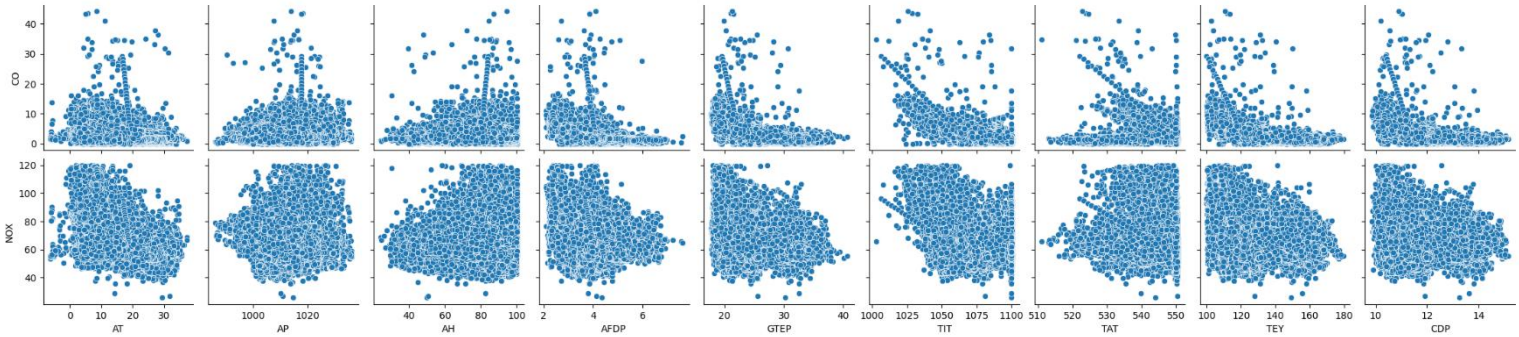
# ترمودینامیک ۲

**نام و نام خانوادگی**: سینا دانیالی **شماره دانشجویی**: 401108931

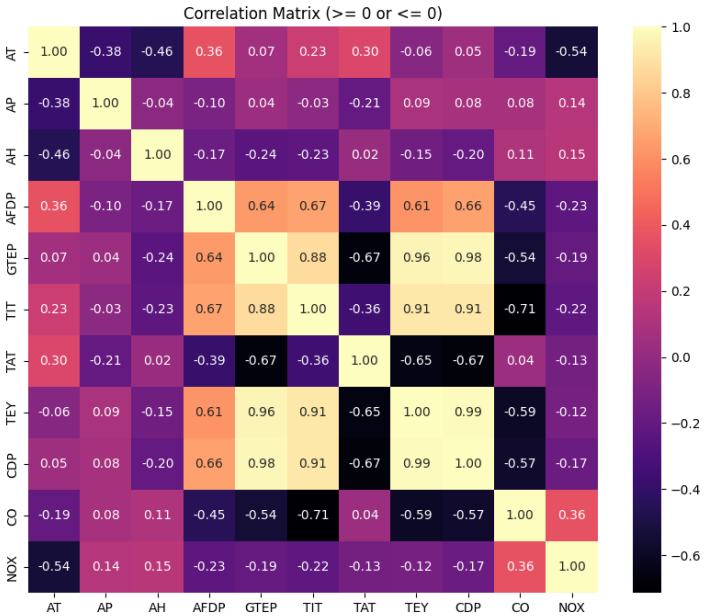
**عنوان گزارش**: پروژه ماشین لرنینگ

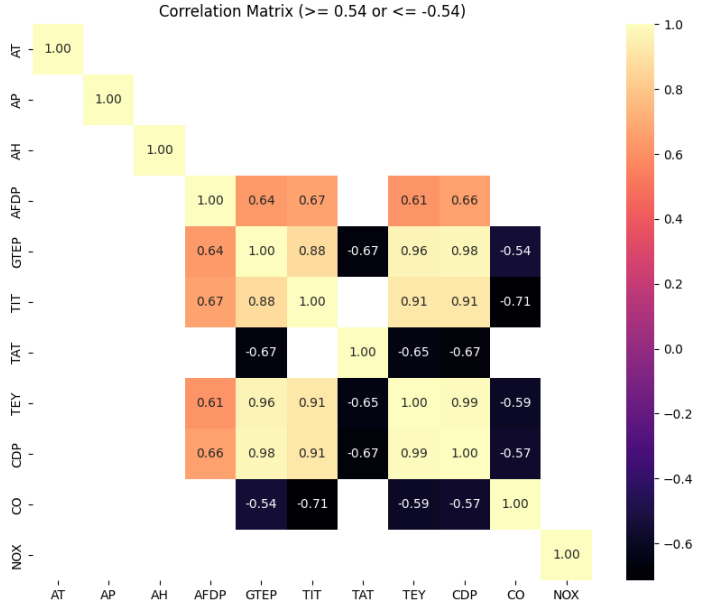
فایل‌های دیتاست به هم اضافه شده و یک فایل را تشکیل داده‌اند.

در ابتدا پس از مشاهده‌ی جلسات و نمونه‌ها، سعی شد الگوریتمی بهینه برای پیش بینی این دو دسته آلاینده پیاده سازی شود. در گام اول این پروژه، به رسم ماتریس کوریلیشن(correlation matrix) پرداختم که ارتباطات خطی موجود دیده شود. همچنین با استفاده از کدهایی که در فایل دوم jupyter موجود است، به رسم گراف‌های این دو آلاینده پرداختم و دیگر فاکتورها را هم بر حسب هم plot کردم که در صورت وجود ارتباط مفید، از آن‌ها استفاده شود. گراف‌ها:




سپس برای ویژگی‌های مختلف correlation matrix (سمت راست: مقادیر بیش از ۰٫۵۴):




پس از ملاحظه‌ی این ماتریس، به این نتیجه رسیدیم که NOX را نمی‌توان با رگرسیون خطی پیش بینی کرد، زیرا ستون مربوط به NOX مقادیر بالایی (نزدیک به یک) ندارد. ولی CO مقادیر خوبی در رگرسیون خطی دارد و امکان پیاده‌سازی این الگوریتم در آن وجود دارد.

دیتاها با هم مرج شدند و در یک فایل گرد هم به نام gt_new.csv جمع شدند.

با سرچ در یوتیوب متوجه روند پیاده‌سازی الگوریتم RF شدم و تصمیم گرفتم از این الگوریتم استفاده کنم. اگر چه که برای CO رگرسیون خطی چندگانه هم به کار رفته است. به علاوه در حال جستجو برای این مسئله بودم که به مقاله‌ای برخوردم. مقادیر عددی حل شده که در جدول این مقاله بود را در فایل قرار می‌دهم.

