Dolores

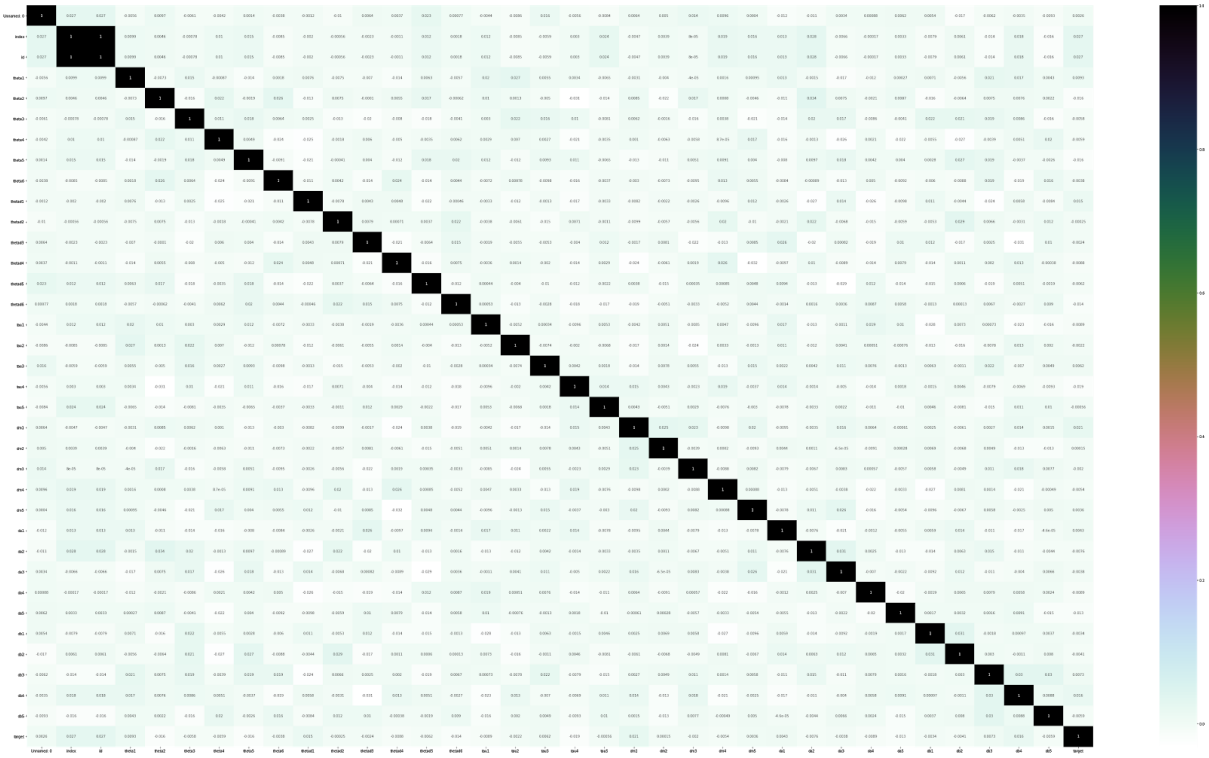
Yash Choudhary **(sofu_ML)**
18118085
Metallurgical and Materials Engineering

Link to Git Repository-https://github.com/angrycharas/Dolores-sofu_ML-

Started with importing several libraries like **numpy**, **pandas**, **matplotlib** and **seaborn**.

Then I imported and analysed the dataset(s). It had **7192** data points and **36** features (Some were highly irrelevant though)
I realised that several features seemed to be related by the **name itself** like theta1, theta2, theta3 etcetera.

I decided to make a covariance matrix of the whole dataset and realised that our **target values** had a positive correlation only with the features namely theta1, thetad1, tau3, dm1, dm2, dm5, da1, db3 and db4.
(But the magnitude of these “positive values” of correlation were fairly low, and were close to zero)
I even plotted a heatmap of these correlation values using seaborn.



So i decided to use all features instead of only these features(with +ve correlation values) because they didn't seem to offer much. (I tried them out of course only to find out a rsme of 1.19 ;))

Then I decided to merge the **obviously related** features to reduce the number of features to 6.
For this I tried some centralization techniques but finally settled with simple Arithmetic Mean.
For Example-
I created a new feature **"theta"**
theta = (theta1 + theta2 + theta3 + theta4 + theta5 + theta6) / 6

Similarly I created 5 more features namely "thetad", "tau", "dm", "da", "db".

With total 6 features this time I once again computed the Covariance matrix.

	target	theta	thetad	tau	dm	da	db
target	1.000000	-0.015737	-0.006753	-0.011134	0.007663	-0.013232	0.004469
theta	-0.015737	1.000000	-0.014377	0.004149	-0.005994	-0.005575	0.022147
thetad	-0.006753	-0.014377	1.000000	-0.028215	-0.019780	-0.016795	-0.003657
tau	-0.011134	0.004149	-0.028215	1.000000	-0.005185	0.004511	-0.014405
dm	0.007663	-0.005994	-0.019780	-0.005185	1.000000	-0.009233	-0.002992
da	-0.013232	-0.005575	-0.016795	0.004511	-0.009233	1.000000	-0.000193
db	0.004469	0.022147	-0.003657	-0.014405	-0.002992	-0.000193	1.000000

Yet again the positive ones had a low magnitude and therefore I decided to proceed with these 6 features.

