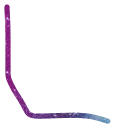
**Table 7**. Hyperparameter tuning details for ML models

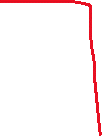
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Hyperparameters tuned** | **Grid or values used for tuning** | **Values Justification** | **Model Structure Analysis** | **Final value Selected** |
| Base LR | - | Default values were used for all parameters. | No Hyper-parameters. | All variants of linear regression assume that the target variable and the feature variables have a linear relationship. Thus, all these models mainly focus on minimizing any deviations from such linear relationship. Lasso also attempts to select some features so that way it attempt to recognize a linear relationship with a spare feature set while Ridge attempt to find a robust linear estimator when multicollinearity exists in the data. Elasti-Net attempt to find a middle ground between these two approaches. | - |
| Ridge | alpha | 500 samples uniformly selected starting from 10-20 to 1020 | When the value is 1, equal weight is given to Base LR objective function as well as the L2 regularizer. When the value is close to 10-20, loss function is equivalent to Base LR and when close to 1020, it’s equivalent to L2 regularizer. Thun values were sampled uniformly in between. | 3.258562e-04 |
| Lasso | alpha | 500 samples uniformly selected starting from 10-20 to 1020 | When the value is 1, equal weight is given to Base LR objective function as well as the L1 regularizer. When the value is close to 10-20, loss function is equivalent to Base LR and when close to 1020, it’s equivalent to L1 regularizer. Thun values were sampled uniformly in between. | 3.237457e-05 |
| Elastic-Net | Alpha | 500 samples uniformly selected starting from 10-20 to 1020 | Alpha used to weight the regularization effect on the loss function. It is similar to Lasso/Ridge, thus similar approached was used to select values. | 2.459130e-05 |
| L1 ratio | 100 samples uniformly selected starting from 0 to 1 | L1 ratio determine the ratio of L1 to L2 regularization effect thus value is between 0 and 1. When 0, there is no L1 effect and when 1, there is no L2 effect. Thus, values were sampled from [0, 1]. | 9.090909e-02 |
| Random Forest | N estimators | 64, 128, 256, 512, 1024, 2048, 4096, 8192 | Small values are generally good for small datasets and large values for large datasets, since there is no reference value, range of values were used. |  | 1024 |
| Max depth | None, 5, 10, 15, 20, 25, 30, 35, 40, 50 | Experimented within a full range of depth. | 15 |
| Min samples split | 2, 4, 8, 16 | Controls the splitting of the tree thus base 2 was used and up to 3 generations. | 2 |
| Min samples leaf | 1, 2, 4, 8 | Controls overfitting and small values were used given the small number of samples. | 1 |
| bootstrap | True, False | This determines if we sample with replacement or not. | True |
| XGBoost | Max depth | 3,4,5 |  |  | 5 |
| Learning rate | 0.01.0.1,0.2 |  | 0.1 |
| N estimators | 100,200,300 |  | 300 |
| Subsample | 0.8, 0.9, 1.0 |  | 0.8 |
| Colsample bytree | 0.8, 0.9, 1.0 |  | 1 |
| AdaBoost | N estimators | 50, 100, 200 |  |  | 200 |
| Learning rate | 0.01, 0.1, 1 |  | 1 |
| Base estimator max depth | 2, 3, 4 |  | 4 |
| ANN | Batch size | [1,2,4,8,16,32,64,128,256,512] | Stochastic gradient descent happens with batch size of 1 thus it was used. Small batch sizes were also used since they act as implicit regularizer. Large batch sizes were also used since they are a accurate estimate of full batch gradient descent. | ANN can find non-linear relationships between target and features and can perform as a universal function approximator. ANN have no prior knowledge (i.e., inductive bias) about the relationship thus requires lots of data to find the relationship. | 16 |
| Learning rate | [0.5,0.1,0.05,0.01,0.005, 0.001,0.0005, 0.0001,0.00005, 0.00001] | Since Adam was used, this is the starting LR. Since regression problem, small values were also used to control the gradient update and large LR were also used to avoid local minima. | 0.001 |
| Number of epochs | 300 | Large number of epochs since copy of the model was saved at the end of each epoch. | 300 |
| Initialisation of weights | X\_n, x\_u | Uniform and Normal distributions are the standard approaches. | Xavier normal |
| Number of hidden layers | 4,3 | Constrained relatively shallow architecture since dataset is relatively small. Used pyramid architecture since it is the simplest approach with a single-target regression model. | 4 |
| Nodes per layer | [256,128,64,16], [128,64,16] | [256,128,64,16] |
| Activation function | Leaky Relu, Relu | Relu results in sparsity while Leaky Relu can control the sparsity with it’s associated alpha hyper-parameter. Since there is no prior knowledge about ANN usage in this domain, both were attempted. | Relu |
| Optimizer | Adam | Adam is generally robust to sub-optimal local minima and can find a better performing minimum point without fine-grained adjustments. | Adam |
| Loss function | MSE | Standard loss function with regression problem. | MSE |
| Regularisations methods used | None at the moment | Since generalization of the model was satisfactory, regularization methods (ex: weight decay) were not used. | None |
| Early stopping method | The training and validation loss was tracked for all 300 training epochs, and the training epoch resulting in the minimum validation loss was considered the optimal. | Manual early stopping was used with saved copies of model at the end of each epoch. | Epoch 290 |

A group of blue and white graphs

Description automatically generated with medium confidence

N\_total = N\_bulk + N\_surface

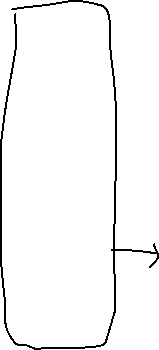
If we assume the nano particle to be a cube,



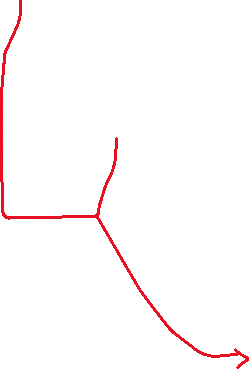
(@ used to denote proportionality, r for the radius)

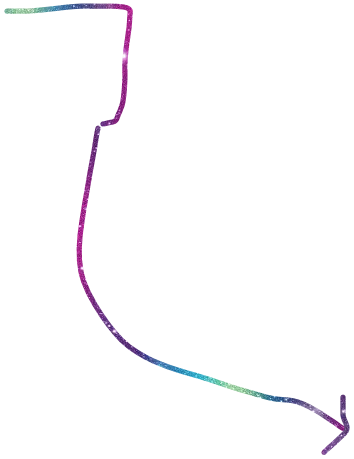
r @ side length

Area @ N\_surface @ r^2



Volume @N\_bulk @ r^3



 N\_surface = k\*r^2

N\_bulk = kk\*r^3

N\_total = kk\*r^3 + k\*r^2 (pay attention to the y-axis values)

This can be extended to these two as well

Linear relationship is because N\_bulk>>> N\_surface

So (approximately) N\_total @ N\_bulk