**Table 4**
Performance indicators for CO prediction of regressors based on RCSV hyperparameters tuning using 5-fold CV.

| Model | Feature | $RMSE_1$(training) | $RMSE_1$(validation) | $R_1^2$(training) | $R_1^2$(validation) | $R_1^2$(harm) | Rank |
|---|---|---|---|---|---|---|---|
| Ridge | Raw | 1.4794 ± 0.0236 | 1.6725 ± 0.1086 | 0.5703 ± 0.0181 | 0.3929 ± 0.1363 | 0.4652 | 30 |
| LASSO | Raw | 1.5166 ± 0.0256 | 1.5794 ± 0.0810 | 0.5485 ± 0.0183 | 0.4627 ± 0.1027 | 0.5019 | 25 |
| kNN | Raw | 0.9805 ± 0.0415 | 1.6029 ± 0.0907 | 0.8112 ± 0.0147 | 0.4422 ± 0.1337 | 0.5724 | 17 |
| Cubist | Raw | 0.9410 ± 0.0331 | 1.7147 ± 0.1757 | 0.8262 ± 0.0098 | 0.3294 ± 0.2687 | 0.4710 | 29 |
| RF | Raw | 0.3959 ± 0.0126 | 1.5148 ± 0.1260 | 0.9692 ± 0.0018 | 0.4931 ± 0.1581 | 0.6537 | 8 |
| LGBM | Raw | 0.9021 ± 0.0242 | 1.4847 ± 0.1109 | 0.8404 ± 0.0054 | 0.5142 ± 0.1493 | 0.6380 | 15 |
| CatBoost | Raw | 0.7192 ± 0.0177 | 1.4962 ± 0.0986 | 0.8985 ± 0.0050 | 0.5083 ± 0.1454 | 0.6492 | 10 |
| DFR | Raw | 0.4203 ± 0.0533 | 1.4508 ± 0.0929 | 0.9647 ± 0.0101 | 0.5355 ± 0.1416 | **0.6887** | 1 |
| Ridge | Top5 | 1.4731 ± 0.0234 | 1.6985 ± 0.1233 | 0.5740 ± 0.0183 | 0.3719 ± 0.1501 | 0.4514 | 31 |
| LASSO | Top5 | 1.5166 ± 0.0256 | 1.5796 ± 0.0809 | 0.5485 ± 0.0183 | 0.4625 ± 0.1028 | 0.5019 | 26 |
| kNN | Top5 | 0.9659 ± 0.0372 | 1.6156 ± 0.0911 | 0.8168 ± 0.0119 | 0.4351 ± 0.1242 | 0.5677 | 19 |
| Cubist | Top5 | 0.9139 ± 0.0282 | 1.6067 ± 0.1059 | 0.8362 ± 0.0064 | 0.4336 ± 0.1515 | 0.5710 | 18 |
| RF | Top5 | 0.3926 ± 0.0143 | 1.5214 ± 0.1282 | 0.9697 ± 0.0018 | 0.4874 ± 0.1629 | 0.6487 | 11 |
| LGBM | Top5 | 0.8766 ± 0.0280 | 1.4876 ± 0.1112 | 0.8492 ± 0.0065 | 0.5149 ± 0.1333 | 0.6411 | 14 |
| CatBoost | Top5 | 0.6793 ± 0.0189 | 1.4915 ± 0.0765 | 0.9095 ± 0.0034 | 0.5133 ± 0.1298 | **0.6562** | 5 |
