکی عملکرد قابل توجهی دارد.

این مدل توانسته است در طبقه‌بندی سطوح آلودگی هوا به دقتی در حدود % دست یابد و در مقایسه با روش‌های تربیتی و غیرتربیتی، نتایج پایدارتر و دقیق‌تری ارائه دهد.

مراحل اصلی پیاده‌سازی مدل شامل چهار فاز است: پیش‌پردازش داده‌ها، تولید جمعیت اولیه با استفاده از DE، اعمال جست‌وجوی تربیتی با استفاده از FWA و GSA و در نهایت آموزش مدل SVM با پارامترهای بهینه. این ساختار به‌گونه‌ای طراحی شده که بتواند هم در داده‌های ساختگی و هم در داده‌های واقعی با کیفیت پایین یا نویز بالا عملکرد مناسبی ارائه دهد. مدل DGFO-SVM به عنوان یک چارچوب تطبیق‌پذیر، توانسته است توانایی الگوریتم‌های مختلف را در یک ساختار منسجم ترکیب کند و کاربردهای گسترده‌ای در مسائل دسته‌بندی، پیش‌بینی و تحلیل داده‌های محیطی داشته باشد.

#### ۴.۳.۱

#### ۵.۳.۱

##### پیاده‌سازی الگوریتم پیشنهادی مقاله:

در این بخش، الگوریتم پیشنهادی مقاله با هدف طبقه‌بندی سطح کیفیت هوای شهری پیاده‌سازی شده است. فرآیند پیاده‌سازی شامل شش گام کلیدی است که هر کدام به صورت دقیق و جداگانه بررسی شده‌اند. این مراحل از بارگذاری داده‌ها و پیش‌پردازش آغاز شده و تا طراحی مدل، بهینه‌سازی، ارزیابی تجربی و تحلیل الگوریتم‌های جستجوی تکاملی ادامه می‌یابد.

##### ۱. آماده‌سازی و پالایش داده‌ها:

داده‌های مورد استفاده مربوط به ایستگاه Aotizhongxin در شهر پکن بوده و از مجموعه داده‌های چندایستگاهی کیفیت هوا استخراج شده‌اند. با توجه به محدودیت زمانی تعیین شده در مسئله، فقط داده‌های مربوط به سال‌های ۲۰۱۳ تا ۲۰۱۵ در تحلیل در نظر گرفته شده‌اند. ابتدا داده‌های دارای مقادیر گمشده حذف شدند تا از بروز خطاهای محاسباتی در مراحل بعدی جلوگیری گردد. در مرحله بعد، به‌منظور افزایش دقت و پایداری مدل، داده‌های پرت نیز با استفاده از روش Interquartile Range (IQR) شناسایی و حذف شدند. پس از پاکسازی داده‌ها، فرآیند برچسب‌گذاری صورت گرفت. با استفاده از مقدار PM2.5، سطوح کیفیت هوا به پنج کلاس مختلف (moderate, good, very unhealthy, unhealthy, hazardous) طبقه‌بندی شدند. این برچسب‌گذاری به‌منظور تعریف یک مسئله یادگیری تحت نظارت انجام شده است.

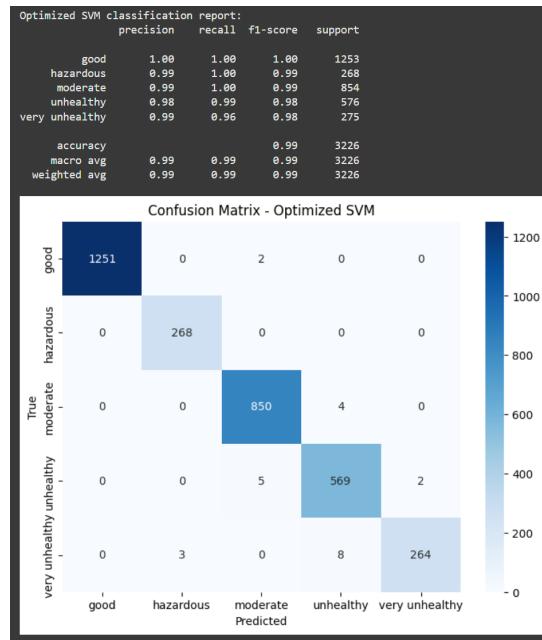
##### ۲. طراحی مدل SVM و بهینه‌سازی پارامترها با DE:

برای طبقه‌بندی سطوح کیفیت هوا، از ماشین بردار پشتیبان (SVM) با کرنل RBF استفاده شد. این مدل به دلیل قدرت تعمیم بالا و عملکرد مناسب در فضاهای غیرخطی انتخاب گردید. از آنجا که عملکرد SVM به‌شدت وابسته به مقادیر پارامترهای C و gamma است، یک مرحله بهینه‌سازی به‌منظور تعیین مقادیر مناسب این پارامترها انجام گرفت.

الگوریتم Differential Evolution (DE) به عنوان روش جستجوی سراسری جهت انجام این بهینه‌سازی مورد استفاده قرار گرفت. تابع هدف برای ارزیابی ترکیب‌های مختلف پارامترها، دقت حاصل از اعتبارسنجی متقابل پنج‌لایه‌ای (5-fold cross-validation) تعريف شد. فضای جستجو برای C و gamma نیز با حدود بالا و پایین معقول انتخاب شد تا هم پوشش کافی داشته باشد و هم از اتلاف منابع محاسباتی جلوگیری شود.

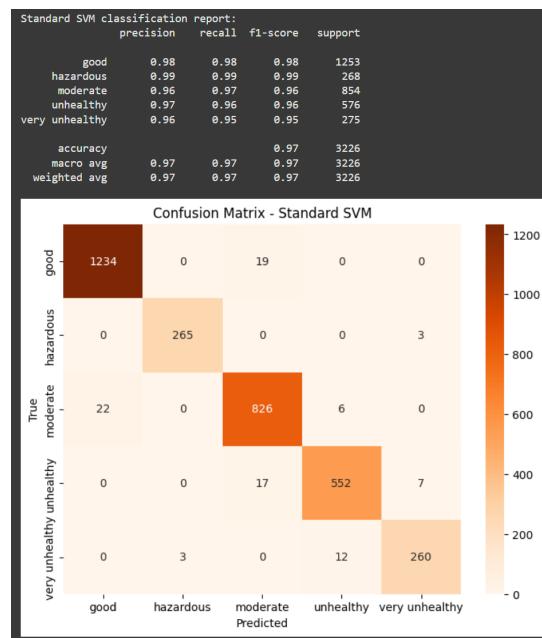
##### ۳. آموزش و ارزیابی مدل بهینه‌شده:

پس از مشخص شدن پارامترهای بهینه، مدل SVM با استفاده از داده‌های آموزش داده شد و بر روی داده‌های تست مورد ارزیابی قرار گرفت. گزارش طبقه‌بندی شامل دقت، فراخوانی و معیار F1-score برای هر کلاس محاسبه گردید. علاوه بر این، برای تحلیل دقیق‌تر عملکرد مدل، ماتریس درهم‌ریختگی (Confusion Matrix) نیز ترسیم شد. نتایج حاصل نشان‌دهنده قدرت مدل در تفکیک کلاس‌های مختلف کیفیت هوا بودند، بهویژه در کلاس‌های میانی مانند moderate و unhealthy در زیر نتایج اجرای مدل بهینه شده را مشاهده می‌کنید:



#### ۴. آموزش مدل پایه و مقایسه عملکرد:

برای ارزیابی مزایای بهینه‌سازی، یک مدل SVM استاندارد نیز با مقادیر پیش‌فرض پارامترها آموزش داده شد. دو مدل (پایه و بهینه‌شده) بر روی یک مجموعه تست مشترک ارزیابی شدند. گزارش مقایسه‌ای نشان داد که مدل بهینه‌شده عملکرد بهتری نسبت به مدل پایه دارد، بهویژه از نظر دقیقیت کلی، توان تفکیک کلاس‌های بحرانی، و کاهش خطاهای طبقه‌بندی اشتباه. ماتریس درهم‌ریختگی مدل بهینه‌شده نیز توزیع همگون‌تری از پیش‌بینی‌ها را نسبت به مدل پایه نشان داد. در زیر نتایج اجرای مدل پایه را جهت مقایسه مشاهده می‌کنید:



## ۵. تحليل تفاوت عملکرده، پيچيدگي زمانی و همکاری بین اجزاء:

در اين مرحله، تحليل عميقتری از تفاوت بین مدل‌ها انجام شد. دقت کلی، زمان آموزش، و میزان پیچیدگی محاسباتی به عنوان معیارهای اصلی مقایسه در نظر گرفته شدند. مدل بهینه‌شده با DE دقت بالاتری در طبقه‌بندی سطوح کیفیت هوا را دارد، اما زمان آموزش آن به طور متوسط بیشتر است. این افزایش زمان ناشی از اجرای الگوریتم بهینه‌سازی بر روی فضای پارامترها پیش از آموزش نهایی مدل بود. با این حال، به دلیل ارتقاء قابل توجه عملکرد، این افزایش زمان قابل توجیه است.

```
Optimized accuracy: 0.9926, training time: 1.10s
Standard accuracy: 0.9724, training time: 2.63s
Optimized model has better accuracy but takes longer to train.
```

