**Linear Regression for Air Quality dataset**

**Task:** Predict one of the gases in the dataset given other attributes.

**Dependencies:**

Pandas: for loading data\_set into data\_frame format which can be easily converted to matrix

Matplotlib: for plotting data

Numpy: for doing some calculation

Sklearn: for machine learning related tools

**Selecting the right features:**

First, In order to select good features to do the prediction we need to look at the dataset and find out how complete each feature is. In other words, how little missing values does each feature. If they have too many missing values, we shouldn’t use it.

|  |  |
| --- | --- |
| **Feature** | **Number of missing values** |
| Time | 0 |
| CO(GT) | 1683 |
| PT08.S1(CO) | 366 |
| NMHC(GT) | 8443 |
| C6H6(GT) | 366 |
| PT08.S2(NMHC) | 366 |
| NOx(GT) | 1639 |
| PT08.S3(NOx) | 366 |
| NO2(GT) | 1642 |
| PT08.S4(NO2) | 366 |
| PT08.S5(O3) | 366 |
| T | 366 |
| RH | 366 |
| AH | 366 |

As seen from the above table, NMHC(GT) has considerably more missing values than other features, so I decided to omit this feature from the prediction.

Date feature is not shown here too because I omitted it as it is not necessary.

Other features have relatively similar number of missing values so they should be fine.

Next, in order to select the best target feature to predict, I decided to test 3 times on different targets.

I predicted CO(GT), C6H6(GT), and PT08.S1(CO). Below is their accuracies.

CO(GT)

Mean squared error: 3299.69394788

Variance R2 score: 0.43159855

C6H6(GT)

Mean squared error: 1.19982339

Variance R2 score: 0.99929704

PT08.S1(CO)

Mean squared error: 5610.23741217

Variance R2 score: 0.94835187

This shows that:

- CO(GT) is a bad feature because it underfits.

- C6H6(GT) is also a bad feature because it overfits.

- PT08.S1(CO) is the good feature to predict as its accuracy is optimal.

Reason why CO(GT) is underfits because of too many missing values compared to C6H6(GT) and PT08.S1(Co).

**Final decision:** Pick PT08.S1(CO) as the target feature. And use the algorithm to predict it given other features (time, humidity, other gases ...)

**Cross-validation**

The result of doing cross-validation of 5-fold on the training data we got:

- Negative mean squared error:

[-26815.66890206 -13552.99411731 -4472.23654668 -4543.15653207

-8228.02389605]

- R2 score:

[ 0.70350609 0.76208437 0.95674566 0.97328018 0.92788725]

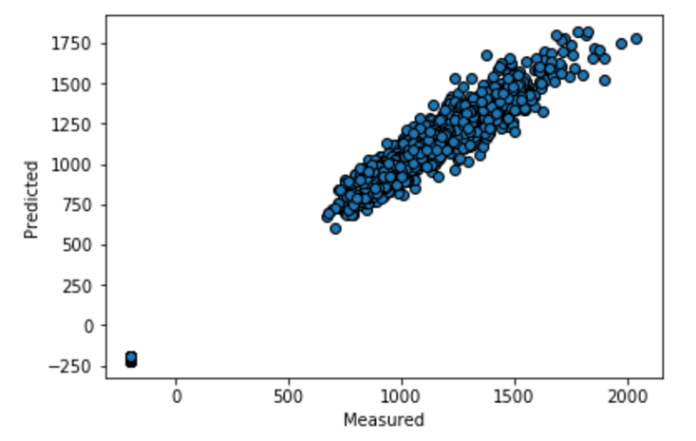
We can see from the score that the 4th fold is the best as its R2 score is closest to 1.

**Fitting the model:**

With ScikitLearn we can easily fit the model with Linear Regression and then test the result with final prediction as:

Mean squared error: 5610.23741217

Variance R2 score: 0.94835187

This shows that the model can fit the data quite well, given the rest of the other features.