I created new dataset(s) of these 6 new variables and named them as **train_AM** and **test_AM**.

I tried out various methods of Regression like **Linear Regression**, **Support Vector Regression**, **Decision Tree**, **Random Forest** etcetera.

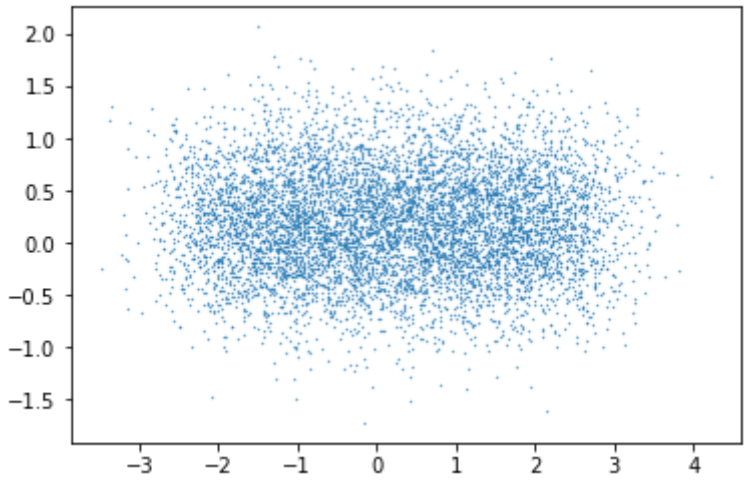
And after trying these I finally settled with Random Forest Regression to train my dataset.

I also made a few twitches with the parameter **n_estimators** to better understand Overfitting and Underfitting of my model.
Finally I managed to get a score of **0.24**

During the process I tried out various visualisations.
Such as different variables against the target values, Various Combinations of variables against the target values.

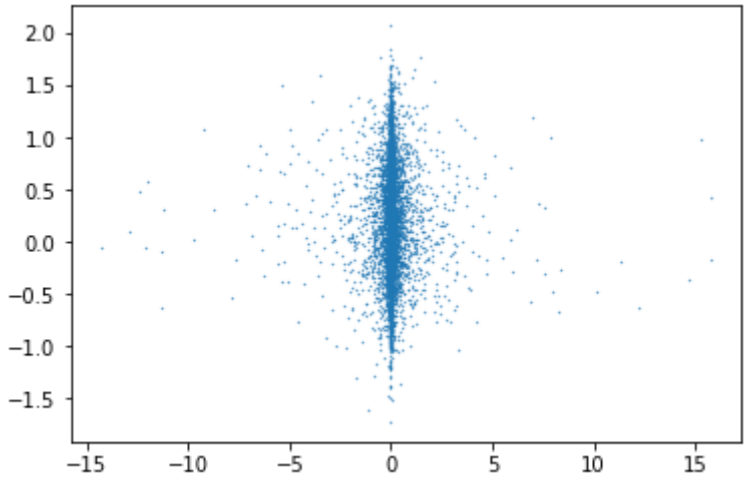
For example-

Target Values V/S theta6



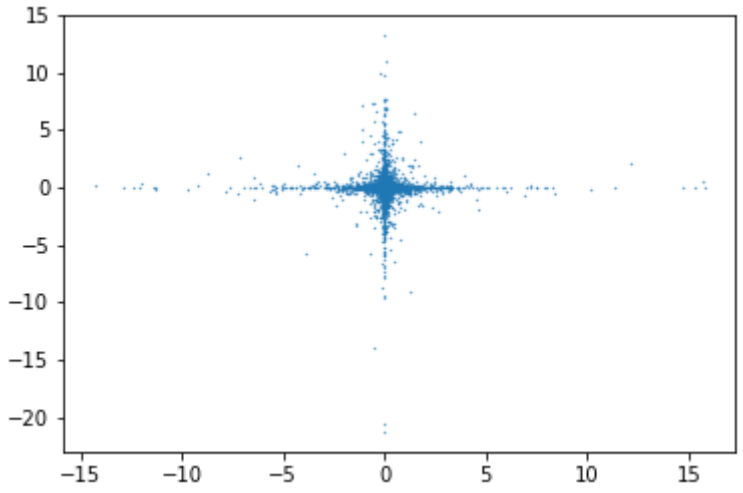
Target Values V/S Alpha

Here Alpha= theta1*theta2*theta3....*theta6



Aplha V/S Beta

Here Alpha= theta1*theta2*theta3....*theta6
Here Beta= thetad1*thetad2*thetad3....*thetad6



“It was a nice learning experience.”
Thank you DSG