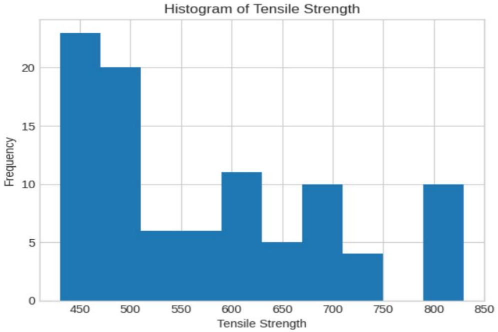
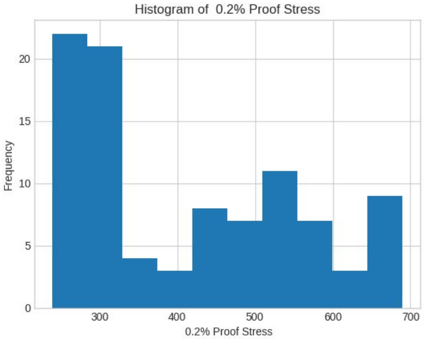
DESIGN OF ADVANCED HIGH- STRENGTH STEEL (AHSS) USING MACHINE LEARNING

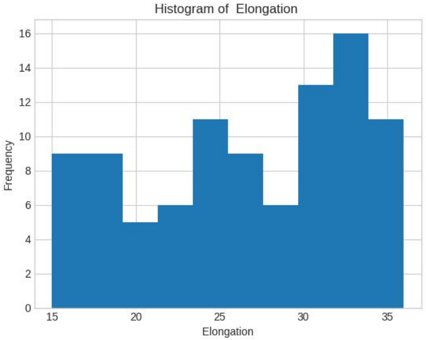
Submitted By
- **Suyash Kushwaha**
  (B21MT035)

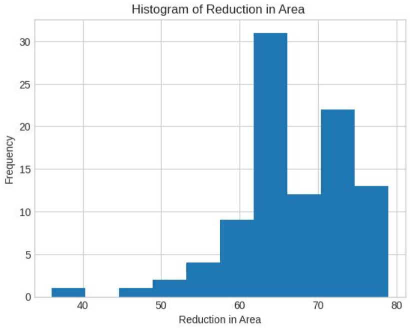
Supervised By
Nitin Kumar Sharma
Assistant professor
Department of metallurgical and materials engineering
IIT,Jodhpur

DataSet









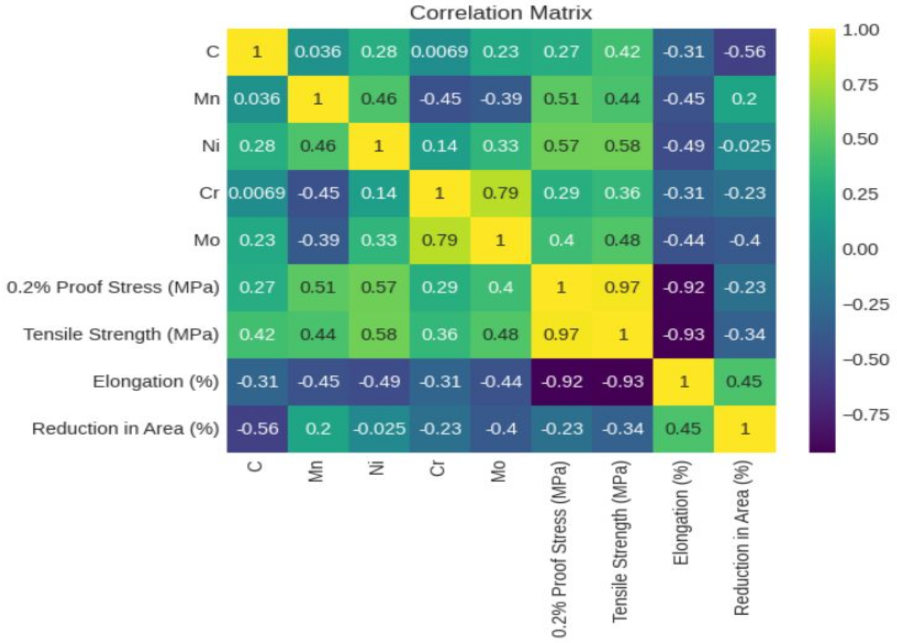
ï»¿Alloy Code	C	Si	Mn	P	S	Ni	Cr	Mo	Cu	V	Al	N	Ceq	Nb + Ta	Temperature (Â°C)	0.2% Proof Stress (MPa)	Tensile Strength (MPa)	Elongation (%)	Reduction in Area (%)
CbX	0.17	0.49	1.26	0.013	0.01	0.01	0.02	0.09	0.02	0.04	0.045	0.0025	0.429	0.0	27	537	672	22	71
CbY	0.13	0.42	1.28	0.014	0.012	0.21	0.22	0.05	0.04	0.03	0.037	0.0072	0.424	0.0	27	476	603	27	76
CbZ	0.16	0.41	1.48	0.014	0.01	0.02	0.02	0.01	0.03	0.08	0.033	0.0079	0.436	0.0	27	577	693	20	62
CCA	0.23	0.23	1.23	0.027	0.012	0.03	0.04	0.017	0.01	0.009	0.003	0.0111	0.0	0.0017	27	359	578	27	70
CCB	0.22	0.22	1.24	0.021	0.008	0.03	0.05	0.017	0.01	0.005	0.005	0.0116	0.0	0.0017	27	382	580	28	70

- Data were collected from kaggle:https://www.kaggle.com/datasets/konghuanqing/matnavi-mechanical-properties-of-lowalloy-steels

Data Preprocessing

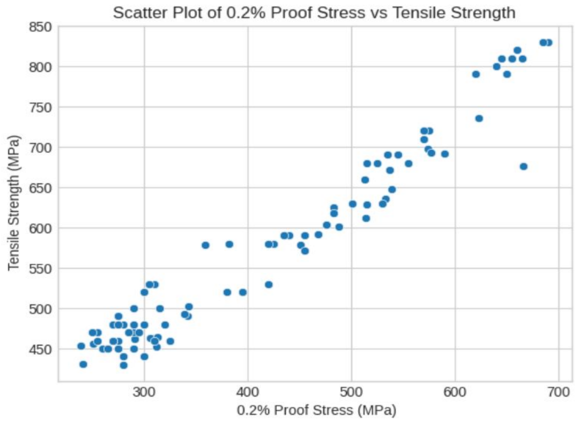
Data cleaning : Checked for Null values, and if found than replace it with mean or median(No null value found). To handle Outliers first figured out then remove them (we found outliers in Elongation and Reduction in Area column).From this data set we didn't need temperature data,so drop that column. Clean data from Duplicate data (No duplicates found).

