COMP9318: DATA WAREHOUSING AND DATA MINING
PROJECT REPORT

Submitted By: BADRIVISHAL IYENGAR (z5300253)
Date: 23.04.2021

Implementation details of Part 1

Upon detailed study of the testing dataset, the provided training dataset needed to be transformed significantly to fit the model for generating predictions.
The first step was to extract the necessary features from the training data (train_data) i.e., the features max_temp, max_dew, max_humid and dailly_cases.
Since the dataset needed to be further transformed, each feature has been maintained as a separate data frame for the ease of transformation and understanding.
Which gives us a data frame of the size: 192*5
A further column and row transformation of these features to obtain day wise ordered details of all features leaves us with a matrix of the size 162*120.
Since this matrix contains data of up to 30 days, this vector is further reduced based on input parameters (past_weather_interval) and (past_cases_interval) and the necessary required dataframe is generated for Training.
For the provided input parameter values; 10 for each, we obtain a training vector of the size 162*40.
Upon fitting this model along with the testing model to the given SVR model with preset Hyperparameters, we obtain a frame with predicted cases as a float value which is transformed into integer values using the built in function math.floor().

The result based on the dataset provided:

```
[945, 897, 832, 881, 907, 921, 1028, 819, 812, 809, 860, 845, 837, 898,
861, 811, 846, 839, 855, 892]
```

Implementation details of Part 2

To begin with the part 2 implementation, a study of the hyper parameters tu ned for the part 1:

```
svm_model = SVR()
svm_model.set_params(**{'kernel': 'rbf', 'degree': 1, 'C': 5000,
                        'gamma': 'scale', 'coef0': 0.0, 'tol': 0.001, '
epsilon': 10})
```

The parameter ‘degree’ is significant only with kernel values ‘sigmoid’ and ‘poly’. Hence, in order to change the degree values so as to get some changes in the model, the kernel values need to be adjusted along with. Upon testing different kernel values such as ‘poly’/ ‘linear’/ ‘sigmoid’ it was established that for the given dataset, only ‘rbf’ was the best suited kernel type as the Mean Absolute Error dropped significantly beyond 100. Similar performance was observed while changing the ‘gamma’ values from default to auto.

The different models used with their results as given below:

Features: max_temp, max_dew, max_humid, max_pressure, daillycases
(Default hyperparameters)

```
[965, 913, 913, 770, 959, 1079, 1031, 800, 748, 953, 835, 806, 974, 860, 887, 840, 986, 836, 833, 1087]
MeanAbsError =  87.3
```

Features: avg_temp, avg_dew, avg_humid, daillycases
(Default hyperparameters)

```
40
41
42 MeanAbsError = mean_absolute_error(predicted_cases_part2, ground_truth)
43 print('MeanAbsError = ', MeanAbsError)
```

```
[947, 910, 860, 778, 895, 1074, 1056, 763, 762, 870, 840, 785, 961, 891, 850, 828, 926, 816, 802, 1027]
MeanAbsError =  83.7
```

Features: max_temp, max_dew, max_humid, daillycases
Hyperparameters: ‘C’:50000

```
[928, 925, 843, 858, 883, 1056, 1118, 754, 812, 834, 890, 816, 943, 965, 836, 833, 902, 848, 827, 990]
MeanAbsError =  83.6
```

With some further increasing the hyperparameter, ‘C’:
The model with significantly greater performance was yielded with the ‘C’:170000, which was:

```
[963, 925, 870, 802, 909, 1089, 1070, 779, 781, 878, 867, 802, 976, 908, 862, 847, 936, 839, 822, 1045]
MeanAbsError =  79.75
```

Hence, by significantly reducing the strength of regulization, the model developed yielded a Mean Absolute Error of 79.75