| DFR | Top5 | 0.4426 ± 0.1059 | 1.4665 ± 0.1019 | 0.9600 ± 0.0199 | 0.5230 ± 0.1564 | **0.6771** | 3 |
| Ridge | Top10 | 1.4662 ± 0.0239 | 1.7061 ± 0.1226 | 0.5780 ± 0.0173 | 0.3671 ± 0.1482 | 0.4490 | 32 |
| LASSO | Top10 | 1.5166 ± 0.0256 | 1.5796 ± 0.0809 | 0.5485 ± 0.0183 | 0.4625 ± 0.1028 | 0.5019 | 27 |
| kNN | Top10 | 0.9555 ± 0.0306 | 1.6279 ± 0.1165 | 0.8209 ± 0.0080 | 0.4329 ± 0.0932 | 0.5668 | 21 |
| Cubist | Top10 | 0.8835 ± 0.0299 | 1.6609 ± 0.1489 | 0.8470 ± 0.0053 | 0.3941 ± 0.1689 | 0.5379 | 23 |
| RF | Top10 | 0.3907 ± 0.0154 | 1.5200 ± 0.1268 | 0.9700 ± 0.0021 | 0.4888 ± 0.1596 | 0.6501 | 9 |
| LGBM | Top10 | 0.8656 ± 0.0244 | 1.4771 ± 0.1083 | 0.8530 ± 0.0052 | 0.5222 ± 0.1320 | 0.6478 | 12 |
| CatBoost | Top10 | 0.5629 ± 0.0166 | 1.4996 ± 0.0949 | 0.9138 ± 0.0019 | 0.5092 ± 0.1293 | 0.6540 | 6 |
| DFR | Top10 | 0.5157 ± 0.0860 | 1.4553 ± 0.0954 | 0.9480 ± 0.0127 | 0.5335 ± 0.1379 | **0.6828** | 2 |
| Ridge | Top15 | 1.4006 ± 0.0229 | 1.6419 ± 0.1009 | 0.6149 ± 0.0168 | 0.4140 ± 0.1348 | 0.4948 | 28 |
| LASSO | Top15 | 1.4946 ± 0.0232 | 1.5643 ± 0.0817 | 0.5615 ± 0.0189 | 0.4726 ± 0.1026 | 0.5132 | 24 |
| kNN | Top15 | 0.9639 ± 0.0339 | 1.6249 ± 0.1137 | 0.8177 ± 0.0087 | 0.4347 ± 0.0937 | 0.5676 | 20 |
| Cubist | Top15 | 0.8657 ± 0.0196 | 1.6262 ± 0.1390 | 0.8530 ± 0.0049 | 0.4227 ± 0.1451 | 0.5653 | 22 |
| RF | Top15 | 0.3881 ± 0.0169 | 1.5283 ± 0.1366 | 0.9704 ± 0.0022 | 0.4841 ± 0.1574 | 0.6460 | 13 |
| LGBM | Top15 | 0.8504 ± 0.0274 | 1.4993 ± 0.1194 | 0.8581 ± 0.0064 | 0.5048 ± 0.1494 | 0.6357 | 16 |
| CatBoost | Top15 | 0.6460 ± 0.0195 | 1.4999 ± 0.0890 | 0.9182 ± 0.0024 | 0.5078 ± 0.1331 | 0.6539 | 7 |
| DFR | Top15 | 0.5483 ± 0.0422 | 1.4757 ± 0.1052 | 0.9407 ± 0.0086 | 0.5211 ± 0.1386 | **0.6707** | 4 |