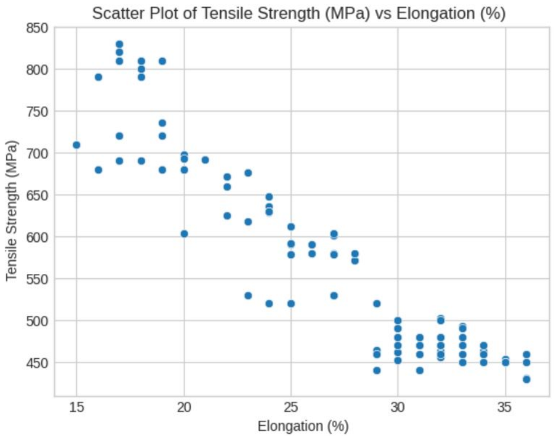
Data Exploration and Analysis :

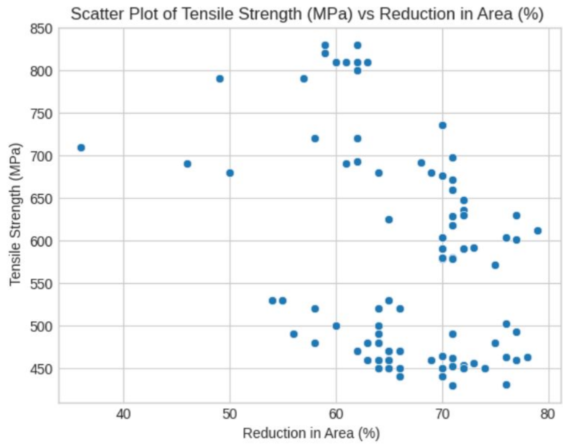


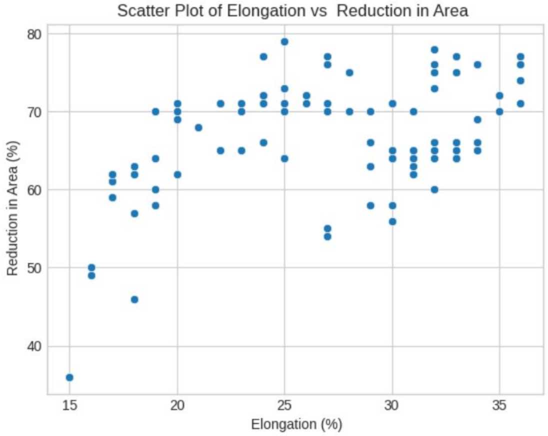
	0.2% Proof Stress (MPa)	Tensile Strength (MPa)	Elongation (%)	Reduction in Area (%)
count	95.000000	95.000000	95.000000	95.000000
mean	413.263158	572.326316	26.642105	66.631579
std	139.756415	118.989222	6.036971	7.403353
min	239.000000	430.000000	15.000000	36.000000
25%	287.500000	470.000000	22.000000	63.000000
50%	380.000000	530.000000	27.000000	66.000000
75%	531.500000	666.000000	32.000000	71.000000
max	690.000000	830.000000	36.000000	79.000000

Data Preprocessing









Tensile strength and yield strength are showing linear relationship(positive correlation).

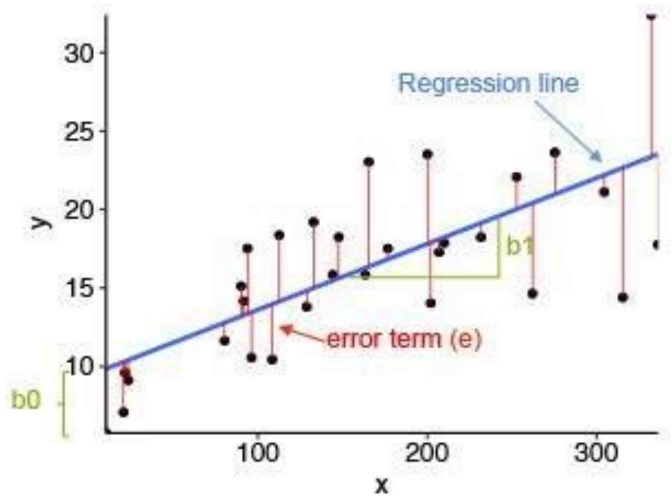
Tensile strength and Elongation are showing negative linear relationship(negative correlation).

Tensile strength and Reduction in Area are showing no such direct relationship.

Reduction in Area and Elongation are also not showing any such direct relationship.

Linear Regression

- **Linear regression** models the relationship between a dependent variable (y) and one or more independent variables (x), used for predicting future values based on the independent variable(s).
- Multiple linear regression involves **multiple independent variables** and one dependent variable, used when multiple factors affect the dependent variable.
- Linear regression is used for predicting **continuous/numeric variables** like sales, salary, age, product price, etc.
- Cost function optimizes regression coefficients/weights, measures model performance, and accuracy of the mapping function (hypothesis function) that maps input to output variable.
- **Mean Squared Error (MSE)** is a common cost function in linear regression, calculating average squared error between predicted and actual values.



For the above linear equation, MSE can be calculated as:

