Project Report

COMP9318

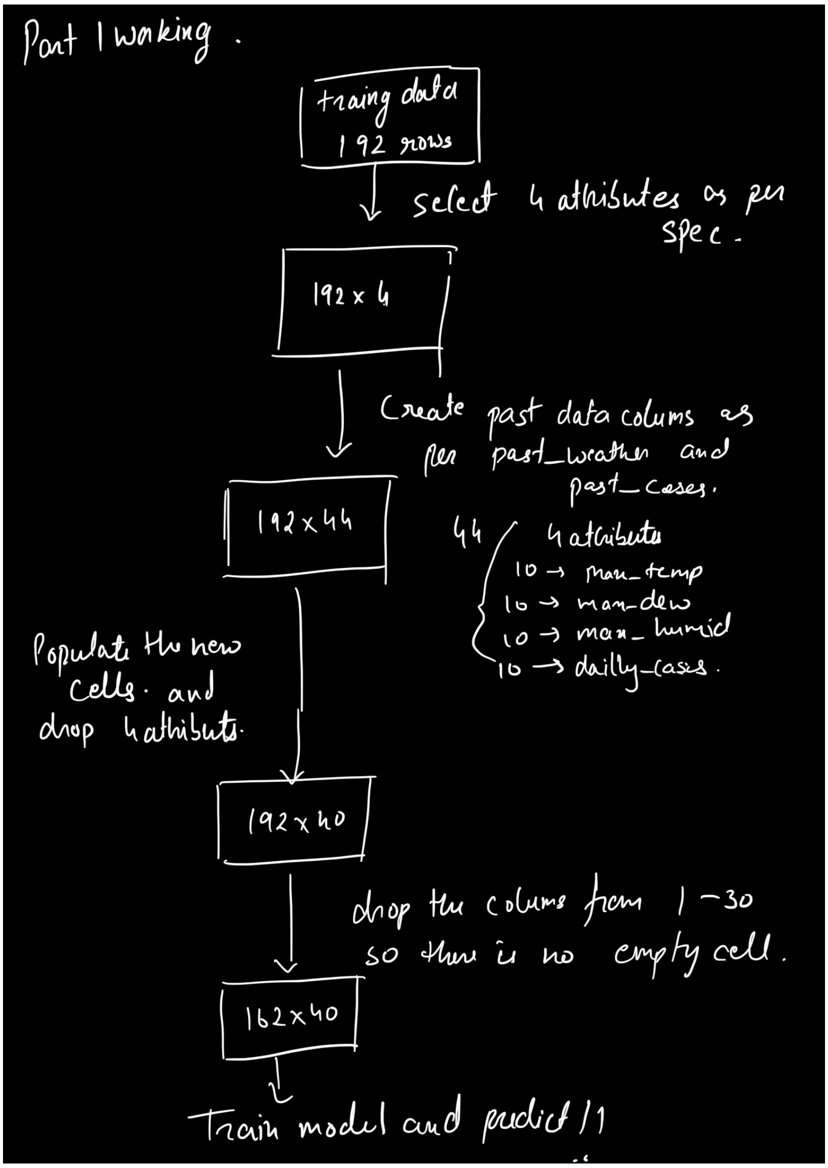
By: Arth Sanskar Patel

z5228942

Project Part 1

Implementation

For Project part 1, we had to create a training matrix of the size 162 x 40. The 162 rows come from days from 30 to 192. The 40 columns come from the attributes for previous days. We had to take 4 attributes namely 'max_temp', 'max_dew', 'max_humid', and ‘dailly_cases’. The attributes 'max_temp', ‘max_dew’, and ‘max_humid' were weather attributes and ‘dailly_cases’ was a case attribute.


Part 1 working .
traing data
192 rows
select 4 attributes as per spec .
192 x 4
Create past data colums as per past_weather and past_cases.
192 x 44
44
4 attributes
10 → max_temp
10 → max_dew
10 → max_humid
10 → dailly_cases .
Populate the new cells. and drop 4 attributes.
192 x 40
drop the colums from 1 -30 so there is no empty cell.
162 x 40
Train model and predict

Instead of going like explained in the lectures to make a matrix of size 162 x 510 and then select the needed days and attributes to get a sub matrix for training of size 162 x 40, I planned on creating directly the 162 x 40 matrix. This was more advantageous as it reduced the time and complexity of the whole code.