## ۶. تحليل عملکرده الگوریتم DE و مقایسه با PSO و GA:

Differential Evolution یکی از الگوریتم‌های تکاملی مبتنی بر جمعیت است که از سه عملیات جهش، ترکیب و انتخاب برای جستجوی سراسری در فضای پارامترها بهره می‌برد. برخلاف الگوریتم (GA)، الگوریتم DE از بردارهای حقیقی استفاده می‌کند که سبب کاهش پیچیدگی محاسباتی و بهبود همگرایی می‌شود. همچنین، در مقایسه با (PSO)، الگوریتم DE از پایداری Particle Swarm Optimization (PSO) پیشتری در مواجهه با نقاط بینی محلی برخوردار بوده و برای مسائل با فضای جستجوی پیوسته و کم‌بعد مانند بهینه‌سازی SVM بسیار مناسب است. در این پیاده‌سازی، DE توانست مقادیر مؤثری برای پارامترهای SVM پیدا کند که منجر به افزایش دقت پیش‌بینی و کاهش خطای طبقه‌بندی سطح کیفیت هوا شد. بنابراین، در مقایسه با روش‌های GA و PSO، الگوریتم DE در این کاربرد خاص، انتخابی قابل اتناکا و موثر به شمار می‌رود.

در این بخش، هدف بررسی نحوه عملکرد الگوریتم (DE) Differential Evolution (DE) و مقایسه آن با سایر الگوریتم‌های بهینه‌سازی مانند Genetic Algorithm (GA) و Particle Swarm Optimization (PSO) Algorithm (GA) در پیاده‌سازی DE است. این الگوریتم یکی از روش‌های مبتنی بر جمعیت برای جستجوی سراسری در فضای پارامترهای مدل‌های مانند SVM، کارایی بالایی دارد.

الگوریتم DE با استفاده از عملیات جهش (mutation) و هم‌جفت‌سازی (crossover) بین اعضای جمعیت، فضای جستجو را به صورت گسترده‌ای کاوش می‌کند و از گرفتار شدن در کمینه‌های محلی جلوگیری می‌نماید. در مقایسه با الگوریتم GA، الگوریتم DE به جای نمایش دودویی، از نمایش شناور (floating-point) استفاده می‌کند که باعث سادگی و سرعت بیشتر در فرآیند بهینه‌سازی می‌شود. همچنین، DE نسبت به PSO رفتار کاوش گرانه‌تری دارد و حساسیت کمتری نسبت به پارامترهای اولیه از خود نشان می‌دهد.

ویژگی‌های بر جسته الگوریتم DE که آن را برای تنظیم پارامترهای مدل SVM مناسب می‌سازد عبارت‌اند از: توانایی بالا در جلوگیری از گیر افتادن در کمینه‌های محلی و یافتن جواب‌های بهینه سراسری؛ عملکرد مؤثر در مسائل با فضای پیوسته و کم‌بعد مانند بهینه‌سازی پارامترهای  $C$  و  $\gamma$  در مدل SVM؛ سادگی پیاده‌سازی و عدم نیاز به مشتق‌پذیریتابع هدف، که در مسائل واقعی بسیار سودمند است.

در مجموع، استفاده از DE در این پروژه به عنوان یک الگوریتم پایه برای بهینه‌سازی، انتخاب مناسبی بوده و با ترکیب آن با سایر الگوریتم‌ها مانند DGFO-SVM کمک چشمگیری کرده است. Gravitational Search و Fireworks

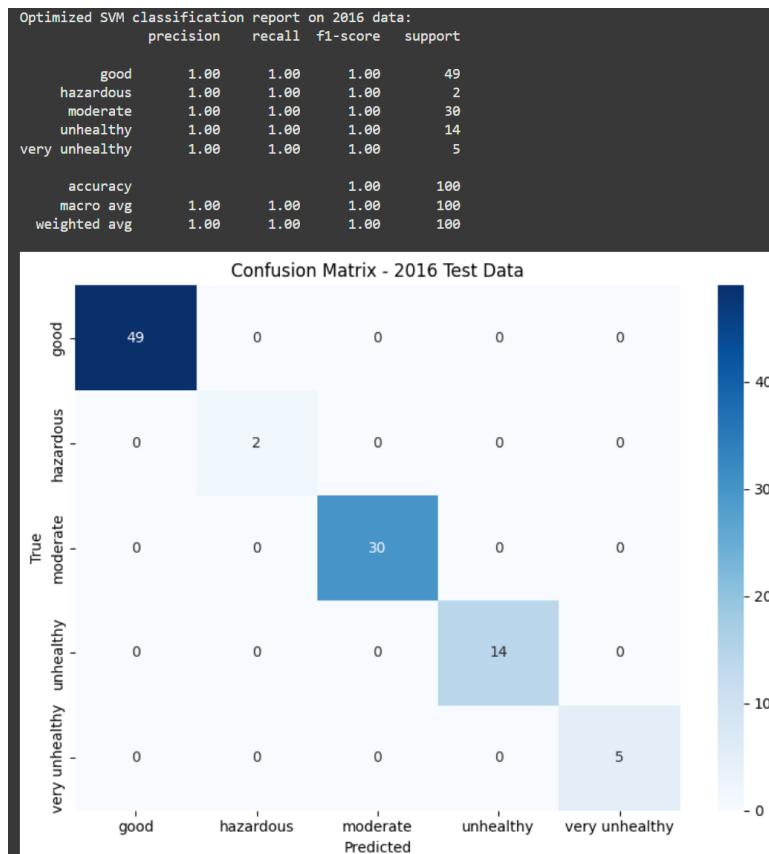
## ۶.۳.۱

## محصول نهایی

در این بخش، هدف ایجاد و ارزیابی یک مدل نهایی و کاربردی بر اساس تحلیل‌های پیشین است. مدل انتخاب شده از نوع SVM بوده که به کمک الگوریتم Differential Evolution برای تنظیم بهینه ابزارهای حساس آن، یعنی  $C$  و  $\gamma$ ، بهینه‌سازی شده است. مقادیر نهایی بدست‌آمده برای این پارامترها به ترتیب  $C = 91.7444$  و  $\gamma = 0.0345$  بوده‌اند.

مدل نهایی تنها با استفاده از داده‌های مربوط به سال‌های ۲۰۱۳ تا ۲۰۱۵ آموزش داده شده و در فرآیند آموزش با ارزیابی اولیه، هیچ‌گونه اطلاعی از داده‌های سال ۲۰۱۶ نداشته است. این جداسازی زمانی در داده‌ها، امکان ارزیابی دقیق‌تری از قدرت تعیین مدل روی داده‌های واقعی و نادیده را فراهم می‌سازد. از آنجا که به سرویس‌های داده‌محور زنده مانند OpenWeather دسترسی وجود نداشت، برای ارزیابی نهایی از داده‌های ثبت‌شده مربوط به سال ۲۰۱۶ استفاده شد. این داده‌ها پس از حذف مقادیر گمشده و حذف پرتهای به صورت تصادفی انتخاب شده و به ۱۰۰ نمونه کاهش یافتند. سپس بر اساس مقدار PM2.5 سطح کیفیت هوای پنج کلاس good, unhealthy, moderate, very unhealthy و hazardous دسته‌بندی گردید. ویژگی‌های ورودی شامل CO, NO2, SO2, PM10, PM2.5 و O3 بوده و مشابه با مرحله آموزش، داده‌ها با StandardScaler نرمال‌سازی شدند. سپس مدل بهینه SVM بدون نیاز به آموزش مجدد، روی این داده‌های واقعی آزمایش شد.

نتایج حاصله بسیار چشم‌گیر بودند: مدل توانست با دقت کامل (Accuracy = 100%) تمامی ۱۰۰ نمونه را به درستی دسته‌بندی کند. بررسی Confusion Matrix نیز نشان داد که هیچ خطای در پیش‌بینی‌ها وجود نداشت. این موضوع نشان‌دهنده قدرت بالای مدل در تشخیص صحیح کلاس کیفی هوای، حتی در شرایط زمانی متفاوت با داده‌های آموزشی است.



با این حال، لازم به ذکر است که این مدل صرفاً برای مسئله دسته‌بندی طراحی و آموزش داده شده است. بنابراین در صورتی که هدف پیش‌بینی دقیق مقدار عددی PM2.5 باشد (یعنی استفاده از مدل در قالب regression)، ممکن است عملکرد به همین خوبی نباشد. این موضوع طبیعی است، چرا که مدل بر



پایه مرازهای طبقه‌بندی آموزش دیده است، نه یادگیری مقادیر عددی دقیق.