$$MSE= 1\frac{1}{N}\sum_{i=1}^{n}(y_i - (a_1x_i + a_0))^2$$

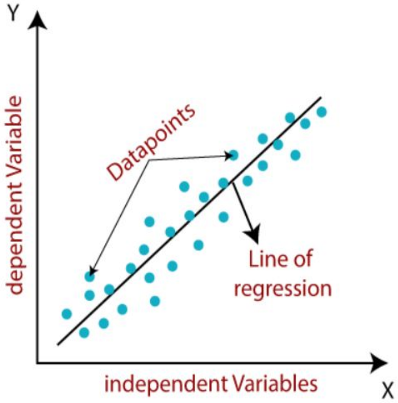
y = mx+c

c is a constant

m is the regression coefficient

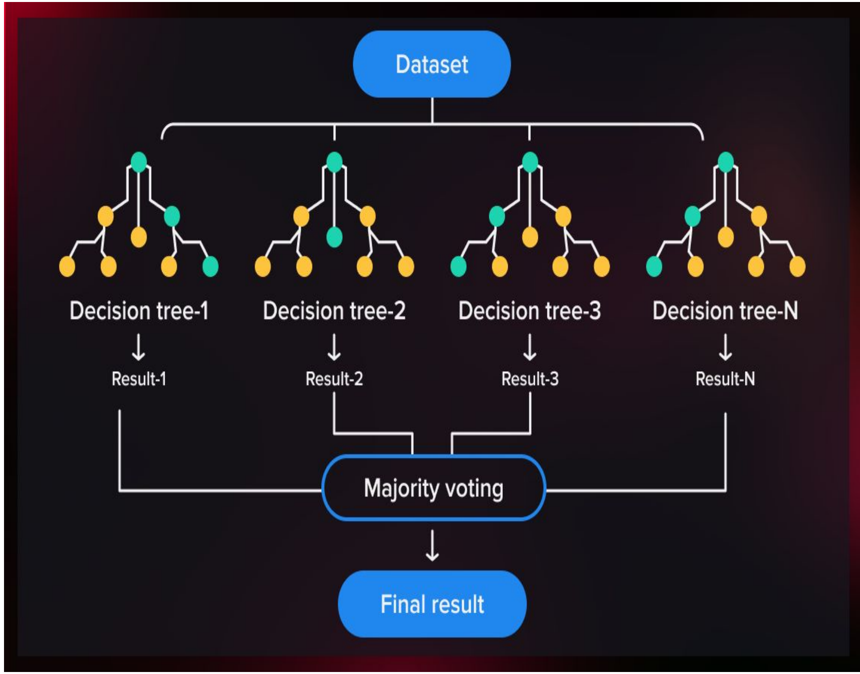
x is the independent variable

y is the dependent variable

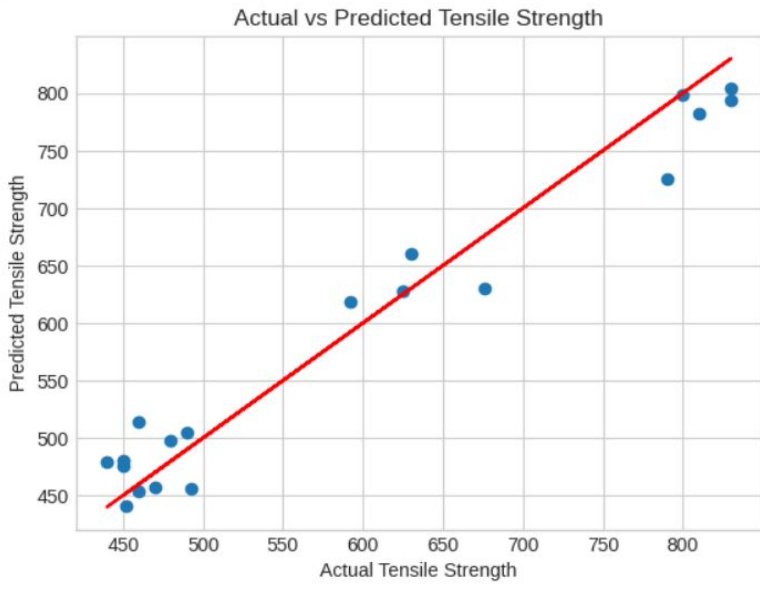


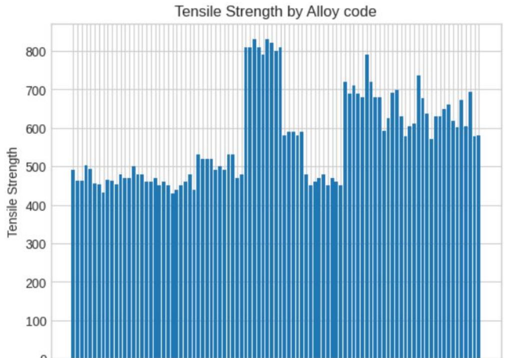
Random Forest

- It can be used for both Classification and Regression problems in ML.
- It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the mode.
- Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset**.**
- It takes less training time as compared to other algorithms.It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.



Linear regression -:Results /Conclusions/ Analysis





The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 336.82

For Target as 0.2% Proof Stress (MPa) :

Mean squared error: 989.00

For Target as Elongation :

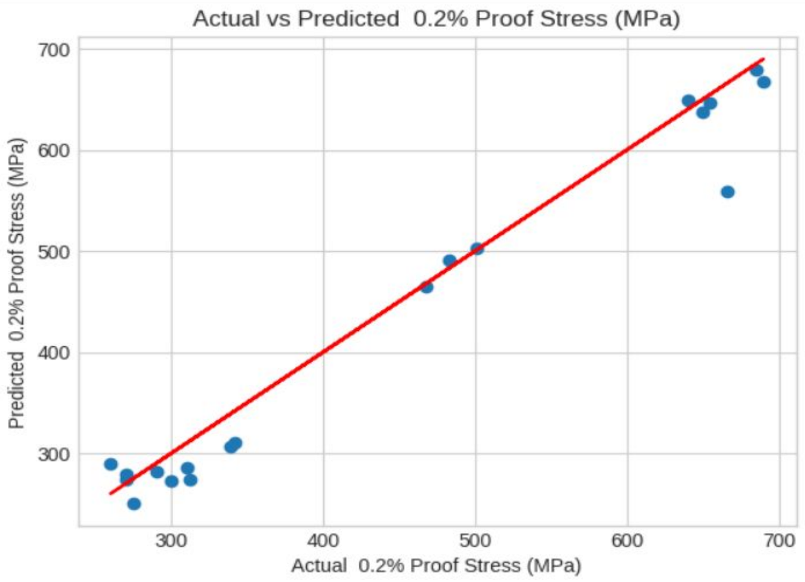
Mean squared error: 2.96

For Target as Reduction in Area (%) :

Mean squared error: 13.22

Linear Regression is best fit for Tensile strength , 0.2% Proof stress ,Elongation as it is numerical data type. And here the relationship is linear between independent variables and target variables, so it is best fit for our dataset.

Linear regression -:Results /Conclusions/ Analysis



The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 336.82

For Target as 0.2% Proof Stress (MPa) :

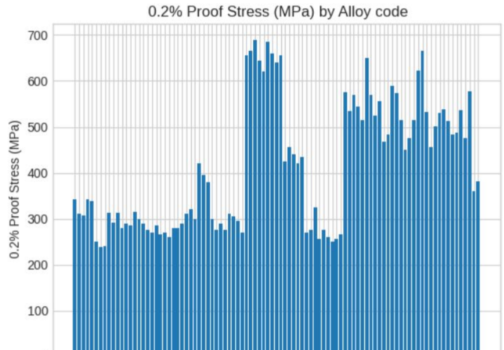
Mean squared error: 989.00

For Target as Elongation :

Mean squared error: 2.96

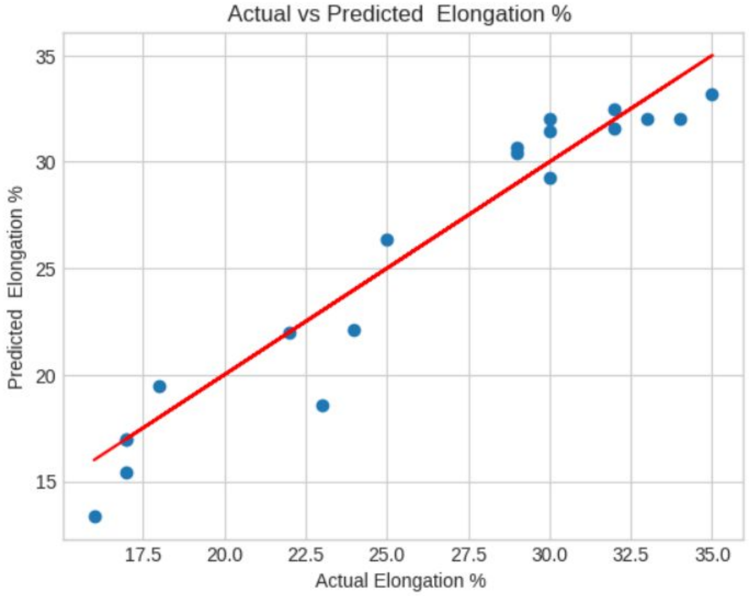
For Target as Reduction in Area (%) :

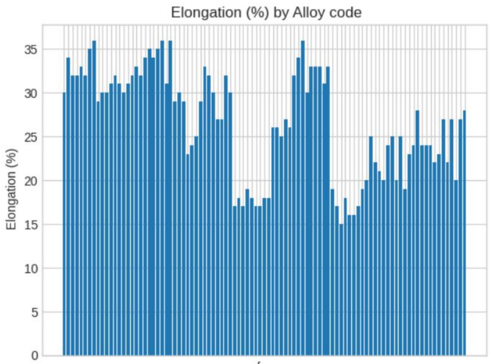
Mean squared error: 13.22



Linear Regression is best fit for Tensile strength , 0.2% Proof stress ,Elongation as it is numerical data type. And here the relationship is linear between independent variables and target variables, so it is best fit for our dataset.

