به نام خدا

- مدل پیش‌بینی آلودگی هوای تهران
- امیرحسین مسعودنژاد_95102298
- Github

0. مسئله ما ساختن مدلی برای پیش‌بینی کیفیت هوای تهران است. اگرچه می‌توان این مسئله را به روش طبقه‌بندی نیز حل نمود (مثلا کلاس‌های ما شامل هوای پاک، سالم، ناسالم و... باشند)، من تصمیم گرفتم این مسئله را به‌صورت regression و به نحوی که در ادامه توضیح خواهم داد حل کنم. احتمالا با این کار چالش بیشتری برای ساخت مدل خواهم داشت؛ اما به هر حال تلاش خود را خواهم کرد و امیدوارم نتیجه قابل قبولی بگیرم.

1. داده‌ها

برای ساخت چنین مدلی ابتدا باید در نظر داشت چه عواملی موثرتر هستند و بر خروجی ما تاثیر مشهودتری دارند. طبیعتا اولین عوامل موثر بر کیفیت هوا که به ذهن می‌رسند وضعیت آب و هوا می‌باشد. برای تهیه این داده‌ها ابتدا به سایت سازمان هواشناسی تهران سر زدم. از یافتن داده‌ها بسیار شادمان بودم که با قیمت 103 میلیون تومانی داده‌هایی که می‌خواستم مواجه شدم! البته دریافت داده‌ها به صورت دانشجویی و رایگان هم ممکن بود؛ اما مدارکی میخواست که من نداشتم (مثل معرفی‌نامه از دانشگاه) و اینکه باید مدارکم بررسی میشد و در آخر داده‌ها را دریافت میکردم. در نهایت به دلیل اینکه میترسیدم داده‌ها سر موقع به دستم نرسند و برای انجام این کار دیر بشود از این کار صرف نظر کردم.

پس از چند ساعتی جست‌وجو در گوگل سایت *Underground Weather* را پیدا کردم که داده‌های هواشناسی تهران را رایگان داشت. با اینکه داده‌های هر ماه را جدا و به صورت جدول آنلاین داشت، بسیار خوب بود. داده‌ها را در Excel کپی کرده و به صورت csv ذخیره کردم. داده‌های این سایت شامل min, max, avg برای متغیرهای دما $T(°\mathrm{F})$، رطوبت $H(\%)$، فشار $P(mmHg)$، سرعت باد $W(mph)$ و نقطه شبنم $D(°\mathrm{F})$ بود.

با اینکه داده‌های خوبی به‌دست آورده بودم، اما هنوز یک داده مهم و تاثیرگذار را در اختیار نداشتم، «میزان بارش». در نهایت ناچار شدم این داده‌ها $(Rain)$ را به صورت آزاد از سایت هواشناسی خریداری کنم. چون با این کار، همان لحظه داده‌ها قابل دانلود خواهند بود. (در حقیقت برای یک دانشجوی مبتلا به OCD و دچار به خشونت نمادینِ چندین ساله، صبر و در انتظار بودن عذاب است.)

عامل تاثیرگذار دیگری که به ذهنم رسید، عامل تعطیلات و روزهای کاری بود. برای این کار با کمک تقویم، به هر یک از روزهای سال در مقیاس 1- تا 2 عددی نسبت دادم؛ به‌طوریکه هرچه به عدد 2 نزدیک‌تر شویم قدرت آن تعطیلی بیشتر می‌شود. مثلا برای اولین روز عید نوروز این عدد 2، برای روزهای کاری عادی 0 و برای روز شلوغ و پرترافیکی مانند اول مهر این عدد 1- است. این ستون از داده‌ها $(HOLS)$ را به صورت دستی وارد جدول کردم. (با درود به حکیم عمر خیام، از پدیدآورندگان تقویم جلالی)

داده‌هایی که تا اینجا ذکر کردم، ورودی ما را به صورت زیر تشکیل می‌دهند :

```
[ ] X.head()
```