در نتیجه می‌توان گفت که مدل نهایی SVM با پارامترهای  $C = 91.7444$  و  $\gamma = 0.0345$  توانسته عملکردی بسیار دقیق در دسته‌بندی کیفیت هوا بر اساس داده‌های واقعی نشان دهد و در صورت دسترسی به داده‌های زنده، قابلیت استفاده در سیستم‌های پیش‌بینی بلادنگ را دارد. با این حال، در صورت نیاز به پیش‌بینی مقادیر آلودگی، باید از مدل‌های رگرسیونی مجزا استفاده گردد.



## ۷.۳.۱

## بخش امتیازی (تحقیقاتی)

۱. جدیدترین روش‌های ارزیابی عملکرد مدل‌های SVM در دسته‌بندی و پیش‌بینی: مدل‌های SVM در هر دو حوزه‌ی classification (دسته‌بندی) و regression (پیش‌بینی پیوسته) کاربرد دارند. بسته به نوع مسئله، روش‌های ارزیابی متفاوتی مورد استفاده قرار می‌گیرند.

در حالت دسته‌بندی، معیارهای پرکاربرد شامل Accuracy، Precision، Recall، F1-Score، AUC-ROC، Confusion Matrix هستند.

Accuracy ساده‌ترین معیار ارزیابی است و درصد کل پیش‌بینی‌های صحیح را نشان می‌دهد. در داده‌های متوازن مناسب است، اما در صورت وجود عدم توازن میان کلاس‌ها، ممکن است عملکرد مدل را بدروستی نشان ندهد.

Precision نسبت نمونه‌های مثبت درست پیش‌بینی شده به کل پیش‌بینی‌های مثبت را می‌سنجد. این معیار در موقعی اهمیت دارد که هزینه‌ی مثبت کاذب زیاد است، مانند تشخیص ایمیل‌های اسپم.

Recall نسبت نمونه‌های مثبت درست پیش‌بینی شده به کل نمونه‌های مثبت واقعی را نشان می‌دهد. در کاربردهای مانند تشخیص بیماری، که از دست دادن موارد مثبت خطرناک است، این معیار اهمیت بیشتری دارد.

F1-Score میانگین هارمونیک precision و recall است و در داده‌های نامتوازن عملکرد مناسبی دارد، چراکه تعادلی میان خطاهای نوع اول و دوم برقرار می‌کند.

AUC-ROC سطح زیر منحنی Receiver Operating Characteristic را نشان می‌دهد و توانایی مدل را در تمایز بین کلاس‌ها در آستانه‌های مختلف اندازه‌گیری می‌کند. این معیار مستقل از آستانه تصمیم‌گیری است و برای ارزیابی جامع‌تر بهویژه در مدل‌های با خروجی احتمالاتی به کار می‌رود. Confusion Matrix تعداد پیش‌بینی‌های درست و نادرست برای هر کلاس را نمایش می‌دهد و ابزار بصری مناسبی برای تحلیل خطای مدل در تمامی کلاس‌هاست.

در حالت پیش‌بینی عددی، Root Mean Squared Error، Mean Squared Error (MSE)، Mean Absolute Error (MAE)، RMSE و R<sup>2</sup>-Score هستند.

MSE میانگین مربعات خطای محاسبه می‌کند و به خطاهای بزرگ حساس است. برای مسائلی که دقت عددی بالا اهمیت دارد مناسب است.

RMSE جذر MSE است و خطای در واحد متغیر خروجی بیان می‌کند. برای مسائل فیزیکی و زمانی کاربرد زیادی دارد.

MAE میانگین قدر مطلق خطاهای اندازه‌گیری می‌کند و نسبت به داده‌های پرت مقاوم‌تر از MSE است. در مسائل اقتصادی و پیش‌بینی تقاضا کاربرد گسترشده‌ای دارد.

R<sup>2</sup> درصد واریانس توضیح داده‌شده توسط مدل را مشخص می‌کند و معیار خوبی برای سنجش توان مدل در مدل‌سازی خطی است.

## ۲. شرایط مناسب برای هر معیار:

در داده‌های کاملاً متوازن، معیار Accuracy بهترین عملکرد را دارد. اما در شرایطی که داده‌ها نامتوازن باشند، مانند تشخیص بیماری‌های نادر یا تشخیص نفوذ امنیتی، معیارهای Recall و F1-Score اهمیت بیشتری دارند. اگر کاهش خطای نوع اول اولویت داشته باشد، Precision باید در اولویت قرار گیرد. برای تحلیل کلی مدل و مقایسه آن‌ها در آستانه‌های مختلف، AUC-ROC انتخاب مناسبی است. در مسائل رگرسیونی، اگر هدف حساسیت به خطای بزرگ باشد MSE و RMSE مفید هستند، ولی در کاربردهای عملیاتی با نویز بالا MAE انتخاب بهتری خواهد بود.

## ۳. بررسی ترکیب مدل‌های SVM با روش‌های بهینه‌سازی:

SVM بهشت به تنظیم دقیق ابرپارامترهای C و gamma وابسته است. این پارامترها عملکرد مدل را در تعادل میان پیچیدگی و دقت کنترل می‌کنند. روش‌هایی مانند Random Search یا Grid Search ممکن است زمان بر یا ناکارا باشند. استفاده از الگوریتم‌های بهینه‌سازی هوشمند مانند Bayesian Optimization یا Particle Swarm Optimization، Genetic Algorithm، Differential Evolution در فضای پارامترها را ممکن سازد. این روش‌ها بهویژه در داده‌های نویزی، غیرخطی و نامتوازن می‌توانند منجر به بهبود چشمگیر عملکرد شوند. در این پروژه، استفاده از DE منجر به دستیابی به مقادیر بهینه C = 91.7444 و gamma = 0.0345 شد که دقت مدل را در داده‌های نادیده به صورت چشمگیری

افزایش داد.

#### ۴. پیشنهادات برای بهبود عملکرد مدل با استفاده از روش‌های نوآورانه:

یکی از روش‌های نوین برای بهبود عملکرد مدل SVM ترکیب آن با سایر مدل‌ها در ساختار Ensemble Bagging-SVM یا Boosting-SVM مانند است که باعث افزایش پایداری و کاهش واریانس می‌شود. همچنین می‌توان از استخراج ویژگی با شبکه‌های عصبی کانولوشنی CNN یا رمزگشایی خودکار Recursive Autoencoder استفاده کرد و سپس این ویژگی‌ها را به عنوان ورودی به مدل SVM داد. استفاده از روش‌های انتخاب ویژگی مانند Feature Elimination (RFE) نیز می‌تواند با حذف ویژگی‌های غیرمؤثر، دقت و سرعت مدل را افزایش دهد. ترکیب مدل SVM با مدل‌های درخت تصمیم مانند XGBoost نیز در ساختارهای stacking نتایج قابل توجهی به همراه دارد.

#### ۵. ساختاردهی پیشنهادات و گزارش‌دهی نهایی:

در پایان تحلیل‌ها باید نتایج به صورت جامع ارائه شوند. هر مرحله از پیش‌پردازش مانند حذف مقادیر پرت یا نرمال‌سازی باید تحلیل و مستند شود. معیارهای ارزیابی به صورت عددی، همراه با نمودارهایی مانند ماتریس آشفتگی، منحنی ROC و مقایسه مدل‌های مختلف آورده شود. پیشنهادهای ترکیبی و نوآورانه نیز باید از نظر تئوری و عملی بررسی شوند تا مبنای تصمیم‌گیری‌های آینده قرار گیرند.

#### ۶. ارجاع به مقالات معتبر:

برای پشتیبانی از تحلیل‌های فوق، منابع معتبر پژوهشی زیر پیشنهاد می‌شوند:

- Suykens, J.A., Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3), 293–300. این مقاله به معروفی LS-SVM می‌پردازد که نسخه‌ای بهینه‌شده برای حل سریع مسائل بزرگ است.
- Xue, B., Zhang, M., Browne, W.N., Yao, X. (2014). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4), 606–626. این مقاله مرور جامعی بر روش‌های بهینه‌سازی مبتنی بر محاسبات تکاملی دارد.
- Tang, J., Alelyani, S., Liu, H. (2014). Feature selection for classification: A review. In *Data classification: Algorithms and applications*. این مرجع به بررسی استراتژی‌های انتخاب ویژگی در مسائل دسته‌بندی می‌پردازد.
- Yan, W., et al. (2018). Air quality forecasting based on hybrid deep learning framework. *IEEE Transactions on Industrial Informatics*, 15(12), 6634–6643. این مقاله به استفاده از ترکیب یادگیری عمیق و مدل‌های کلاسیک مانند SVM در پیش‌بینی کیفیت هوای پرداخته است.
- این مقالات چارچوبی علمی و دقیق برای توسعه و تحلیل مدل‌های SVM در کاربردهای ترکیبی، واقعی و مقیاس‌پذیر فراهم می‌کنند.