Linear regression -:Results /Conclusions/ Analysis





The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 336.82

For Target as 0.2% Proof Stress (MPa) :

Mean squared error: 989.00

For Target as Elongation :

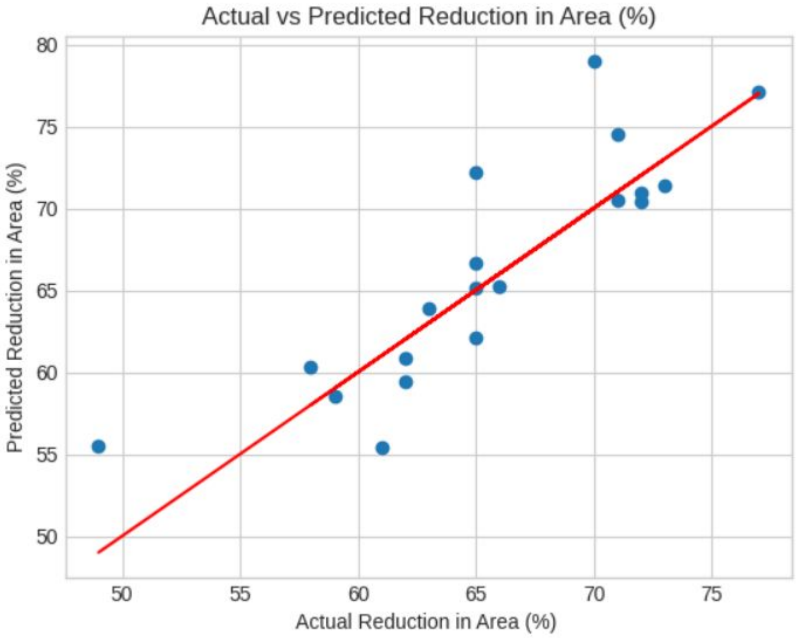
Mean squared error: 2.96

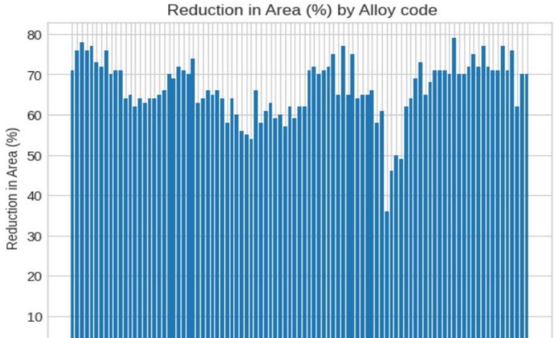
For Target as Reduction in Area (%) :

Mean squared error: 13.22

Linear Regression is best fit for Tensile strength , 0.2% Proof stress ,Elongation as it is numerical data type. And here the relationship is linear between independent variables and target variables, so it is best fit for our dataset.

Linear regression -:Results /Conclusions/ Analysis





The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 336.82

For Target as 0.2% Proof Stress (MPa) :

Mean squared error: 989.00

For Target as Elongation :

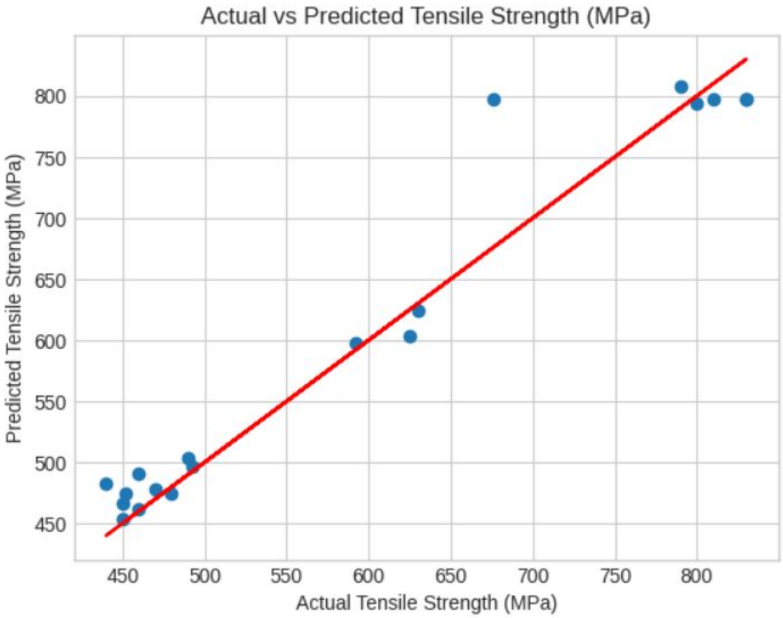
Mean squared error: 2.96

For Target as Reduction in Area (%) :

Mean squared error: 13.22

Linear Regression is best fit for Tensile strength , 0.2% Proof stress ,Elongation as it is numerical data type. And here the relationship is linear between independent variables and target variables, so it is best fit for our dataset.

K-Nearest Neighbour : Results /Conclusions/ Analysis



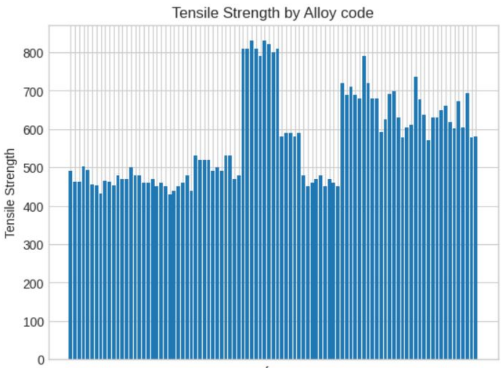
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 1143.39

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 1537.73

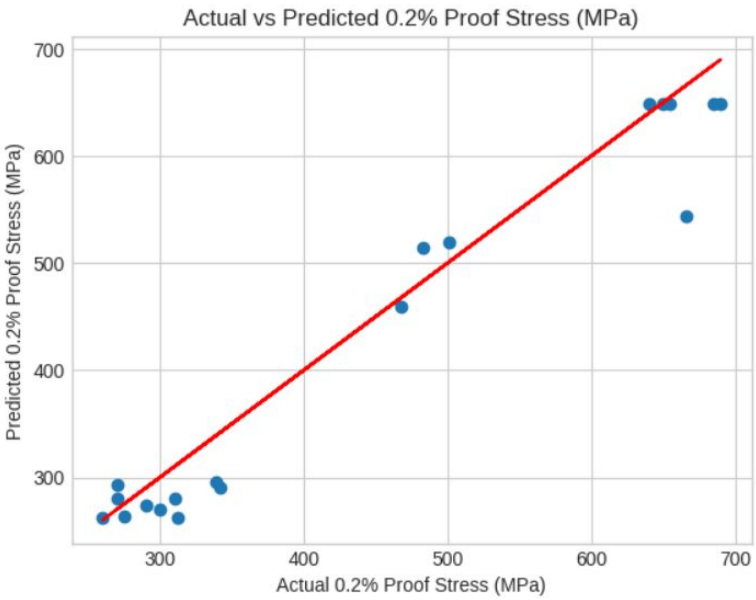
For Target as Elongation :
Mean squared error: 1.20

For Target as Reduction in Area (%) :
Mean squared error: 6.98



KNN makes predictions based on the local neighborhood of the data point, which can be beneficial in cases where the relationship between the features and the target variable may vary locally, such as in spatial or temporal data, that's why it is giving almost high accuracy.