**Table 6**
Performance indicators for $NO_x$ prediction of regressors based on RCSV hyperparameters tuning using 5-fold CV.

| Model | Feature | $RMSE_2$(training) | $RMSE_2$(validation) | $R_2^2$(training) | $R_2^2$(validation) | $R_2^2$(harm) | Rank |
|---|---|---|---|---|---|---|---|
| Ridge | Raw | 8.0201 ± 0.2835 | 8.9742 ± 1.3405 | 0.5241 ± 0.0147 | 0.3055 ± 0.1522 | 0.3860 | 31 |
| LASSO | Raw | 8.1722 ± 0.2307 | 8.8594 ± 1.3761 | 0.5058 ± 0.0089 | 0.3226 ± 0.1625 | 0.3939 | 30 |
| kNN | Raw | 3.9581 ± 0.1333 | 8.1884 ± 1.4900 | 0.8841 ± 0.0040 | 0.4079 ± 0.2070 | 0.5582 | 17 |
| Cubist | Raw | 3.4166 ± 0.1246 | 9.0217 ± 1.9114 | 0.9136 ± 0.0036 | 0.2830 ± 0.2693 | 0.4322 | 26 |
| RF | Raw | 1.4852 ± 0.0515 | 8.0454 ± 2.3642 | 0.9837 ± 0.0006 | 0.4181 ± 0.2894 | 0.5868 | 13 |
| LGBM | Raw | 4.0848 ± 0.1898 | 7.8623 ± 2.4172 | 0.8766 ± 0.0061 | 0.4421 ± 0.2921 | 0.5878 | 12 |
| CatBoost | Raw | 3.4740 ± 0.1417 | 7.5844 ± 2.2079 | 0.9107 ± 0.0035 | 0.4831 ± 0.2568 | 0.6313 | 5 |
| DFR | Raw | 1.3876 ± 0.0339 | 7.6528 ± 2.1068 | 0.9866 ± 0.0004 | 0.4737 ± 0.2446 | **0.6466** | 1 |
| Ridge | Top5 | 7.8477 ± 0.2828 | 8.8006 ± 1.2802 | 0.5443 ± 0.0150 | 0.3320 ± 0.1448 | 0.4125 | 27 |
| LASSO | Top5 | 8.0746 ± 0.2293 | 8.8308 ± 1.3028 | 0.5175 ± 0.0094 | 0.3279 ± 0.1499 | 0.4015 | 29 |
| kNN | Top5 | 3.8759 ± 0.1831 | 8.4803 ± 1.4437 | 0.8888 ± 0.0070 | 0.3689 ± 0.1972 | 0.5214 | 18 |
| Cubist | Top5 | 3.0717 ± 0.1790 | 9.1445 ± 1.8200 | 0.9301 ± 0.0057 | 0.2585 ± 0.2803 | 0.4046 | 28 |
| RF | Top5 | 1.4229 ± 0.0701 | 8.0976 ± 2.4076 | 0.9850 ± 0.0009 | 0.4081 ± 0.3044 | 0.5771 | 16 |
| LGBM | Top5 | 3.9416 ± 0.2119 | 7.8735 ± 2.3924 | 0.8850 ± 0.0078 | 0.4432 ± 0.2848 | 0.5906 | 11 |
| CatBoost | Top5 | 3.2241 ± 0.1945 | 7.6050 ± 2.3047 | 0.9230 ± 0.0066 | 0.4812 ± 0.2592 | **0.6326** | 3 |