## ۲ پرسش دوم

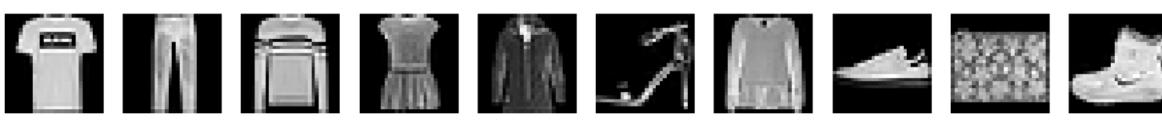
مجموعه داده‌ی MNIST Fashion شامل ۷۰۰۰۰ تصویر سیاه و سفید با اندازه‌ی  $28 \times 28$  از انواع پوشاک در ۱۰ دسته مختلف است.

۱.۲

(الف)

ابتدا مجموعه‌داده FashionMNIST را بارگذاری کرده و از هر یک از ۱۰ کلاس، یک تصویر به صورت دستی انتخاب کردیم. این تصاویر به صورت یک ردیف افقی نمایش داده شدند تا نماینده‌ای از هر دسته در اختیار داشته باشیم. سپس به هر تصویر، نویز گاووسی با میانگین صفر و انحراف معیار ۰.۰۵ اضافه شد. برای جلوگیری از خروج مقادیر پیکسل از محدوده مجاز، با استفاده ازتابع clip مقادیر به بازه  $[0, 1]$  محدود شدند. در نهایت، نسخه‌های نویزی همان تصاویر در یک ردیف جداگانه نمایش داده شدند تا اثر نویز بر کیفیت تصویر به صورت بصری قابل درک باشد.

Original Images



Noisy Images



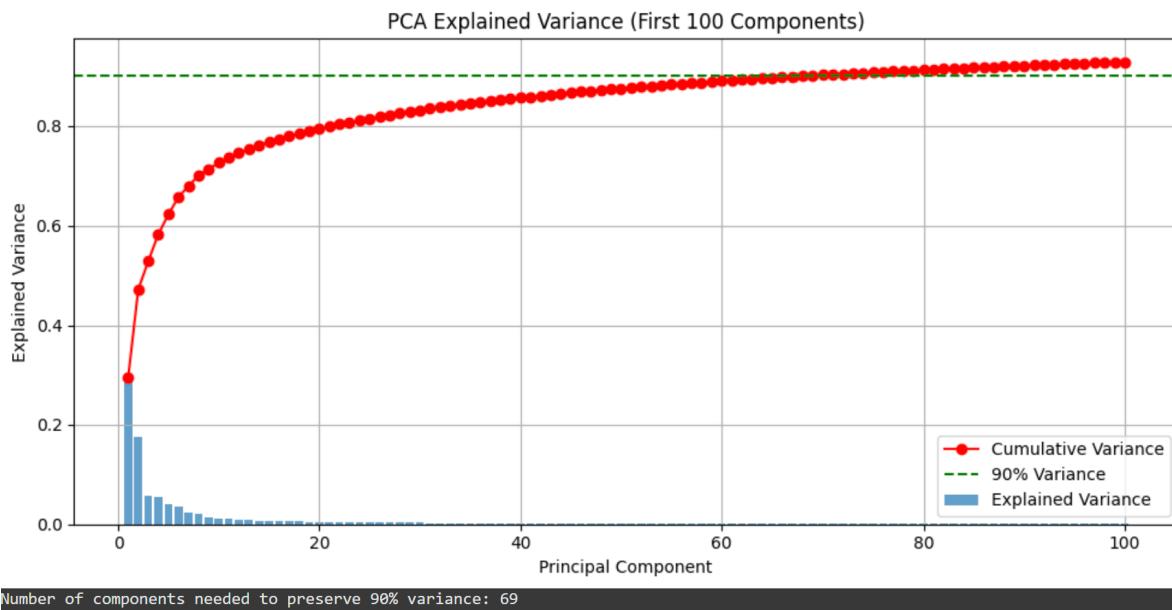
(ب)

تحلیل مؤلفه‌های اصلی (PCA) بدون استفاده از کتابخانه آماده

برای پیاده‌سازی الگوریتم PCA، ابتدا از ۱۰۰۰ تصویر اول مجموعه داده FashionMNIST استفاده کردیم و تصاویر را به آرایه‌ای دو بعدی تبدیل کردیم که هر سطر آن نماینگر یک تصویر و هر ستون آن نماینگر یک پیکسل بود.

در مرحله بعد، میانگین هر ویژگی (پیکسل) را محاسبه کرده و آن را از مقادیر کم کردیم تا داده‌ها نرمال‌سازی شوند. سپس ماتریس کوواریانس ویژگی‌ها محاسبه شد و مقادیر ویژه و بردارهای ویژه استخراج گردید. این مقادیر بر اساس مقدار ویژه مرتباً شدنده تا مهم‌ترین مؤلفه‌ها در اولویت قرار گیرند. نسبت واریانس توضیح‌داده شده برای هر مؤلفه اصلی با تقسیم مقدار ویژه هر مؤلفه بر مجموع مقادیر ویژه به دست آمد. نمودار میله‌ای این نسبت‌ها به همراه واریانس تجمعی رسم شد.

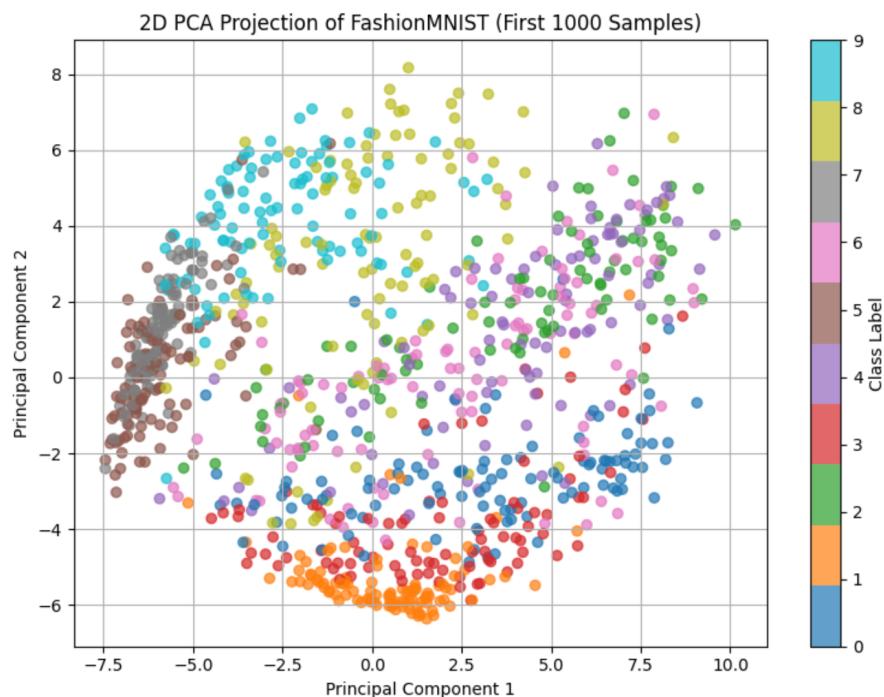
طبق نتایج، با در نظر گرفتن ۶۹ مؤلفه اول، بیش از ۹۰٪ از واریانس کل داده‌ها حفظ می‌شود. بنابراین، می‌توان از این تعداد مؤلفه برای کاهش بعد داده‌ها بدون از دست رفتن بخش عمده‌ای از اطلاعات استفاده کرد.



ج)

### کاهش بعد با استفاده از PCA و نمایش دو بعدی داده ها

در این بخش، با استفاده از کتابخانه scikit-learn الگوریتم PCA را پیاده سازی کردیم و داده ها را به دو مؤلفه اصلی کاهش دادیم. برای این منظور، از ۱۰۰۰ نمونه اول مجموعه داده FashionMNIST استفاده شد و تصاویر به بردارهای یک بعدی تبدیل شدند. سپس با استفاده از PCA دو مؤلفه اصلی استخراج شده و داده ها در فضای دو بعدی تصویرسازی شدند. در نمودار برآکنده حاصل، هر نقطه نماینده یک تصویر بوده و رنگ آن نشان دهنده کلاس (دسته) مربوطه می باشد. از نگاشت رنگی tab10 برای تفکیک دسته ها استفاده شد تا تفکیک کلاس ها به صورت بصیری قابل تشخیص باشد.