K-Nearest Neighbour : Results /Conclusions/ Analysis



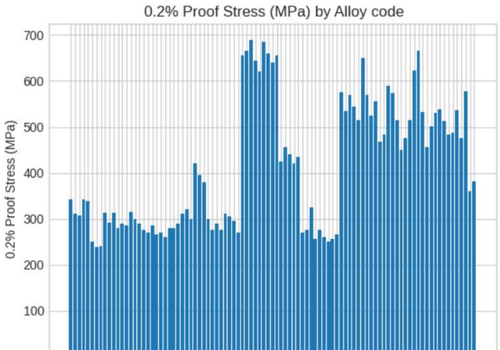
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 1143.39

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 1537.73

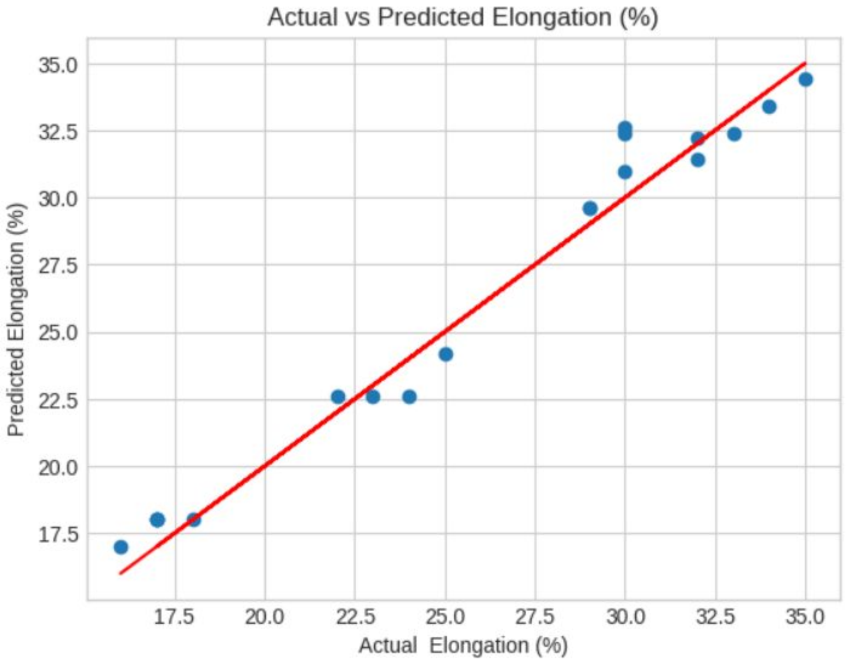
For Target as Elongation :
Mean squared error: 1.20

For Target as Reduction in Area (%) :
Mean squared error: 6.98



KNN makes predictions based on the local neighborhood of the data point, which can be beneficial in cases where the relationship between the features and the target variable may vary locally, such as in spatial or temporal data, that's why it is giving almost high accuracy.

K-Nearest Neighbour : Results /Conclusions/ Analysis



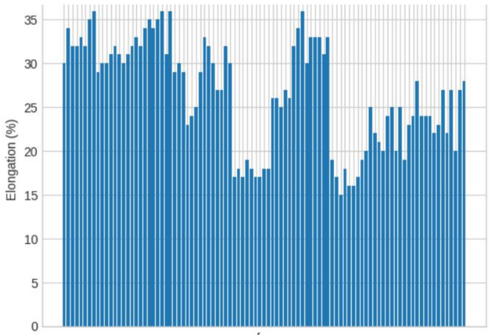
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 1143.39

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 1537.73

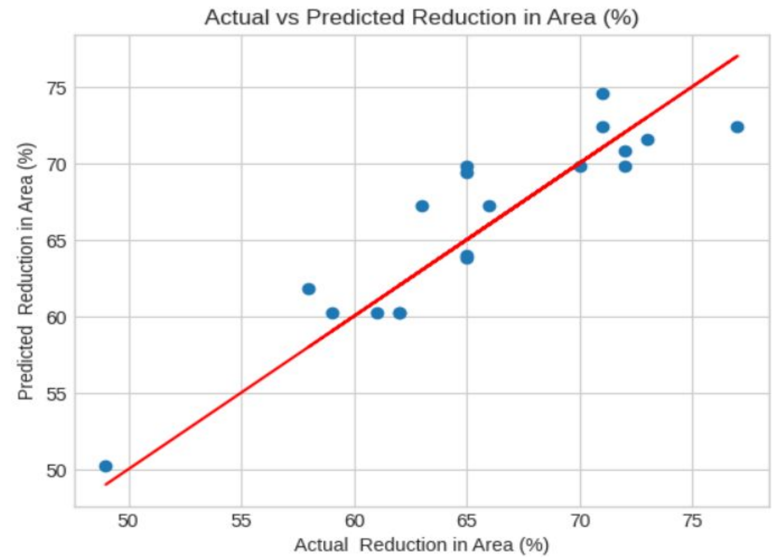
For Target as Elongation :
Mean squared error: 1.20

For Target as Reduction in Area (%) :
Mean squared error: 6.98



KNN makes predictions based on the local neighborhood of the data point, which can be beneficial in cases where the relationship between the features and the target variable may vary locally, such as in spatial or temporal data, that’s why it is giving almost high accuracy.

K-Nearest Neighbour : Results /Conclusions/ Analysis



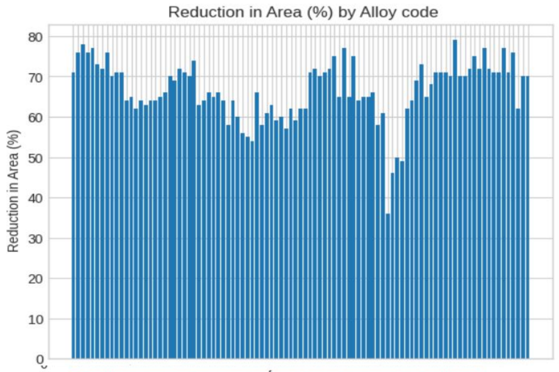
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 1143.39

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 1537.73

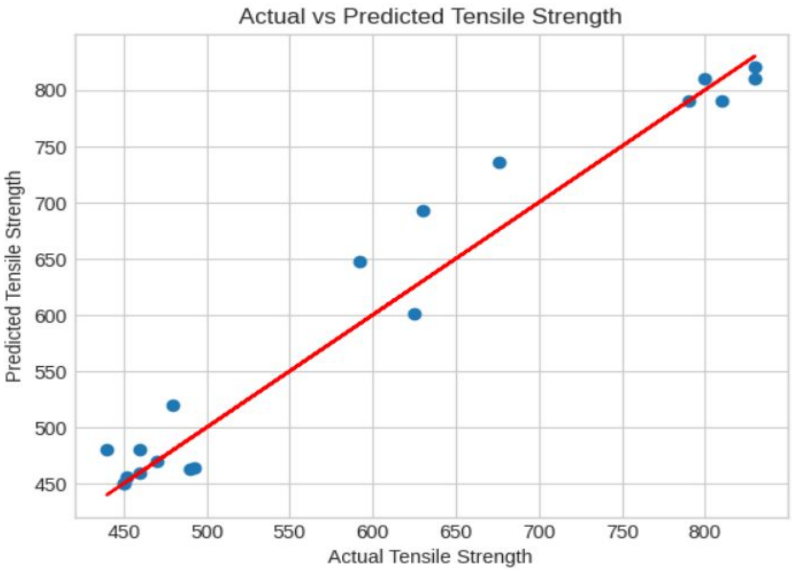
For Target as Elongation :
Mean squared error: 1.20

For Target as Reduction in Area (%) :
Mean squared error: 6.98



KNN makes predictions based on the local neighborhood of the data point, which can be beneficial in cases where the relationship between the features and the target variable may vary locally, such as in spatial or temporal data, that's why it is giving almost high accuracy.

