COMP9318 - Project report

Yao Yuan z5092195

Part-1

The core part for solving Part-1 is at how to construct the feature matrix for the training set and apply the same structure to the test set.

As detailed in the specification, we have a maximum 30-day limit on the past instances for each weather, precipitation and daily case.

In my code, I used the same name conventions as that in test_features (which is provided) for past instances.

Therefore, I firstly create a list of names which will be lately used as column names in training set. It has 510 in total length including day.

```
1 feature_matrix_X_for_max_interval.columns
```

```
Index(['max_temp-1', 'max_temp-2', 'max_temp-3', 'max_temp-4', 'max_temp-5',
       'max_temp-6', 'max_temp-7', 'max_temp-8', 'max_temp-9', 'max_temp-10',
       ...
       'dailly_cases-21', 'dailly_cases-22', 'dailly_cases-23',
       'dailly_cases-24', 'dailly_cases-25', 'dailly_cases-26',
       'dailly_cases-27', 'dailly_cases-28', 'dailly_cases-29',
       'dailly_cases-30'],
      dtype='object', length=510)
```

In the second step, I create the dataset for the training set by searching in train_df (which is provided).

```
# create the data with 30days past records
i = 30
add_rows = []
while i < train_df.shape[0]:
    row = []
    for word in expanding_list:
        feature,day_number = word.split('-')
        row.append(train_df.iloc[i-int(day_number)][feature])
    add_rows.append(row)
    i += 1
```

We know that there are 192 total number of records in train_df and I need to computer the past 30 days instances for 162 of them because for the first 30 records, they do not have sufficient past 30 days instances.
Noted that “max_temp-1” of the day 30 is the “max_temp” of the day 29 instead of day 1.

After adding the dataset into a data frame, we can get a 162x510 feature matrix for training set.

Since in part-1, we only need to consider the number of past instances which is specified by parameters “past_cases_interval” and “past_weather_interval”.

We can directly select those past instances from the feature matrix which has past instances for 30 days to form the training data set for part-1.

For the labels of the training set data, we can simply select the last 162 records out of 192 records in column “daily_cases”.

Besides, we also need to select the target feature instances for test data set.
Noted that because we are doing the prediction one by one, in order to feed the test data into the model, we need to reshape the test data set.

```
# get the part1 testing data
part_1_test_features_X = test_feature[selected_columns].values.reshape(1, -1)
```

After we have all these three datasets:

- Training feature set X from the pre-set past interval (size: 162x40)
- Training labels (size: 162x1)
- Test feature set X from the pre-set past interval. (size: 20x40)

We can fit training sets to the pre-defined SVM model.
And get the predicted values for the test set.

At the last step, we need to take the floor integers of predicted values.

Part-2

Feature selecting:
I tried [1,5,10,15,20,25,30] time interval to “temp”, “dew”, “humid” and “daily-cases”.
Ordered the MAE in ascending order and pick and first one.
As shown in the below image, when time interval for case, temp, dew, humid are 15, 10, 25, 30 respectively, the MAE has the smallest value.

```
68.27427442180252: cases:15, temp:10, dew:25, humid:30, MAE:68.27427442180252
68.30895179713727: cases:15, temp:30, dew:30, humid:10, MAE:68.30895179713727
68.33403664044988: cases:15, temp:10, dew:20, humid:30, MAE:68.33403664044988
68.34725766870294: cases:15, temp:10, dew:30, humid:30, MAE:68.34725766870294
68.35040634085308: cases:15, temp:20, dew:30, humid:30, MAE:68.35040634085308
68.35148441502739: cases:15, temp:10, dew:15, humid:30, MAE:68.35148441502739
68.35450298652862: cases:15, temp:15, dew:25, humid:30, MAE:68.35450298652862
68.36357980952448: cases:15, temp:15, dew:30, humid:30, MAE:68.36357980952448
68.36841151982296: cases:15, temp:20, dew:25, humid:30, MAE:68.36841151982296
68.398603846287: cases:15, temp:10, dew:30, humid:25, MAE:68.398603846287
```

For the combination of max, min, avg of temp, dew, humid, I find that use all of them at the same time can produce the best results.

Model Parameter tuning:

For SVR model, we have a parameter called “C” which is a regularization parameter. From the following graph, we can see that when C equals to 21000, MAE has the lowest value.

```
C:1000, MAE:96.3707527427412
C:3000, MAE:88.27629868656665
C:5000, MAE:81.19348154707329
C:7000, MAE:76.97963781393494
C:9000, MAE:73.11019846253512
C:11000, MAE:71.37238248703423
C:13000, MAE:70.81720068346046
C:17000, MAE:68.70261668208221
C:20000, MAE:68.49518009338455
C:21000, MAE:68.45511185603972
C:25000, MAE:69.89461031239087
```

For SVR model, we have a parameter called “kernel” which specifies the kernel type used in the algorithm. We can see that when kernel mode is ‘poly’, MAE has the lowest value.

```
kernal:poly, MAE:68.45511185603972
kernal:rbf, MAE:87.59685860152368
kernal:sigmoid, MAE:9964.628011678658
```

I also tuned the “degree” parameter when I use the poly kernel mode and I find that when degree equals to 1, MAE has the lowest value which is 68.45.

I did the same procedure to coef0 and tol and epsilon. I find that

Kernel	Degree	C	Gamma	Coef0	tol	epsilon
Poly	1	17000	scale	7.5	2.5	0.04

The parameters above can produce the best result which is 66.9.