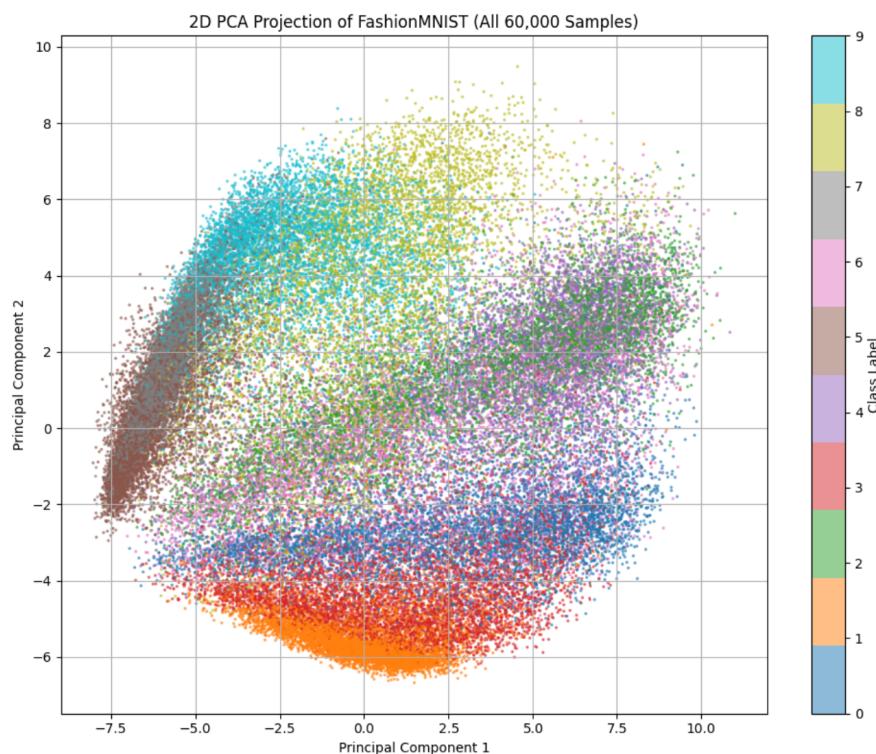


### تحلیل تصویری کل داده‌ها

در این مرحله، بهجای استفاده از زیرمجموعه‌ای از داده‌ها، کل مجموعه آموزشی FashionMNIST شامل ۶۰۰۰۰ تصویر مورد استفاده قرار گرفت. ابتدا تصاویر به بردارهایی از پیکسل تبدیل شده و سپس با استفاده از الگوریتم PCA، بعد داده‌ها به دو مؤلفه اصلی کاهش یافت. نمودار پراکندگی دو بعدی حاصل، نمایش دهنده نگاشتی از داده‌های اصلی در فضای دو بعدی مؤلفه‌های اصلی است. در این نمودار، هر نقطه نشان دهنده یک تصویر است و رنگ آن نمایانگر دسته (کلاس) مربوطه می‌باشد. به کارگیری نگاشت رنگی tab10 باعث شده تا تمایز بین کلاس‌ها به صورت بصری قابل مشاهده باشد.

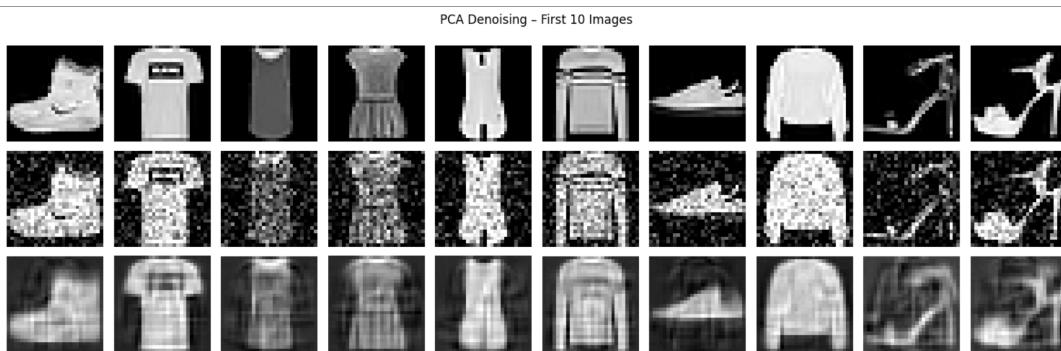
همان‌طور که در نمودار مشخص است، برخی از کلاس‌ها مانند Shirt و Bag هم‌پوشانی بیشتری دارند، در حالی که دسته‌هایی مانند Trouser یا Sandal به صورت خوش‌های تمایزتری دیده می‌شوند. این نمایش دو بعدی اگرچه بخشی از اطلاعات را از دست می‌دهد، اما دید کلی از ساختار و توزیع داده‌ها ارائه می‌دهد که در بسیاری از کاربردهای تحلیل بصری یا پیش‌پردازش می‌تواند بسیار مفید باشد.

Label	0	1	2	3	4	5	6	7	8	9
Class Name	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot



(۵)

برای این بخش، ما از PCA برای فشرده‌سازی و بازسازی نسخه‌های نویزی تصاویر استفاده می‌کنیم. تصاویر نویزی را به زیرفضای ۱۰۰ مؤلفه اصلی نگاشت می‌کنیم، سپس با استفاده از همین مؤلفه‌ها آن‌ها را بازسازی کرده و در کنار نسخه اصلی و نویزی نمایش می‌دهیم.



#### بازسازی کامل تصاویر نویزی با استفاده از PCA

در این بخش، بازسازی تصاویر نویزی با استفاده از الگوریتم PCA بر روی کل مجموعه آموزشی FashionMNIST انجام گرفت. ابتدا نویز گاوی با میانگین صفر و انحراف معیار ۰.۲ به تمام تصاویر افزوده شد و سپس این تصاویر به بردارهایی از اندازه ۷۸۴ تبدیل شدند. سپس با استفاده از PCA و انتخاب ۱۰۰ مؤلفه اصلی، فضای ویژگی‌ها به زیرفضای کاهش‌یافته نگاشته شد و نگاشت معکوس جهت بازسازی تصاویر انجام شد. تصاویر بازسازی شده به شکل اولیه ۲۸x۲۸ تبدیل شدند.

برای ارزیابی بصری، برای ۱۰ تصویر اول سه ردیف در نظر گرفته شد: ردیف اول شامل تصاویر اصلی، ردیف دوم تصاویر نویزی، و ردیف سوم تصاویر بازسازی شده. همان‌طور که از نتایج قابل مشاهده است، نسخه‌های بازسازی شده علی‌رغم وجود نویز، شباهت زیادی به نسخه اصلی دارند و بسیاری از جزئیات تصویر حفظ شده‌اند.

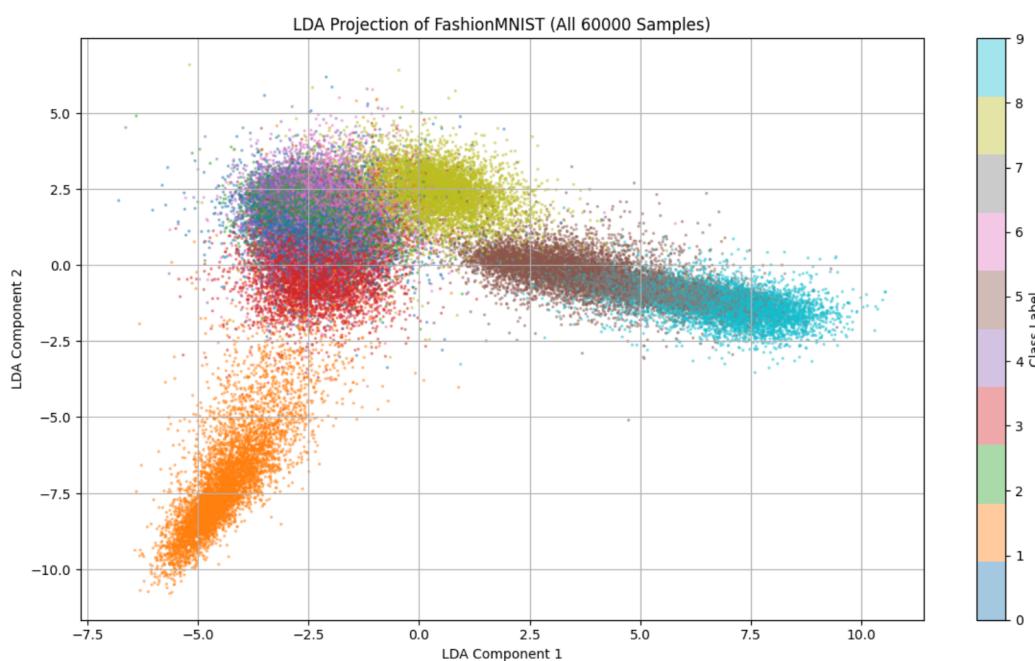
این فرآیند نشان می‌دهد که PCA حتی با فشرده‌سازی داده‌ها به ۱۰۰ مؤلفه می‌تواند ساختار کلی تصویر را حفظ کرده و به عنوان روشی برای کاهش نویز و بازیابی تصویر عمل کند.

### تحلیل کاهش بعد بالگوریتم LDA و مقایسه با PCA

در این بخش، الگوریتم LDA با استفاده از کتابخانه scikit-learn پیاده‌سازی شد. از تمام مجموعه داده FashionMNIST برای این تحلیل استفاده گردید. تصاویر به بردارهایی از طول 784 تبدیل شده و سپس با استفاده از LDA به دو مؤلفه خطی کاهش یافتند.