So first I extracted the required attributes from the train_df. After that I divided them into weather attributes and case attribute. For each weather attribute, I created the empty data columns which were named like the ones found in the test_features and number of these data columns were equal to the past_weather_interval. After that I populated those data cells with the required previous data. I repeated the same process for cases attribute as well. In the end, I had a data frame which was of size 192 x 44. So dropped the columns which were initially the attributes and dropped the rows before day number 31.

After training the model with the aforementioned data frame, I extracted the required features from test_feature and did the prediction.

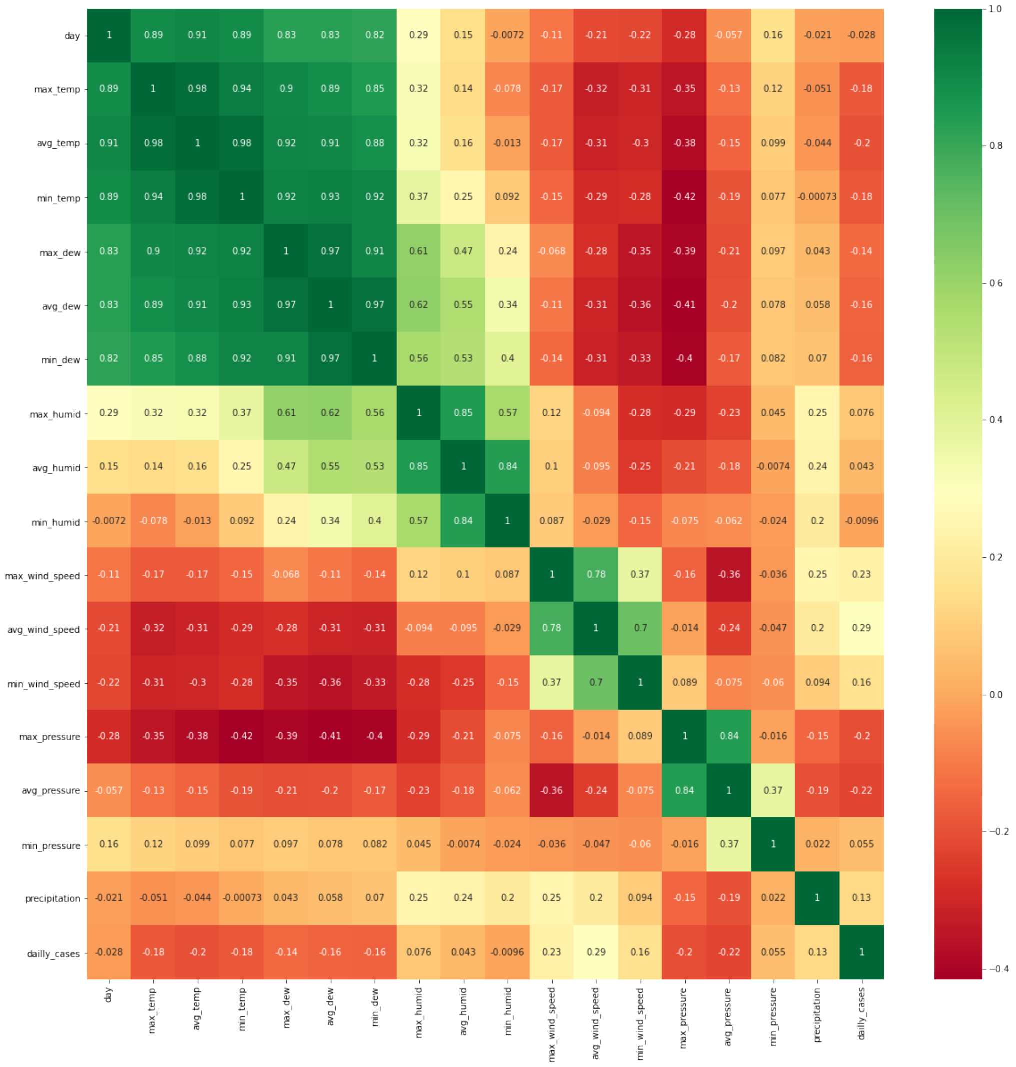
The image above shows my working for part 1 before I wrote a single line of code.

Project Part 2

Feature Selection

The implementation for this part was really tricky. A lot of feature analysis algorithms and different feature engineering practices revealed surprising results which eventually influenced the way I created my model for part 1.