Decision Tree : Results /Conclusions/ Analysis



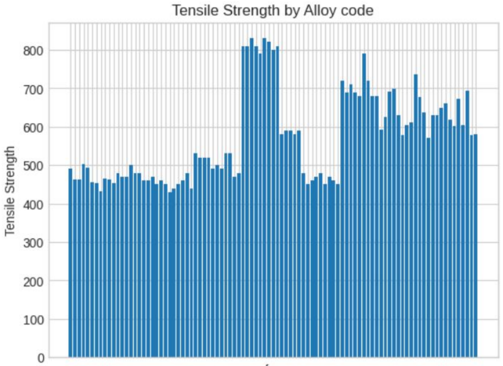
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 919.31

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 2593.84

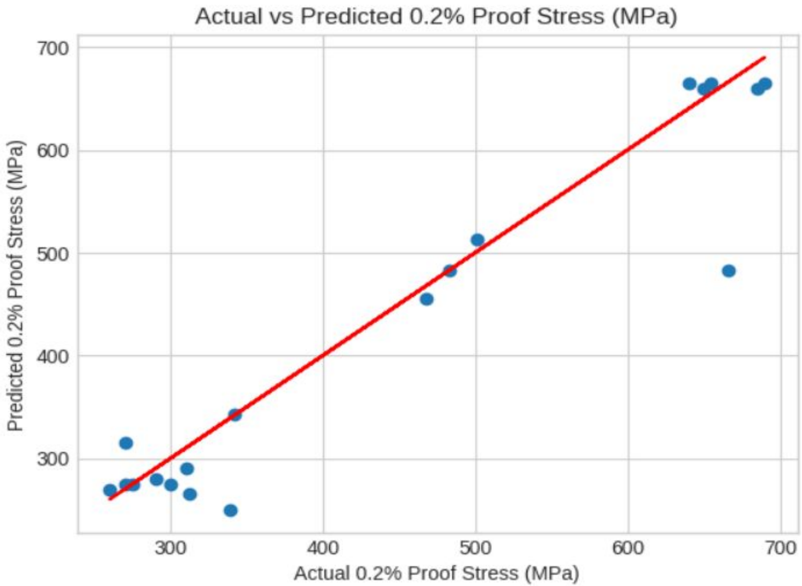
For Target as Elongation :
Mean squared error: 4.21

For Target as Reduction in Area (%) :
Mean squared error: 10.84



Decision Trees can split the data based on different values of variables at each node, allowing them to capture non-linear patterns in the data. Decision Trees are computationally efficient and can handle large datasets with multiple features. This makes them suitable for datasets with a large number of observations and variables.

Decision Tree : Results /Conclusions/ Analysis



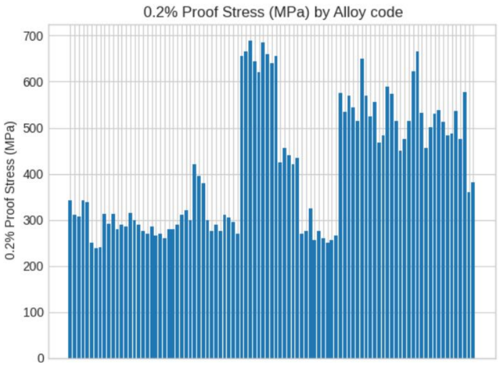
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 919.31

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 2593.84

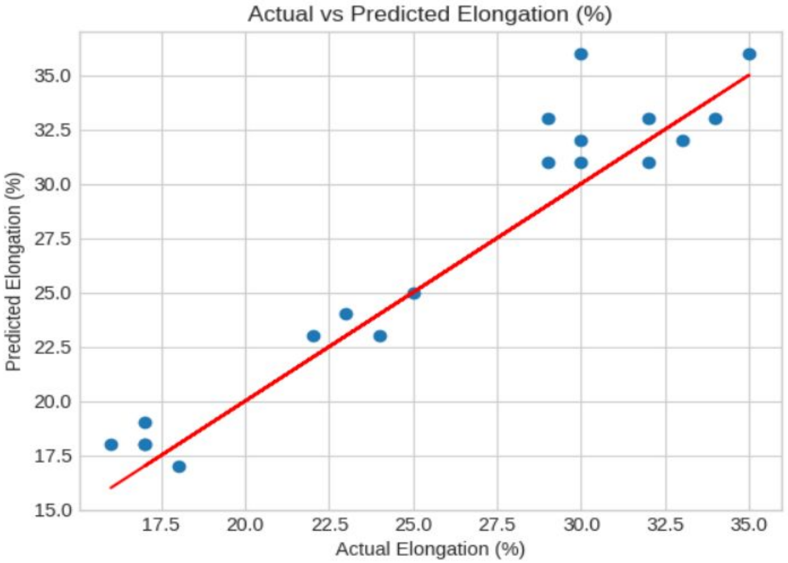
For Target as Elongation :
Mean squared error: 4.21

For Target as Reduction in Area (%) :
Mean squared error: 10.84



Decision Trees can split the data based on different values of variables at each node, allowing them to capture non-linear patterns in the data. Decision Trees are computationally efficient and can handle large datasets with multiple features. This makes them suitable for datasets with a large number of observations and variables.

Decision Tree : Results /Conclusions/ Analysis



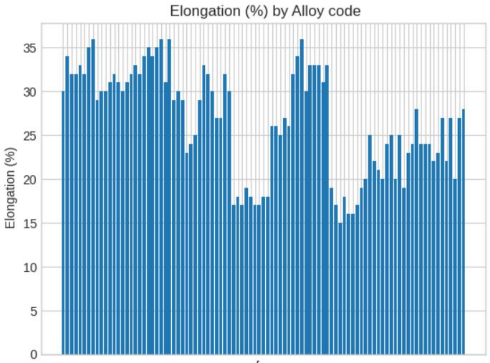
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 919.31

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 2593.84

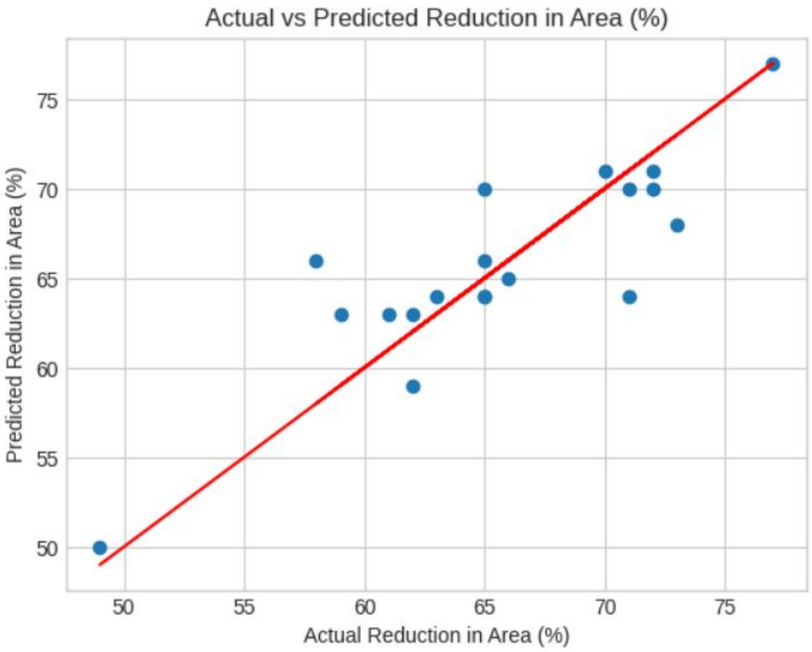
For Target as Elongation :
Mean squared error: 4.21

For Target as Reduction in Area (%) :
Mean squared error: 10.84



Decision Trees can split the data based on different values of variables at each node, allowing them to capture non-linear patterns in the data. Decision Trees are computationally efficient and can handle large datasets with multiple features. This makes them suitable for datasets with a large number of observations and variables.