	T_Max	T_Avg	T_Min	D_Max	D_Avg	D_Min	H_Max	H_Avg	H_Min	W_Max	W_Avg	W_Min	P_Max	P_Avg	P_Min	Rain	HOLS
1/1/2019 0:00	50	44.0	37	25	20.9	14	60	40.8	30	23	10.4	0	26.2	26.1	26.1	0.00	0.0
1/2/2019 0:00	50	44.3	37	32	22.5	18	66	43.5	28	7	2.6	0	26.2	26.1	26.1	0.00	0.0
1/3/2019 0:00	46	38.8	34	37	33.9	30	93	83.3	66	18	6.7	0	26.1	26.0	26.0	0.61	0.2
1/4/2019 0:00	52	41.0	34	32	27.8	21	87	62.0	30	16	3.0	0	26.2	26.1	26.1	0.00	1.0
1/5/2019 0:00	52	42.2	32	34	28.6	21	81	60.4	33	7	3.5	0	26.2	26.1	26.1	0.00	0.0

برای داده‌های خروجی، همان‌طور که عرض کرده بودم، قصد دارم یک مسئله رگرسیون حل کنم؛ پس تصمیم گرفتم خروجی را شاخص کیفیت هوا (AQI) در نظر بگیرم. برای تهیه این داده به سایت کنترل کیفیت هوای تهران سر زدم. خوشبختانه آن‌ها از سازمان هواشناسی بخشنده‌تر بودند و داده‌های خود را رایگان در اختیار عموم قرار می‌دادند.

چون داده‌های هواشناسی متعلق به ایستگاه مهرآباد تهران بود، داده‌های آلودگی هوا را از ایستگاه شریف انتخاب کردم که نسبتا نزدیک هم باشند. داده‌های کیفیت هوا چندین خروجی به شرح زیر داشت :

	وضعیت آلودگی هوا به لحاظ آلاینده و به تفکیک روز												
2	ایستگاه	تاریخ	CO	O3	NO2	SO2	PM10	PM2.5	AQI	پاک	قابل قبول	ناسالم برای گروه حساس	ناسالم	بسیار ناسالم	خطرناک
3	جمع کل	-								15	168	157	24		
4	شریف_منطقه2	1397/10/11	18	23	120	7	57	77	77		1				
5	شریف_منطقه2	1397/10/12	42	16	117	11		86	86		1				
6	شریف_منطقه2	1397/10/13	32	3	124	13	71	130	130			1			
7	شریف_منطقه2	1397/10/14	29	8	108	9	38	91	91		1				
8	شریف_منطقه2	1397/10/15	44	17	119	10	61	115	115			1			
9	شریف_منطقه2	1397/10/16	27	11	112	16	77	152	152				1		
10	شریف_منطقه2	1397/10/17	30	13	115	12	68	131	131			1			
11	شریف_منطقه2	1397/10/18	27	24	111	6	77	77	77		1				
12	شریف_منطقه2	1397/10/19	23	16	110	9	74	98	98		1				
13	شریف_منطقه2	1397/10/20	17	29	103	7	55	63	63		1				
14	شریف_منطقه2	1397/10/21	28	18	107	12	47	62	62		1				
15	شریف_منطقه2	1397/10/22	21	20	102	9	55	83	83		1				
16	شریف_منطقه2	1397/10/23	28	17	114	14	62	108	108			1			
17	شریف_منطقه2	1397/10/24	36	9		16	63	124	124			1			
18	شریف_منطقه2	1397/10/25	30	15	108	16	61	114	114			1			
19	شریف_منطقه2	1397/10/26	77	33	119	12	55	76	77		1				
20	شریف_منطقه2	1397/10/27	17	17	110	10	51	86	86		1				
21	شریف_منطقه2	1397/10/28	14	19	103	5	23	44	44	1					
22	شریف_منطقه2	1397/10/29	22	30	102	7	50	63	63		1				
23	شریف_منطقه2	1397/10/30	25	11	106	11	60	92	92		1				
24	شریف_منطقه2	1397/11/01	28	13	112	14	78	148	148			1			
25	شریف_منطقه2	1397/11/02	28	18	115	11	68	123	123			1			
26	شریف_منطقه2	1397/11/03	31	17	117	15	80	139	139			1			