1. So first I created a correlation matrix or a heat map to find which features are best for choosing while creating a new model. The result was as follow:

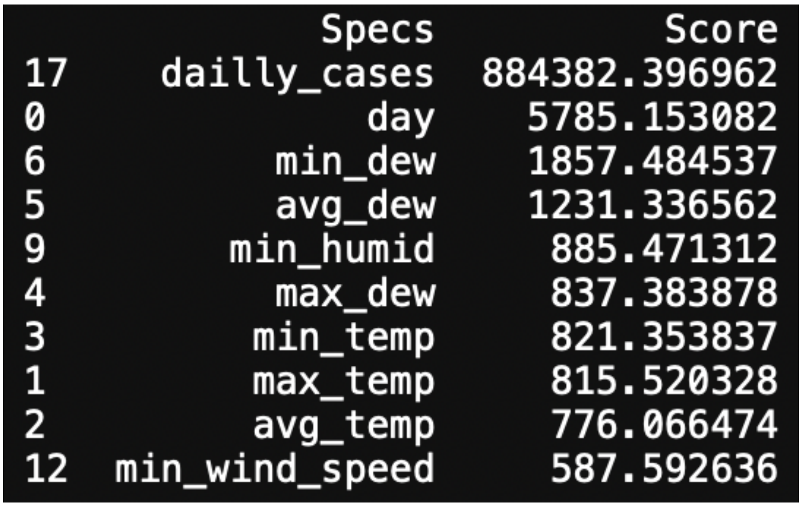


	day	max_temp	avg_temp	min_temp	max_dew	avg_dew	min_dew	max_humid	avg_humid	min_humid	max_wind_speed	avg_wind_speed	min_wind_speed	max_pressure	avg_pressure	min_pressure	precipitation	dailly_cases
day	1	0.89	0.91	0.89	0.83	0.83	0.82	0.29	0.15	-0.0072	-0.11	-0.21	-0.22	-0.28	-0.057	0.16	-0.021	-0.028
max_temp	0.89	1	0.98	0.94	0.9	0.89	0.85	0.32	0.14	-0.078	-0.17	-0.32	-0.31	-0.35	-0.13	0.12	-0.051	-0.18
avg_temp	0.91	0.98	1	0.98	0.92	0.91	0.88	0.32	0.16	-0.013	-0.17	-0.31	-0.3	-0.38	-0.15	0.099	-0.044	-0.2
min_temp	0.89	0.94	0.98	1	0.92	0.93	0.92	0.37	0.25	0.092	-0.15	-0.29	-0.28	-0.42	-0.19	0.077	-0.00073	-0.18
max_dew	0.83	0.9	0.92	0.92	1	0.97	0.91	0.61	0.47	0.24	-0.068	-0.28	-0.35	-0.39	-0.21	0.097	0.043	-0.14
avg_dew	0.83	0.89	0.91	0.93	0.97	1	0.97	0.62	0.55	0.34	-0.11	-0.31	-0.36	-0.41	-0.2	0.078	0.058	-0.16
min_dew	0.82	0.85	0.88	0.92	0.91	0.97	1	0.56	0.53	0.4	-0.14	-0.31	-0.33	-0.4	-0.17	0.082	0.07	-0.16
max_humid	0.29	0.32	0.32	0.37	0.61	0.62	0.56	1	0.85	0.57	0.12	-0.094	-0.28	-0.29	-0.23	0.045	0.25	0.076
avg_humid	0.15	0.14	0.16	0.25	0.47	0.55	0.53	0.85	1	0.84	0.1	-0.095	-0.25	-0.21	-0.18	-0.0074	0.24	0.043
min_humid	-0.0072	-0.078	-0.013	0.092	0.24	0.34	0.4	0.57	0.84	1	0.087	-0.029	-0.15	-0.075	-0.062	-0.024	0.2	-0.0096
max_wind_speed	-0.11	-0.17	-0.17	-0.15	-0.068	-0.11	-0.14	0.12	0.1	0.087	1	0.78	0.37	-0.16	-0.36	-0.036	0.25	0.23
avg_wind_speed	-0.21	-0.32	-0.31	-0.29	-0.28	-0.31	-0.31	-0.094	-0.095	-0.029	0.78	1	0.7	-0.014	-0.24	-0.047	0.2	0.29
min_wind_speed	-0.22	-0.31	-0.3	-0.28	-0.35	-0.36	-0.33	-0.28	-0.25	-0.15	0.37	0.7	1	0.089	-0.075	-0.06	0.094	0.16
max_pressure	-0.28	-0.35	-0.38	-0.42	-0.39	-0.41	-0.4	-0.29	-0.21	-0.075	-0.16	-0.014	0.089	1	0.84	-0.016	-0.15	-0.2
avg_pressure	-0.057	-0.13	-0.15	-0.19	-0.21	-0.2	-0.17	-0.23	-0.18	-0.062	-0.36	-0.24	-0.075	0.84	1	0.37	-0.19	-0.22
min_pressure	0.16	0.12	0.099	0.077	0.097	0.078	0.082	0.045	-0.0074	-0.024	-0.036	-0.047	-0.06	-0.016	0.37	1	0.022	0.055
precipitation	-0.021	-0.051	-0.044	-0.00073	0.043	0.058	0.07	0.25	0.24	0.2	0.25	0.2	0.094	-0.15	-0.19	0.022	1	0.13
dailly_cases	-0.028	-0.18	-0.2	-0.18	-0.14	-0.16	-0.16	0.076	0.043	-0.0096	0.23	0.29	0.16	-0.2	-0.22	0.055	0.13	1