Decision Tree : Results /Conclusions/ Analysis



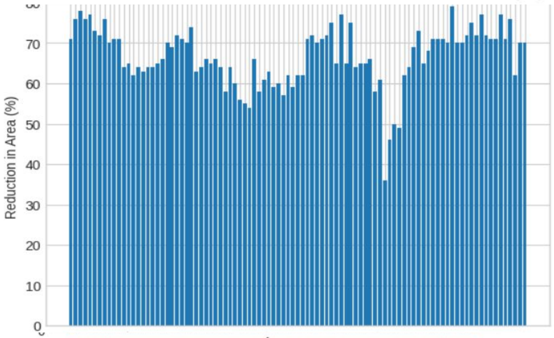
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 919.31

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 2593.84

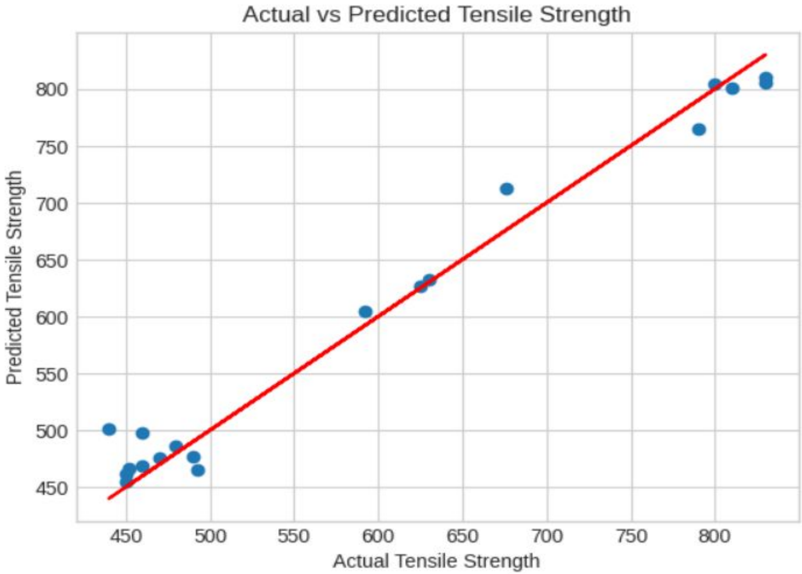
For Target as Elongation :
Mean squared error: 4.21

For Target as Reduction in Area (%) :
Mean squared error: 10.84



Decision Trees can split the data based on different values of variables at each node, allowing them to capture non-linear patterns in the data. Decision Trees are computationally efficient and can handle large datasets with multiple features. This makes them suitable for datasets with a large number of observations and variables.

Random Forest : Results /Conclusions/ Analysis



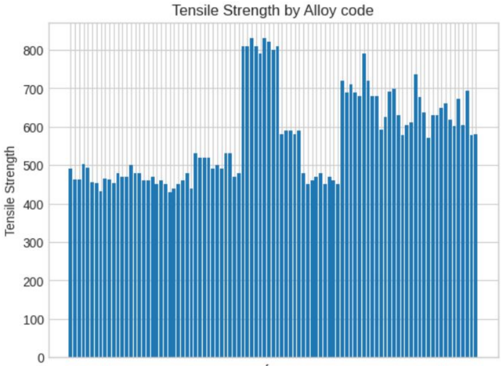
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 522.26

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 1747.13

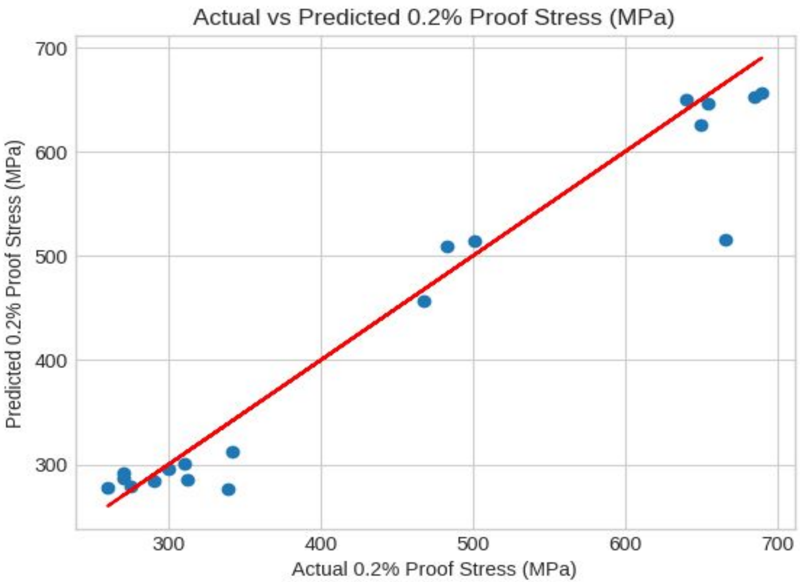
For Target as Elongation :
Mean squared error: 2.59

For Target as Reduction in Area (%) :
Mean squared error: 5.91



Random Forest can effectively handle datasets with a large number of features or high-dimensional data. Outliers or noisy data points are less likely to have a significant impact on the overall prediction, as the algorithm takes a consensus of predictions from multiple trees. Can be useful in datasets where the relationships between features are non-linear. Resilience to overfitting: Random Forest is less prone to overfitting compared to a single decision tree

Random Forest : Results /Conclusions/ Analysis



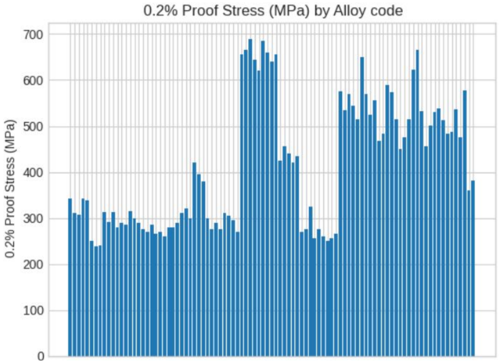
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 522.26

