**Pseudo Code for Model Training**

Importing Libraries like Pandas, Numpy Scikit-learn, Matplotlib and Seaborn

Reading the data and define Input Parameters like 'Ni%', 'Cr%', 'Fe%', 'Mn%', 'Cu%', 'Al%', 'Si%', 'C%', 'S%', 'Mo%', 'Ti%', 'Co%', 'B%', 'P%', 'Nb & Ta%', 'Ni% + Co%', 'W%', 'La%', 'Zr%', 'IP (Amp)', 'Ton (µS)', 'Toff (µS)', 'Voltage (Volts)'

Storing input features in variable x.

Define target variables "Surface Roughness (µm)" and "MRR (mm3/min)"

Storing output variables in y variable.

Algorithms like Linear Regression, Random Forest, KNN and SVM are imported from scikit-learn library and stored in a list named models.

Dividing the dataset in training and testing sets.

Scaling the input variables by using Standard Scaler.

# Iterating over each model

For each model in models:

Selecting a model from list.

Fitting/Training the set of input and output variables

Predicting the test dataset.

# Calculating Root Mean Squared Error

Appending the predicted SR and MRR and actual SR and MRR in relevant lists.

sr\_p = Predicted SR values

sr\_a = Actual SR values

mrr\_p = Predicted MRR values

mrr\_a = Actual MRR values

Setting the initial sums of RMSE to zero

For i in range(0, len(predictions):

Calculating the differences between actual and predict variables

Squaring the Differences

Adding these squared differences to the initial sum

Taking the mean of the Squared Differences

Taking Square root of the Squared Differences

A dictionary is made, where the name of Model Applied, The RMSE of SR and RMSE of MRR are appended

This dictionary is then appended to a list.

A dataframe is made by converting the above dictionary.

This dataframe is then saved to a comma separated file

Note: This procedure was repeated with some minor changes for Cross Validation and Parameter Tunning.