The result is very surprising as not a single feature was highly correlated with the daily cases. All of them had very small influence over how the daily cases came out. A lot of the features had negative relation and the ones that had positive were showing very low correlation. As the industry standard is to take features with correlation higher than 0.5, here the highest feature had correction of 0.29. So this didn't prove helpful In terms of feature selection.

2. Second I did K-best selection of feature using chi squared analysis. Chi square test is also used for finding correlation between features by using different statistic as compared to a correlation matrix.

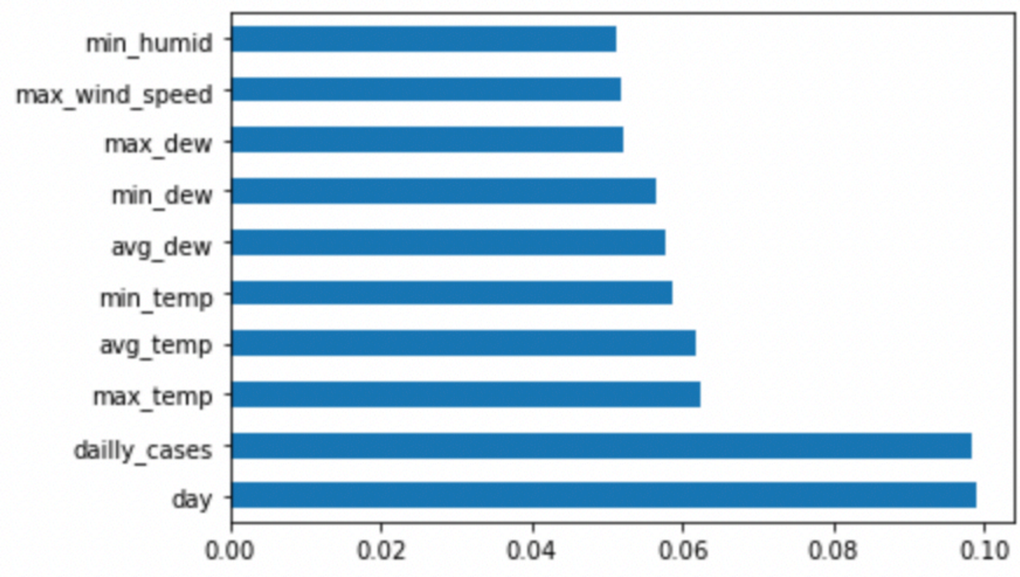
The result of chi squared test was as follows:



```
              Specs          Score
17     dailly_cases  884382.396962
0               day    5785.153082
6           min_dew    1857.484537
5           avg_dew    1231.336562
9         min_humid     885.471312
4           max_dew     837.383878
3          min_temp     821.353837
1          max_temp     815.520328
2          avg_temp     776.066474
12   min_wind_speed     587.592636
```

The result shows top 10 features for the training data. The ‘dailly_cases’ feature here is again the the best followed by ‘day’ and then followed by ‘min_dew’. There is very humungous difference between the scores of 1st and 2nd feature. This test also proved that the final answer heavily relies only on the ‘dailly_cases’ feature and nothing else.