| DFR | Top5 | 1.3343 ± 0.0623 | 7.8488 ± 2.1422 | 0.9868 ± 0.0010 | 0.4514 ± 0.2496 | 0.6195 | 7 |
| Ridge | Top10 | 7.1284 ± 0.3165 | 8.2374 ± 1.3735 | 0.6240 ± 0.0198 | 0.4132 ± 0.1488 | 0.4972 | 22 |
| LASSO | Top10 | 7.3737 ± 0.2507 | 8.1341 ± 1.2779 | 0.5976 ± 0.0140 | 0.4294 ± 0.1313 | 0.4998 | 19 |
| kNN | Top10 | 3.8935 ± 0.1961 | 8.7830 ± 1.3248 | 0.8878 ± 0.0080 | 0.3261 ± 0.1855 | 0.4769 | 23 |
| Cubist | Top10 | 2.9779 ± 0.1602 | 9.2755 ± 1.8734 | 0.9344 ± 0.0045 | 0.2372 ± 0.2903 | 0.3784 | 32 |
| RF | Top10 | 1.4166 ± 0.0720 | 8.0678 ± 2.3633 | 0.9852 ± 0.0009 | 0.4137 ± 0.2949 | 0.5827 | 14 |
| LGBM | Top10 | 3.9261 ± 0.2156 | 7.8846 ± 2.2774 | 0.8859 ± 0.0081 | 0.4446 ± 0.2712 | 0.5921 | 10 |
| CatBoost | Top10 | 3.1678 ± 0.1915 | 7.6168 ± 2.0930 | 0.9257 ± 0.0062 | 0.4859 ± 0.2301 | **0.6373** | 2 |
| DFR | Top10 | 1.2718 ± 0.0707 | 7.8449 ± 1.9908 | 0.9880 ± 0.0010 | 0.4535 ± 0.2344 | 0.6217 | 6 |
| Ridge | Top15 | 7.1072 ± 0.3163 | 8.2306 ± 1.3712 | 0.6262 ± 0.0197 | 0.4146 ± 0.1466 | 0.4989 | 21 |
| LASSO | Top15 | 7.3664 ± 0.2523 | 8.1389 ± 1.2822 | 0.5984 ± 0.0140 | 0.4290 ± 0.1310 | 0.4997 | 20 |
| kNN | Top15 | 3.9220 ± 0.2127 | 8.9632 ± 1.3194 | 0.8861 ± 0.0090 | 0.2994 ± 0.1846 | 0.4476 | 24 |
| Cubist | Top15 | 3.0055 ± 0.1803 | 9.0422 ± 1.5951 | 0.9331 ± 0.0061 | 0.2873 ± 0.2104 | 0.4393 | 25 |
| RF | Top15 | 1.4099 ± 0.0741 | 8.1053 ± 2.3890 | 0.9853 ± 0.0009 | 0.4083 ± 0.2996 | 0.5773 | 15 |
| LGBM | Top15 | 3.8962 ± 0.2230 | 7.8800 ± 2.1927 | 0.8877 ± 0.0083 | 0.4474 ± 0.2580 | 0.5949 | 9 |
| CatBoost | Top15 | 3.1638 ± 0.1911 | 7.6521 ± 2.1359 | 0.9259 ± 0.0062 | 0.4792 ± 0.2393 | **0.6316** | 4 |
| DFR | Top15 | 1.1262 ± 0.0663 | 7.9106 ± 1.9222 | 0.9879 ± 0.0009 | 0.4460 ± 0.2260 | 0.6146 | 8 |

