Description

For our dataset, we have arranged them into three parts, training data, validation data (Beijing and Shenyang) and evaluation/test data (Shanghai/Guangzhou). Our implementation is briefly introduced as below.

Data preprocessing
First, we merge the data from Beijing and Shenyang and store them in a data frame. Since the features of the data are in different scale, we standardize the features to have mean value as 0 and standard deviation as 1. Then we split the data into training set and validation set.

Fit/Predict/Score functions
KNN classification does not have a real "learning" phase. The prediction will totally depend on the training data without learning a model from it. Thus, KNN classification does not need a fit function to infer the model parameters from the training data. Instead, it has a hyperparameter k which we need to choose.

We build a class called knn_classifier. The main functions in the class are focused on two parts, predict and score.

- **Predict function**
  The prediction is based on the nearest neighbors' labels. We use Euclidean distance to calculate distance between the test data point and all the other training data points. We select k nearest data points, and the label with the highest proportion in these data points is the predicted label for the test data point.
- **Score function**
  The function is to calculate the accuracy rate of predicted labels compared with the real labels of the test data points.

To choose the best hyperparameter k, we do prediction on our split validation data for different values of k. When k is 29, we get a relatively high accuracy of 0.809. We will use this setting for further evaluation.

Finally, we evaluate our model on Shanghai data and Guangzhou data separately. The prediction for Shanghai data gets an accuracy of 0.759. The prediction for Guangzhou data gets an accuracy of 0.767. The reason why the test score is lower than the training score may be that there exist different feature patterns between different cities. For further improvements, we can tune our model on different feature combinations and weights. We can also try with other classification model like support vector machine.