3. Till now I was unsure whether to do more feature analysis of just go ahead or training with only ‘dailly_cases’. The last test I did was feature importance selection using an ExtraTreeClassifier. It revealed the top 10 features as follows:



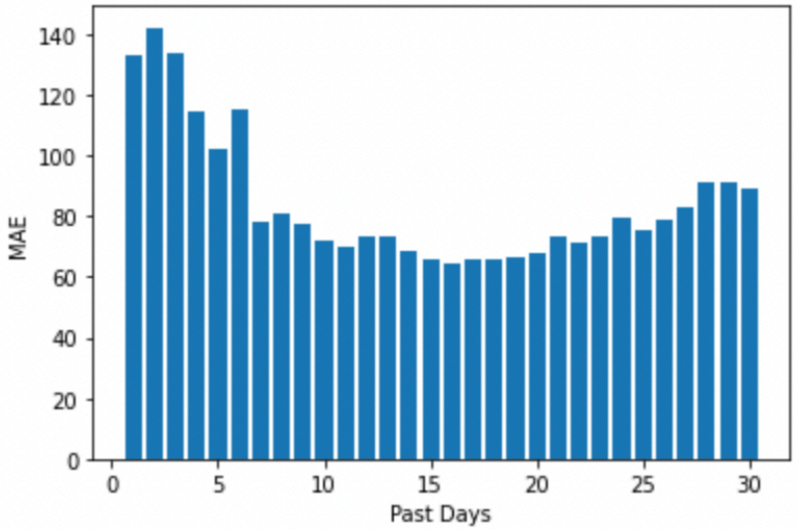
This final test proved that all the features were literally useless in terms of helping for better model prediction so I decided to drop all of them except for ‘dailly_cases’ and moved forward with tuning the hyper parameters.

Hyper parameter tuning and new SVR model proposition

1. The first thing I did with hyper parameter tuning was creating a GridSearchCV model. GridSearchCV essentially tries different combinations of hyper parameters to find the best parameters to have the best score possible.

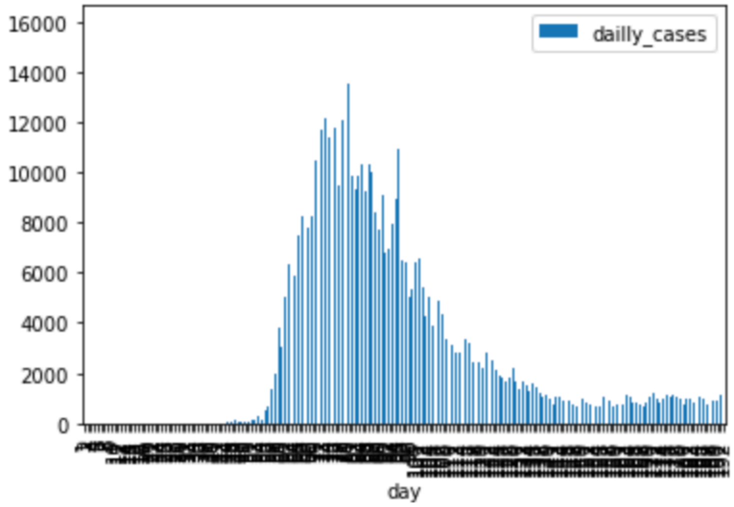
   Choosing from 112060 combinations of different hyper parameters, GridSearchCV returned the best parameters to choose for the SVR model. Using those hyper parameters for the the training data with just 'dailly_cases' gave the score of nearly 86 with past_cases_interval being 10 days.

2. Next, I tried all different values for 'past_cases_interval' to see what number of 'days' were appropriate for the model.



The graph above shows different MAE for different number of past days selected for 'past_cases_interval'. As clear from the graph, the lowest MAE was achieved when past days was 16. So for more analysis I chose the past days to be from 15 to 20.

3. The final part for creating a new SVR model to achieve MAE lower than 80. No amount of hyper parameter tuning and feature selection lowered the MAE score below 80. So I did a basic analysis on the training data. The result was as follow:



The data had high amount of variance. It has an increasing phase, a decreasing phase and a constant phase. For a single model of SVR, it is hard to accumulate such high amount of variance. So I divided the data into 3 parts and created an individual model for each part. The increasing phase ranged from day 50 to day 80. The decreasing phase ranged from day 81 to day 137. The constant phase ranged from day 138 to day 192.

After training all 3 models on separate training datas for there corresponding ranges, I made all of them predict on the full data set and gathered those results in a data frame of size 192 x 3. In the end I used this newly formed data frame to train my final SVR model and do the prediction.

So when I got the test_feature, I extracted the relevant days from it as mentioned in part 1, did 3 predictions on those extracted features using my 3 phase models and then finally did the prediction on the results from phase model by a final model to get the averaged out number of cases prediction.

This allowed me to reduce my MAE from 80 to 65.

Hence this new model of SVR proved helpful in better predictions.