CS771 Assignment 3

The Brainiacs

Aishwarya Srivastava (200064)

Akanksha Singh (200070)

Anushka Panda (200174)

Kshiteesh Bhardwaj (200525)

Raj Vardhan Singh (200760)

1 Calibrate Away

In this task, we will calibrate two sensors, one measuring the level of O3 and another measuring the level of NO2. Both these sensors are electrochemical in nature i.e. in response to changing levels of the pollutant they are measuring, they output two voltages called OP1 and OP2. More specifically, the O3 sensor outputs voltages named o3op1, o3op2 whereas the NO2 sensor outputs voltages named no2op1, no2op2. These sensors are cross-sensitive in nature i.e. the ozone sensor measures levels of not just ozone but also nitrogen dioxide. Hence, specifically we wish to learn some real-valued constants `po3, qo3, ro3, so3, to3` such that the true level of ozone is given by -

`po3 · o3op1 + qo3 · o3op2 + ro3 · no2op1 + so3 · no2op2 + to3`

and for some other real-valued constants `pno2, qno2, rno2, sno2, tno2`, we have the true level of nitrogen dioxide given by -

`pno2 · o3op1 + qno2 · o3op2 + rno2 · no2op1 + sno2 · no2op2 + tno2` .

1.1 Predict the true level of O3 and NO2 using linear model

In this task we have used just the 4 voltage values to predict O3 and NO2 values using different regularizers and different loss functions.

Preprint. Under review.

1.1.1 MAE values using various linear models

Linear model	MAE for O3	MAE for NO2
Linear Regression	5.753484037817828	6.488030728000253
Ridge Regression	5.752187286097613	6.476017848587365
Lasso	5.780007568390212	6.536223147679995
SGD	38.170080286532487	18.989000080218488

We have seen that the method that gave us the best-performing linear model (in terms of MAE on training data) is Ridge Regression.

MSE values For O3

Alpha	Least Squares Loss	ϵ-insensitive Loss
100	5.752187286097613	5.914484037817828
1	5.754481871697708	5.654484037817828
0.1	5.754460212606847	5.524484037817828

MSE values For NO2

Alpha	Least Squares Loss	ϵ-insensitive Loss
100	6.476017848587365	6.48354800802848
1	6.453572656782226	6.08354800802848
0.1	6.453950249048202	5.97354800802848

1.2 With the rest of the data

For trying new approaches along with the rest of the data provided to us, we tried multiple but the best approach turned out to be the random forest regression mechanism.

1.2.1 What is random forest algorithm

The random forest algorithm is an ensemble algorithm that uses multiple decision trees to come up with a final output. Each decision tree is given a random subset of the features, and based on those features each decision tree gives an output. Then an aggregate is taken for all the trees and that aggregate is used as the final output of the ensemble.

1.2.2 With or without you attributes

Random forest has its advantage over the decision tree since it reduces overfitting and takes into account the importance of each attribute by checking how a decision tree will perform *with or without the subset of attributes*.

1.2.3 But wait, isn't decision tree a classifying mechanism?

A subtle property of decision tree is manipulated when using it for regression. Instead of classifying according to reduction in entropy, some loss function (like the MSE) is used to make the split between the more likely and the less likely values. Thus a decision tree is able to send a given test point to the more MSE/ less MSE branches. (explanation from: this source)

1.2.4 Approach

For the random forest, we took maximum depth to be 15, number of trees to be default hundred, and the loss function to be Mean Squared error. The Random forest was implemented using the `sklearn.ensemble.RandomForestRegressor` class, and it uses the default `sklearn.tree.DecisionTreeRegressor`'s loss function, which is by default set to `square_error`.
We calculated the results for ozone and NO2 with and without the time values. The time values were included as a feature using the following heuristic.

- The median of values of sunset and sunrise time during March, April and May turns out to be *lying in April*, hence the value turns out to be `19:45:00` and `06:55:00`. (According to this source).

- The time between these timestamps is taken to be night, and the rest of the time is taken to be day

- According to these times, the list `in_the_moonlight` stores value `0` for nights, and `1` for days, and the feature is included in the overall feature array.

2 Code implementing our Random forest model

...can be checked out in `submit.py` :)

2.1 Results

With time

T_test	MAE for O3	MAE for NO2
0.30030031520000194	1.978254960281057	1.4418261711239608

Without time

T_test	MAE for O3	MAE for NO2
0.22307990819899715	1.4307856524666667	0.949665824140477