لینک مقاله: https://www.sciencedirect.com/science/article/pii/S0016236123019804

پس از اینها با تغییر مداوم مجموعه ویژگی ها سعی کردم به وضعیت بهینه دست یابم. یعنی دیتاهایی که به نسبت ۸۰ به ۲۰ تقسیم شده بودند، در تست و ترین استفاده شدند. تست برای پیش‌بینی و ترین هم برای آموزش مدل.

باقی نکات و... در کد موجودند. برای اینکه نحوه پیش بینی بهتر به چشم بیاید، نمودار هم رسم شده است. البته نمودارهای پایانی رسم نشده که البته اهمیتی ندارد، چرا که بیش از همه دقت دارند.

در زمان‌های پایانی اتمام پروژه، با مشورت یک فرد با تجربه، تصمیم گرفتیم به دلیل ضعیف بودن درخت تصمیم، از رگرسیون غیرخطی هم استفاده کرده و نتیجه را ببینیم. به وسیله‌ی امکانی که در پایتون وجود داشت، موفق شدیم به دقت بالای ۰/۷ برای هر دو آلاینده دست پیدا کنیم. برای یافتن این ماکسیمم مقدار، از greedy search استفاده کردیم. به این صورت که ابتدا مجموعه‌های سه‌تایی از ویژگی‌ها را گرفتیم و واریانس آن‌ها را بررسی کردیم. سپس مجموعه‌های چهارتایی را امتحان کردیم که دیدیم دقت بهتری دارند. بررسی مجموعه‌های پنج‌تایی ممکن بود، ولی به شدت زمان‌بر بوده و به CPU فشار وارد می‌کرد (در چهارتایی‌ها هم همینطور، به شدت طول می‌کشید).

از بین درجه‌هایی که موجود بود، با چند تست، دیدیم درجه سه نسبت به بقیه بهینه‌تر است (البته درجه پنج با مجموعه‌های سه‌تایی هم بهینه بود).

نکته‌هایی که عدم رعایت آنها موجب دردسر بود و بعد فهمیدم:

۱- فایل اکسل، csv نیست و باید به آن تبدیل شود.

۲- در استفاده از ترین و تست باید دقت کرد. آموزش با ترین است و پیش‌بینی فقط با دیتای تست باید انجام شود.

۳- متغیرها و... در ژوپیتر ذخیره می‌شود. برای ران کردن کد باید ابتدا کرنل را ری‌استارت کرد تا از خطاهای احتمالی جلوگیری شود.

۴- وقتی از greedy search استفاده کردم، در jupyter خیلی اذیت شدم. پیشنهاد دوستان این بود که از google colab استفاده کنیم که به CPU دستگاه فشار وارد نشود و سرعت بیشتری داشته باشد.

۵- در پروژه، برای گرفتن واریانس از RF.score(x_train,y_train) استفاده می‌کردم که متوجه شدم اشتباه است و باید از x_test و y_real استفاده می‌کردم. این‌ها دیتای تست هستند و درست. RF در کد تعریف شده است.

۶- ابتدا که میخواستم پروژه را شروع کنم، !wget که در فایل بود اجازه Run نمی‌داد، با بررسی متوجه شدم این برای نسخه‌های قبل ژوپیتر است و برای دانلود فایل دیتاست، ما به آن نیازی نداریم.

۷- برای پلات کردن نمودار نیاز به اندیس داشتم که آن را با linspace ساختم.

کدهای مربوط به ماتریس‌ها و پلات‌ها موجود است، در صورت نیاز قابل ارسال می‌باشد.