نمودار حاصل یک نمایش دوبعدی از داده‌ها در فضای مؤلفه‌های LDA است که در آن هر نقطه نمایانگر یک تصویر است و رنگ آن نشان‌دهنده کلاس مربوطه می‌باشد. از نگاشت رنگی tab10 برای تمایز بصری بین کلاس‌ها استفاده شده است.

در مقایسه با نمودار PCA، خروجی LDA تفکیک بهتری بین کلاس‌ها نشان می‌دهد. دلیل این امر آن است که PCA فقط به واریانس کلی داده‌ها توجه دارد، در حالی که از اطلاعات برچسب‌ها برای یافتن جهاتی استفاده می‌کند که بیشترین تمایز بین کلاس‌ها را ایجاد می‌کنند. بنابراین، LDA برای اهدافی مانند طبقه‌بندی یا تحلیل کلاس‌بندی شده عملکرد بهتری نسبت به PCA در نمایش ساختار داده‌ها در فضای کاهش‌یافته دارد.



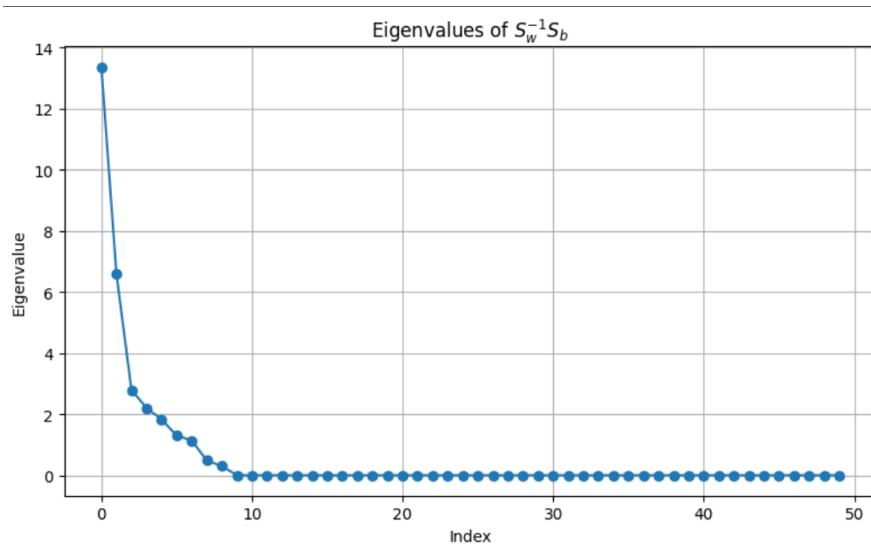
(و)

### محاسبه ماتریس‌های پراکندگی و تحلیل مقادیر ویژه ماتریس تفکیک‌پذیری کلاس‌ها

در این بخش، برای تحلیل قابلیت جداسازی داده‌ها از نظر خطی، ابتدا ماتریس پراکندگی درون‌کلاسی  $S_w$  و بین‌کلاسی  $S_b$  بر اساس داده‌های مجموعه FashionMNIST محاسبه شدند.

ماتریس  $S_w$  با جمع ماتریس کوواریانس مرکز داده‌شده هر کلاس تشکیل شد. در مقابل،  $S_b$  با استفاده از فاصله بردار میانگین‌های هر کلاس نسبت به میانگین کل داده‌ها و ضرب آن در تعداد نمونه‌های آن کلاس ساخته شد. در مرحله بعد، ماتریس  $S_w^{-1}S_b$  تشکیل شد و مقادیر ویژه آن به صورت عددی محاسبه گردید. از این ماتریس برای یافتن جهات تفکیک‌پذیر بین کلاس‌ها در روش LDA استفاده می‌شود.

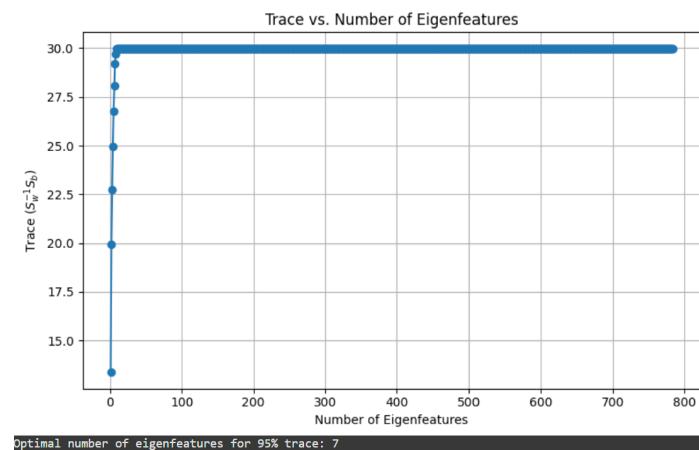
نمودار به دست‌آمده از 50 مقدار ویژه بزرگ‌تر نشان می‌دهد که فقط تعداد محدودی از مؤلفه‌ها دارای مقادیر قابل توجه هستند و بخش بزرگی از ساختار قبل جداسازی داده‌ها در این مؤلفه‌های اولیه متمرکز است. این یافته با تئوری روش LDA که حداکثر  $C - 1$  مؤلفه تفکیک‌پذیر برای  $C$  کلاس ارائه می‌دهد، سازگار است.



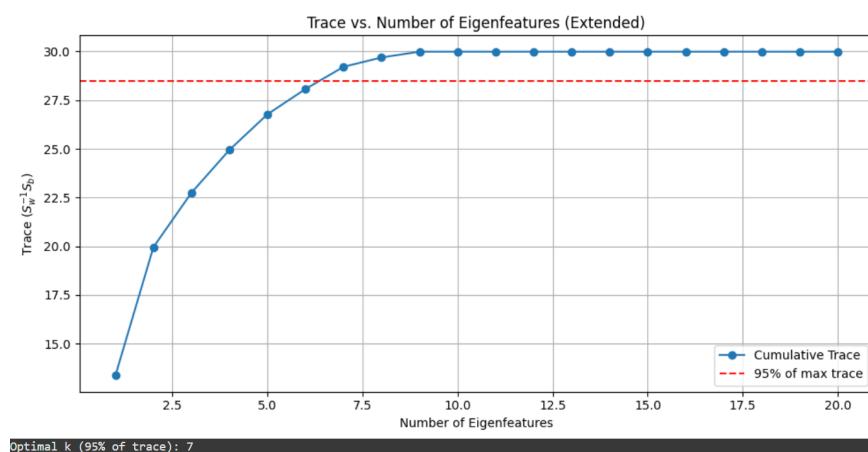
(ز)

### تحلیل مقدار trace( $S_w^{-1}S_b$ ) نسبت به تعداد ویژگی‌ها

در این مرحله، پس از محاسبه ماتریس تفکیک‌پذیری  $S_w^{-1}S_b$ ، مقدار ویژه (یا همان eigenvalue)‌های آن استخراج شدند. سپس با استفاده از مجموع تجمعی eigenvalue، مقدار trace( $S_w^{-1}S_b$ ) برای تعداد ویژگی‌های انتخاب شده محاسبه شد. نموداری با محور افقی "تعداد ویژگی‌ها" و محور عمودی "مقدار trace" رسم شد تا روند رشد این معیار را با افزایش eigenfeature‌ها نشان دهد. همان‌طور که در نمودار مشاهده شد، در ابتدا با افزایش سریع رشد می‌کند، اما پس از تعداد خاصی از مؤلفه‌ها به حالت اشباع نزدیک می‌شود. طبق تحلیل عددی انجام شده، مشاهده شد که استفاده از 7 eigenfeature اولیه کافی است تا 96% از مقدار trace پوشش داده شود. این بدان معناست که می‌توان به جای استفاده از تمام component، تنها با انتخاب 7 مؤلفه اول، تقریباً تمام قابلیت تفکیک‌پذیری بین class‌ها را حفظ کرد. این روش می‌تواند راهنمایی مؤثر برای تعیین تعداد ویژگی‌های بهینه در کاربردهای نظری classification و dimensionality reduction باشد، به ویژه در شرایطی که سرعت و کارایی پردازش اهمیت دارد.



البته مشاهده می‌شود با انتخاب ۹ مؤلفه اول به اشباع میرسیم و چون اختلاف زیادی با ۷ ندارد می‌توان به جای ۷ از مقدار بهینه ۹ نیز استفاده کرد.