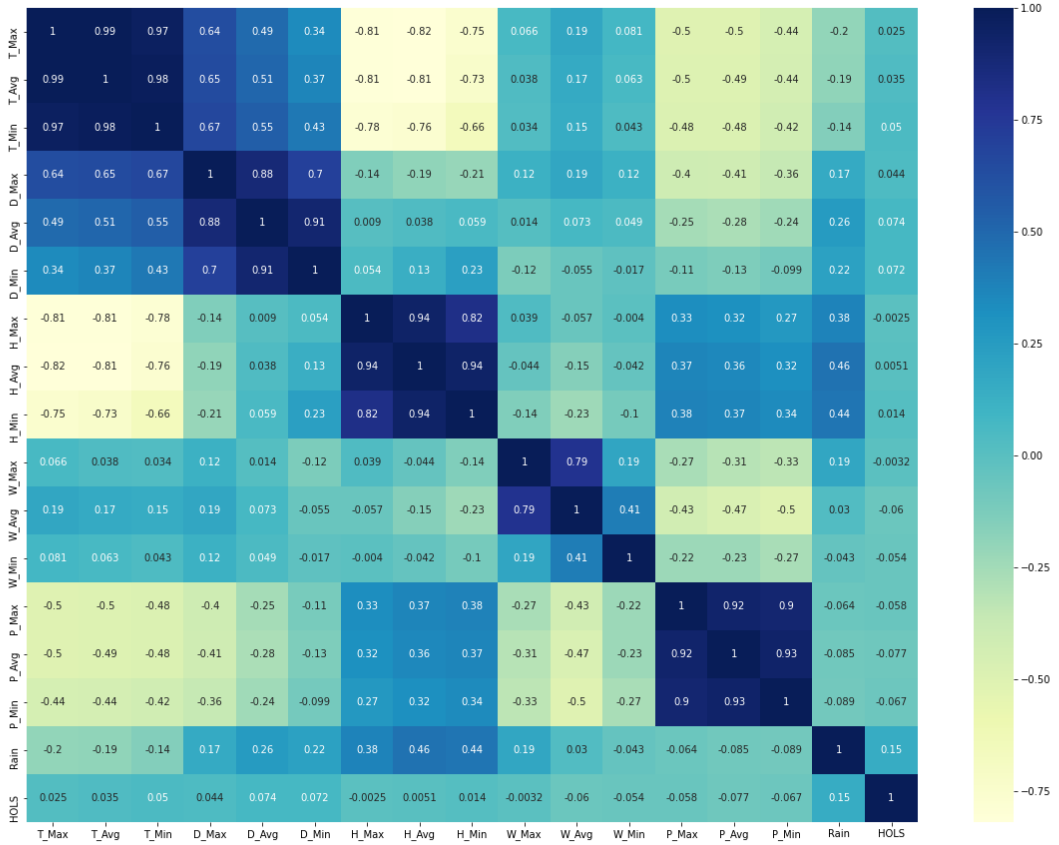
در این لحظه بود که تصمیم گرفتم مسئله را به صورت regression با 7 خروجی (و با تاکید بر AQI) حل کنم.

```
[ ] y.head()
```

	CO	O3	NO2	SO2	PM10	PM2.5	AQI
1397/10/11	18.0	23.0	120.0	7.0	57.0	77.0	77.0
1397/10/12	42.0	16.0	117.0	11.0	NaN	86.0	86.0
1397/10/13	32.0	3.0	124.0	13.0	71.0	130.0	130.0
1397/10/14	29.0	8.0	108.0	9.0	38.0	91.0	91.0
1397/10/15	44.0	17.0	119.0	10.0	61.0	115.0	115.0

این نکته را هم بگویم که داده‌های تهیه شده مربوط به ابتدای سال 2019 تا انتهای آن سال می‌باشند؛ زیرا نمی‌خواستم این داده‌ها وارد روزهای کرونایی و محدودیت‌هایی که به دنبال آن ایجاد شد بشوند. در آن صورت احتمالا مدل ما دچار پیچیدگی‌های جدید و سخت‌تری میشد که البته موقت نیز هستند.

