# **Africa Soil Property Prediction Challenge Solution**

**My Score** : **0.42980**

**Method :**

**Step 1 :**

# **Finding relevant variables , feature reduction :**

Since , this data set had a large number of features find correlation using *df.corr()* was not a good option and also feature reduction did not lead to betterment of outcomes .

**Step 2 :**

# **Outlier detection :**

The standard method I used was plotting BoxPlot of each output variable and looking for possible outliers :

***The outcomes were :***

All P > 1 were outliers .

All SOC > 2 were outliers

All Ca > 1 were outliers

All pH> 2 were outliers

And sand had no outliers .

All possible combinations were tried .

All gave better outcome in the in-sample dataset but were not so good on out-sample data . The outliers in P though when removed gave optimum results both in the in-sample and out-sample datasets .

So , *finally only Outliers in P were removed as all others were affecting the prediction accuracy and were not necessarily useless* .

**Step 3 :**

**# Choosing regression technique :**

**Linear :**

This is a regression analysis , so we can choose many techniques , linear is not good , as the number of features is way too large and the outputs may not necessarily follow the linear trend .

**SVR:**

The main difference to ordinary regression is that the SVM regression function f(x) will not change when adding additional data points (y\_i, x\_i) as long as f(x\_i) does not deviate more than a certain margin epsilon from y\_i. Furthermore, further deviations are, in effect, penalized linearly instead of quadratically. As a third difference, the objective function also contains a quadratic term penalizing too high absolute pre - factors of x. This is referred to as "flatness" and I do not understand the rationale behind it. .

Thus in this case it was a very good algorithm to use and the one which i used to get the final outcome.

**DecisionTreeRegressor**

DecisionTreeRegressor also does a similar task , but due t very large number of r=features , the results in SVR were much better than DecisionTreeRegressor .

**Libraries / packages / environment / language used:**

Python 3

Jupyter notebook

Numpy

Pandas

Scikit-learn

Matplotlib

Code can be found here :