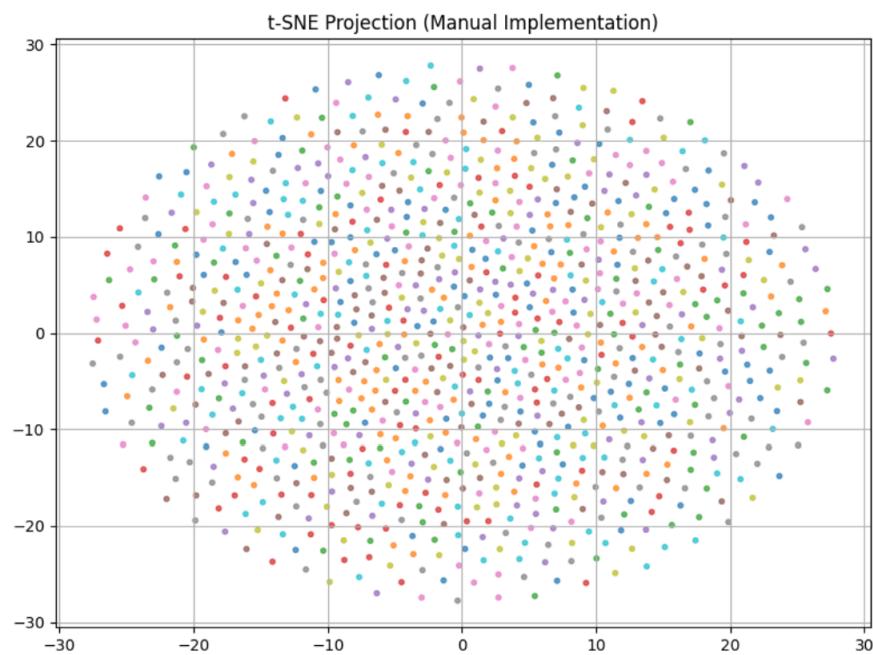
## ۲.۲

## مزایا و مقایسه روش t-SNE با PCA و LDA

روش PCA برخلاف t-SNE که صرفاً به واریانس داده‌ها توجه دارد، و LDA که بر پایه تفکیک بین کلاس‌ها طراحی شده است، بر حفظ ساختار محلی و همسایگی نمونه‌ها تمرکز دارد. همین ویژگی باعث می‌شود که t-SNE برای مصورسازی خوشه‌های پنهان در داده‌ها بسیار مناسب باشد. از مزایای اصلی این روش می‌توان به تولید نقشه‌های دوبعدی با خوشه‌بندی بسیار واضح اشاره کرد. اما در مقابل، هزینه محاسباتی بالاتر و حساسیت به پارامترهای مانند learning rate و perplexity از جمله محدودیت‌های آن است.

همچنین، t-SNE برای یادگیری بدون ناظر بسیار مناسب است ولی برخلاف LDA قابلیت استفاده در طبقه‌بندی مستقیم را ندارد. بهطور خلاصه اگر بخواهیم پپردازیم، t-SNE ابزار قدرتمندی برای مصورسازی داده‌ها در فضاهای پایین‌بعدی است، ولی جایگزینی برای روش‌های کاهش‌بعد خطی مانند PCA یا LDA در کاربردهای تحلیلی دقیق نیست.

پیاده سازی دستی آن به همراه مولفه‌های آن به شکل زیر است، توجه کنید این ساده ترین نمایش است زیرا با نسخه کتابخانه‌ای که پر از بهینه سازی‌ها متفاوت پارامترها و تحلیل‌های ریاضیاتی که برای عدم دریافت پارامترها و نوسان آن‌ها طراحی شده بسیار متفاوت است به هر حال برای این مورد و تعداد ۱۰۰۰ نمونه حاصل به شکل زیر است:





### مروی کامل بر الگوریتم t-SNE و مقایسه آن با سایر روش‌های کاهش بُعد:

الگوریتم t-SNE (t-distributed Stochastic Neighbor Embedding) یک روش کاهش بُعد غیرخطی و بدون نظارت است که برای مصورسازی داده‌های با ابعاد بالا استفاده می‌شود. این الگوریتم تمرکز ویژه‌ای بر حفظ ساختار محلی داده‌ها دارد و برخلاف روش‌هایی مانند PCA، به دنبال حفظ فواصل جهانی بین نقاط نیست بلکه تلاش می‌کند همسایگی‌های نزدیک را در فضای جدید حفظ نماید.

مراحل اصلی این الگوریتم عبارت‌اند از:

ابتدا، برای داده‌های با بعد بالا، فاصله اقلیدسی بین هر جفت نقطه محاسبه می‌شود و سپس این فاصله‌ها به احتمال تبدیل می‌شوند. احتمال اینکه نقطه  $i$  همسایه نقطه  $j$  باشد با استفاده از یک توزیع Gaussian تعریف می‌شود که دارای واریانس  $\sigma_i^2$  است. برای انتخاب مقدار بهینه  $\sigma_i$ ، از تکنیک binary search استفاده می‌شود تا perplexity (که بیانگر اندازه مؤثر همسایگی است) برابر مقدار مطلوب انتخاب شده (مثلاً 30) شود.

پس از محاسبه احتمالات شرطی  $P_{ij}$ ، آن‌ها به یک توزیع احتمال مشترک متقارن  $P_{ij}$  تبدیل می‌شوند:

$$P_{ij} = \frac{P_{j|i} + P_{i|j}}{2N}$$

در مرحله بعد، نقاط داده‌ها در فضای کم‌بعد (معمولًاً دو بعدی) با مقادیر تصادفی اولیه مقداردهی می‌شوند. در این فضا، شباهت بین نقاط با استفاده از توزیع Student-t با درجه آزادی 1 (که دارای دنباله‌های سنگین است) مدل می‌شود:

$$Q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

هدف الگوریتم t-SNE این است که توزیع  $Q$  در فضای کم‌بعد تا حد ممکن شبیه توزیع  $P$  در فضای با بعد بالا باشد. برای این کار ازتابع زیان مبتنی بر Kullback-Leibler divergence می‌شود:

$$KL(P\|Q) = \sum_{i \neq j} P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

گرادیان این تابع نسبت به موقعیت‌های  $y_i$  محاسبه شده و با استفاده از روش gradient descent بهینه‌سازی می‌شود. برای بهبود همگرایی، از تکنیک‌های learning rate scheduling و early exaggeration، momentum و early exaggeration استفاده می‌شود.

### متغیرهای کلیدی در t-SNE

- Perplexity: کنترل کننده اندازه مؤثر همسایگی. معمولًاً بین 5 تا 50 تنظیم می‌شود. - Learning rate: نرخ یادگیری که اگر خیلی کم باشد همگرایی کند خواهد بود و اگر زیاد باشد به نتایج ناپایدار منجر می‌شود. - Early exaggeration: در تکرارهای ابتدایی برای بزرگنمایی فواصل استفاده می‌شود تا خوش‌ها بهتر از هم جدا شوند. - KL divergence: معیار اختلاف بین دو توزیع احتمال که به عنوان تابع هزینه به کار می‌رود. - Y: نگاشت کم‌بعد نهایی داده‌ها که به روزرسانی می‌شود تا  $KL(P\|Q)$  کمینه شود.

در نهایت، یک ابزار قدرتمند برای مصورسازی ساختار درونی داده‌هاست، به ویژه در داده‌های پیچیده و غیرخطی، اما نیازمند تنظیم دقیق پارامترهاست و به شدت نسبت به آن‌ها حساس است. همچنین هزینه محاسباتی بالا دارد و خروجی آن در صورت نبود مقدار ثابت برای random seed تکرارپذیر نیست.

### مقایسه با سایر روش‌ها: (PCA، LDA)

پس از اجرای کامل و مقایسه‌ای سه روش کاهش بعد یعنی مدل‌های مبتنی بر بدون نظارت (PCA، t-SNE) و با نظارت (LDA)، به نتایج کاربردی و تحلیلی رسیدیم که می‌تواند در انتخاب روش مناسب برای نوع داده و هدف کاربردی مفید باشد.

**Mزايا و معایب PCA (Principal Component Analysis):** یک روش کاهش بعد خطی و بدون نظارت است که تغییرپذیری را در مشخصه‌ها حفظ می‌کند. بسیار سریع، قابل فهم و مناسب برای فشرده‌سازی داده‌هاست. اما قادر نیست روابط غیرخطی را مدل کند و با داده‌های با ساختار پیچیده عملکرد مناسبی ندارد.

**مزایا و معایب LDA**: LDA روشی خطی با نظرات است که برای جداسازی کلاس‌ها قدرتمند است. تفاوت بین کلاسی را مشخصاً هدف قرار می‌دهد. اما فرض می‌کند توزیع‌ها normal هستند و تعداد مولفه‌ها به تعداد کلاس‌ها محدود می‌شود. در موقعي که داده‌ها بحسب دارند و جداسازی کلاس، مدنظر است، بسیار مؤثر و تفسیر یزیر است.