- در داده‌های خروجی تعداد کمی missing data وجود داشت که با استفاده از Iterative Imputer آن را رفع کردم.
- با رسم correlation matrix برای داده‌ها، می‌بینیم همانطور که انتظار داشتیم متغیرهایی مانند دما، رطوبت و فشار هم‌بستگی بیشتری با هم دارند. اما در کل اوضاع چندان خوب به نظر نمی‌رسد. به همین دلیل و برای کم‌کردن احتمال شکست، یک بار PCA را نیز روی داده‌های ورودی اعمال کردم.



---

2. فیت کردن مدل‌ها

ابتدا برای صرفه‌جویی در وقت و کد، چند تابع مفید تعریف کردم. این توابع شامل Learning Curve، Validation Curve و همچنین تابع Grid Search می‌باشند. برای فیت کردن داده‌ها، از 5 مدل زیر استفاده کردم :

- SVR
- $Ridge$
- $BayesianRidge$
- $SGDRegressor$
- $KNeighborsRegressor$

اما مشکلی که وجود داشت این بود که 4 مدل اول، تنها یک خروجی داشتند و برای مدلی با چند خروجی می‌بایست از Multioutput Regressor و یا RegressorChain استفاده می‌کردم. من تصمیم گرفتم از RegressorChain استفاده کنم. چرا؟ چون در Multioutput Regressor مسئله را طوری حل می‌کند که گویی چند مسئله رگرسیون مستقل داریم؛ درحالی‌که خروجی‌های مسئله ما که غلظت آلاینده‌های مختلف هستند، احتمالا با یکدیگر هم‌بستگی خواهند داشت. در نتیجه از RegressorChain استفاده کردم که در پیش‌بینی هر خروجی، علاوه بر داده‌های ورودی، از خروجی‌های به‌دست آمده‌ی قبل از خود نیز استفاده می‌کند.

این کار یک مشکل جدید برای ما ایجاد می‌کند و آن عدم توانایی برای استفاده از توابع Grid Search و Validation Curve می‌باشد. برای حل این مسئله تصمیم گرفتم برای tune کردن مدل خود، از مهم‌ترین خروجی خود استفاده کنم؛ یعنی AQI. در نهایت پس از تنظیم پارامترهای مدل، آن را روی مسئله اصلی که هفت خروجی داشت اعمال کردم.

- در هرکدام از مدل‌ها، بیشترین زمان صرف شده مربوط به بخش Grid Search بود و خود فیت کردن مدل، مدت زمان کمی (اکثرا کمتر از 30ثانیه) طول می‌کشید. همچنین، در همه مدل‌ها test size برابر با 0.2 بود.

- در آخر پس از فیت کردن همه مدل‌ها، نتایج زیر به‌دست آمد :

Model Name	SVR	$Ridge$	$BayesianRidge$	$SGDRegressor$	$KNeighborsRegressor$
Best Score	0.333	0.348	0.342	-1.59	0.404

بهترین Score به‌دست آمده مربوط به $KNeighborsRegressor$ و تقریبا برابر 0.4 بود. اکنون که به اینجای کار رسیدم، فکر می‌کنم اگر همان مسئله طبقه‌بندی را حل می‌کردیم، پاسخ‌های به مراتب بهتری برای آن شاهد بودیم. به هر حال، حل مسئله به صورت رگرسیون هم چندان خالی از لطف نبود.

باتشکر از زمانی که برای خواندن این مطلب صرف کردید.

با آرزوی سلامتی و حال خوب.