**t-SNE**: برای مصورسازی داده‌های پیچیده و غیرخطی بسیار مناسب است و نتایجی بسیار شهودی و قابل تفسیر ارائه می‌دهد. با این حال، هزینه محاسباتی بالا دارد، نتایج آن در صورت عدم تنظیم random seed پایدار نیست و به شدت به پارامترهایی مانند perplexity و learning rate حساس است. همچنین برای کاربردهایی مانند فشردهسازی داده‌ها یا یادگیری ویژگی‌ها مناسب نیست و صرفاً برای مصورسازی کاربرد دارد. در نتیجه، انتخاب بین PCA، LDA و t-SNE بستگی کامل به نوع مسئله، هدف کاربردی، ساختار داده و وجود برچسب‌ها دارد. برای فشردهسازی و سرعت بالا PCA مناسب است، برای طبقه‌بندی با نظارت LDA و برای مصورسازی ساختارهای غیرخطی پیچیده t-SNE استفاده از t-SNE در کتابخانه‌ها و تفاوت آن با پیاده‌سازی دستی

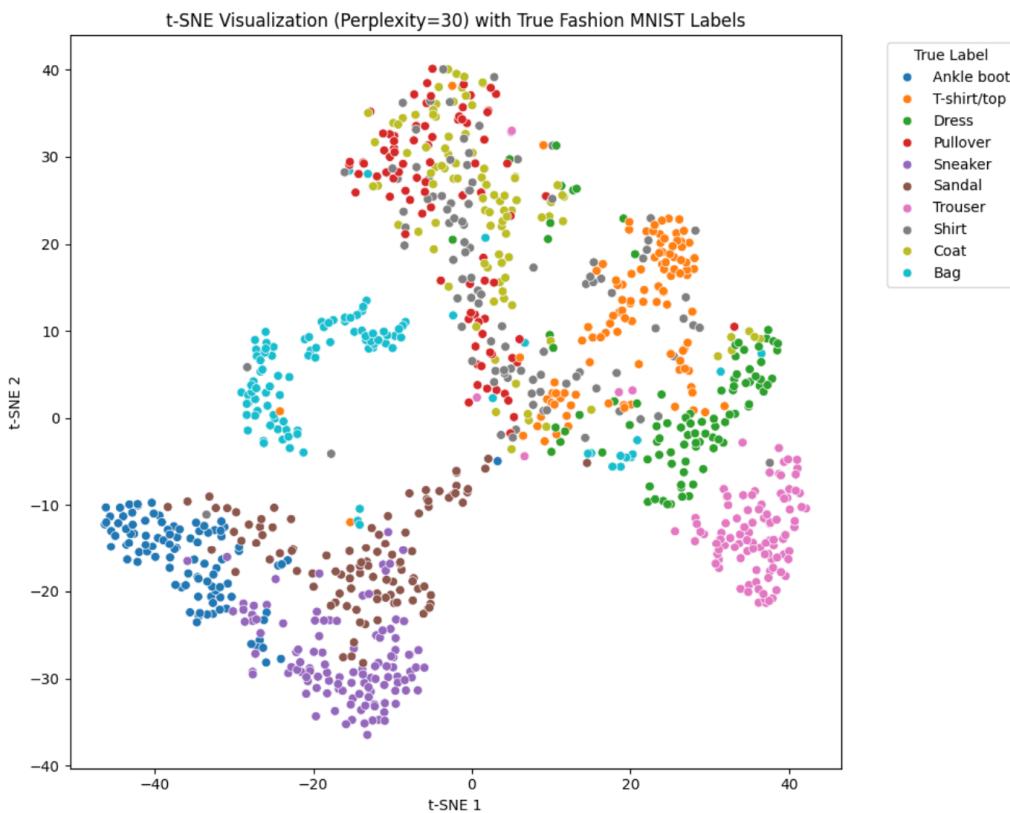
روش t-SNE در کتابخانه‌ای مانند scikit-learn به صورت بهینه و دقیق پیاده‌سازی شده است و از تکنیک‌هایی استفاده می‌کند که در پیاده‌سازی دستی به سادگی قابل اجرا نیستند. به همین دلیل، خروجی این کتابخانه‌ها معمولاً بسیار واضح‌تر، تفکیک‌پذیر‌تر و شهودی‌تر از نسخه‌های پیاده‌سازی شده به صورت دستی است.

در کتابخانه scikit-learn، الگوریتم t-SNE ابتدا فاصله‌های اقلیدسی را در فضای اصلی داده محاسبه می‌کند، سپس با استفاده ازتابع چگالی گاووسی، توزیع احتمال شباهت بین نقاط را در فضای با بعد بالا می‌سازد. در مرحله بعد، در فضای کاوش‌یافته (معمولًاً دوبعدی)، توزیع  $\text{KL divergence}$  به کار گرفته می‌شود و با استفاده از کمینه‌سازی KL divergence بین توزیع اولیه و توزیع نهایی، نقاط به گونه‌ای نگاشت می‌شوند که ساختار همسایگی اولیه حفظ شود.

یکی از عوامل مهم در تفاوت نتایج خروجی کتابخانه‌ای با پیاده‌سازی ساده، تنظیمات و بهینه‌سازی‌های عددی مانند: early exaggeration،

در آزمایش انجام شده با داده های FashionMNIST، خروجی t-SNE کتابخانه ای توانست به خوبی کلاس های مختلف مانند T-shirt, Bag, Sneaker و سایر موارد را به صورت نواحی متمایز در فضای دوبعدی تفکیک کند، در حالی که پیاده سازی ساده دستی نتوانست چنین خوش بندی واضحی ایجاد کند و اغلب به الگوهای به معنا یا خوشه های در هم ریخته منتهی شد.

نتیجه‌گیری مهم این است که برای کاربردهای واقعی یا تحقیقاتی، استفاده از پیاده‌سازی کتابخانه‌ای t-SNE توصیه می‌شود، زیرا از نظر کیفیت، سرعت، بهینه‌سازی و قابلیت اطمینان در سطح بسیار بالاتری قرار دارد. همچنین با تغییر پارامترهایی مانند learning rate، perplexity، و تعداد تکرارها می‌توان



### تحلیل خروجی t-SNE بر اساس نمودار بصری خوشه‌ها

با اعمال الگوریتم t-SNE بر داده‌های FashionMNIST و مصورسازی خروجی دو بعدی آن، مشاهده شد که برخی کلاس‌ها به خوبی از یکدیگر جدا شده‌اند، در حالی که برخی دیگر در فضاهای مشترک یا نزدیک به هم قرار گرفته‌اند.

به طور خاص، کلاس‌های مرتبط با footwear شامل Ankle boot و Sandal در نواحی مجاور یکدیگر قرار گرفته‌اند. این نزدیکی به دلیل شباهت‌های ساختاری و بصری بین تصاویر کفش‌هاست، اما همچنان با دقت قابل قبولی قابل تفکیک هستند.

همچنین کلاس‌های Trouser و Dress نیز به هم نزدیک هستند. دلیل این امر آن است که در بسیاری از تصاویر، این دو نوع لباس از نظر فرم، طول یا بافت شباهت‌هایی دارند که الگوریتم آن‌ها را در فضای کم بعد نزدیک به هم نگاشت می‌دهد.

در مقابل، کلاس‌های مربوط به لباس‌های بالاتنه که Shirt، Pullover و Coat ناحد زیادی در هم ادغام شده‌اند و خوشه‌های متمايزی تشکیل نداده‌اند. این موضوع به دلیل شباهت زیاد تصاویر این دسته‌ها از نظر ساختار، یقه، آستین و شکل کلی لباس است. در تصاویر خاکستری مقیاس FashionMNIST، تفاوت این لباس‌ها ممکن است برای مدل در ابعاد پایین‌مرتبه به راحتی قابل تمایز نیاشد.

با این حال، کلاس Bag کاملاً از سایر کلاس‌ها جدا شده و خوشه‌ای مستقل تشکیل داده است. این امر نشان‌دهنده‌ی تفاوت بارز این دسته از نظر ظاهر و ساختار نسبت به سایر کلاس‌هاست و الگوریتم توانسته است این تفاوت را به خوبی شناسایی و بازنمایی کند.

همچنین برخلاف برخی کلاس‌های بالاتنه که با یکدیگر همپوشانی دارند، کلاس T-shirt/top در ناحیه‌ای دورتر از Shirt، Pullover و Coat قرار گرفته است. این موضوع بیانگر آن است که تفاوت‌های تصویری متمايزی دارند که آن‌ها را از سایر لباس‌های بالاتنه جدا می‌سازد.

در مجموع، الگوریتم t-SNE توانسته است تا حد زیادی ساختار محلی و شباهت‌های بصری بین کلاس‌ها را بازتاب دهد و کلاس‌هایی که از لحاظ ظاهری شباهت دارند را به صورت خوشه‌های نزدیک‌تر نمایش دهد. در مواردی نیز، کلاس‌هایی که تفاوت ظاهری بارزی با سایر دسته‌ها دارند، مانند Bag با موفقیت به طور کامل از دیگر کلاس‌ها تفکیک شده‌اند. با این حال، در مواجهه با دسته‌هایی که از لحاظ ظاهری بسیار مشابه هستند، مثل لباس‌های بالاتنه، تفکیک کامل ممکن نیست و نیازمند استفاده از ویژگی‌های سطح بالاتر یا داده‌های رنگی و با کیفیت‌تر می‌باشد.