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 1747.13

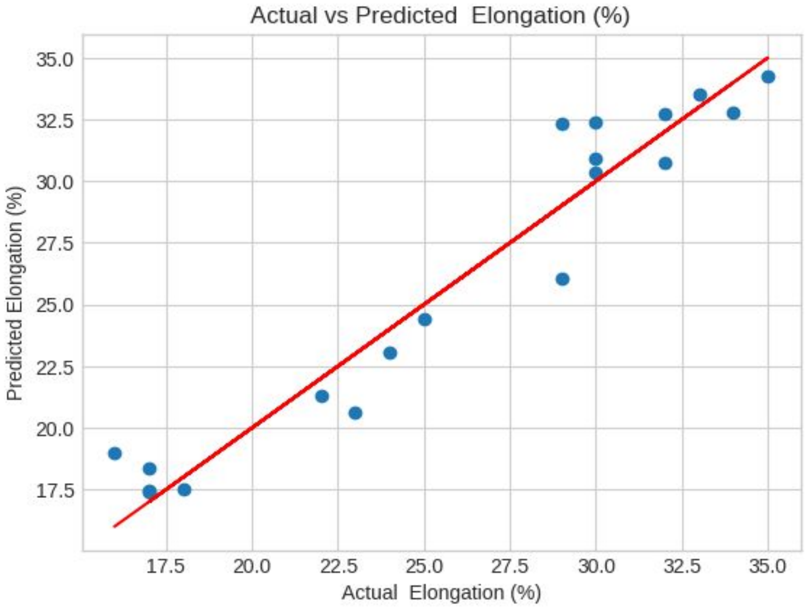
For Target as Elongation :
Mean squared error: 2.59

For Target as Reduction in Area (%) :
Mean squared error: 5.91



Random Forest can effectively handle datasets with a large number of features or high-dimensional data. Outliers or noisy data points are less likely to have a significant impact on the overall prediction, as the algorithm takes a consensus of predictions from multiple trees. Can be useful in datasets where the relationships between features are non-linear. Resilience to overfitting: Random Forest is less prone to overfitting compared to a single decision tree

Random Forest : Results /Conclusions/ Analysis



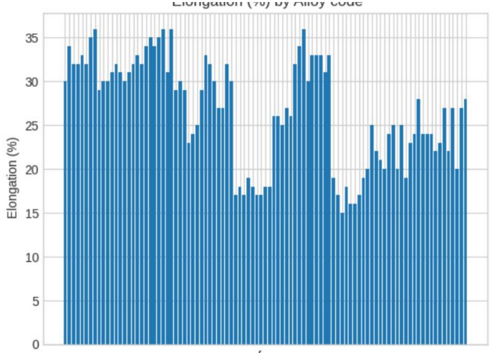
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 522.26

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 1747.13

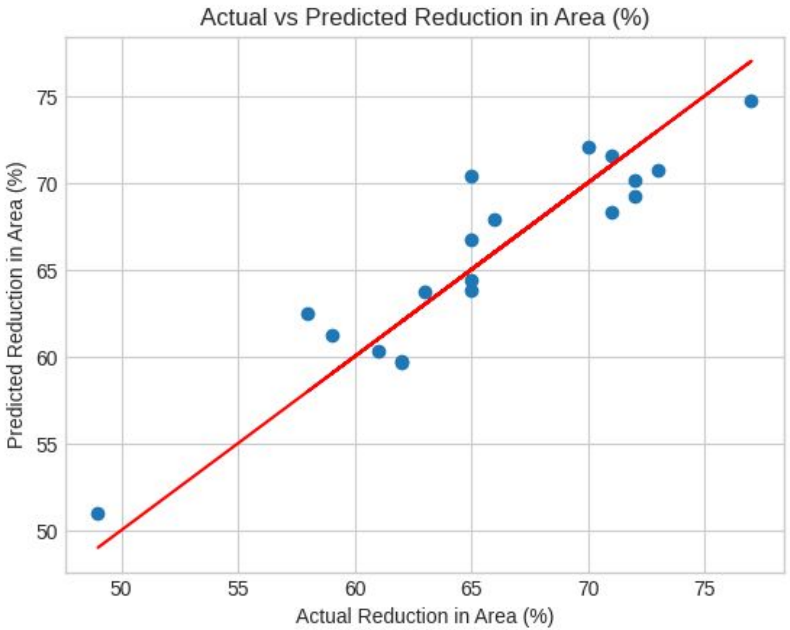
For Target as Elongation :
Mean squared error: 2.59

For Target as Reduction in Area (%) :
Mean squared error: 5.91



Random Forest can effectively handle datasets with a large number of features or high-dimensional data. Outliers or noisy data points are less likely to have a significant impact on the overall prediction, as the algorithm takes a consensus of predictions from multiple trees. Can be useful in datasets where the relationships between features are non-linear. Resilience to overfitting: Random Forest is less prone to overfitting compared to a single decision tree

Random Forest : Results /Conclusions/ Analysis



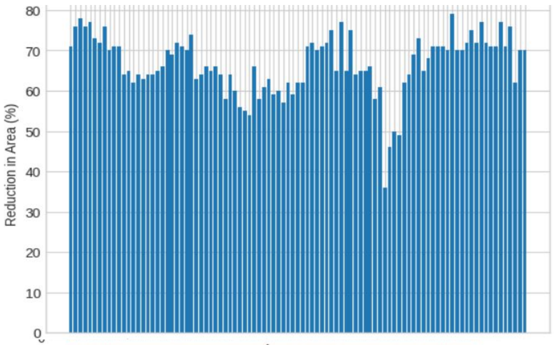
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 522.26

For Target as **0.2% Proof Stress (MPa) :**
Mean squared error: 1747.13

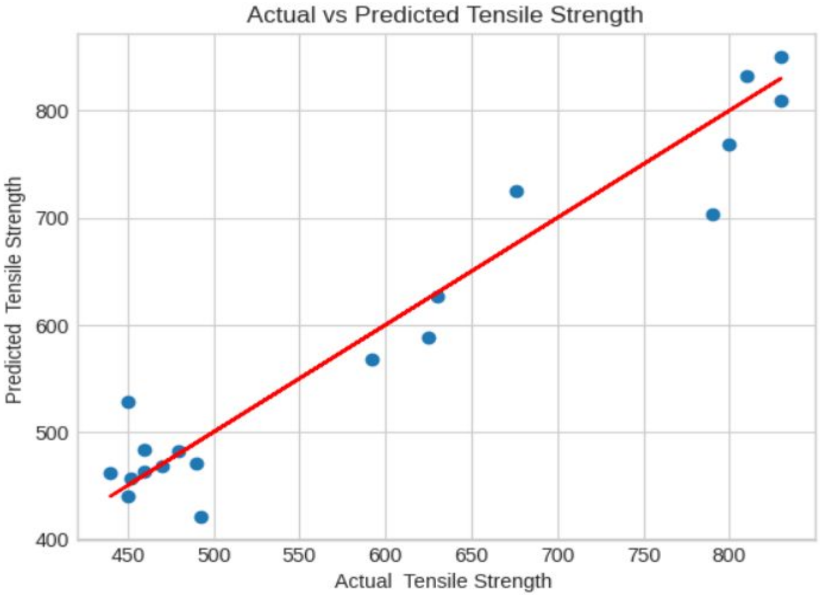
For Target as **Elongation :**
Mean squared error: 2.59

For Target as **Reduction in Area (%) :**
Mean squared error: 5.91



Random Forest can effectively handle datasets with a large number of features or high-dimensional data. Outliers or noisy data points are less likely to have a significant impact on the overall prediction, as the algorithm takes a consensus of predictions from multiple trees. Can be useful in datasets where the relationships between features are non-linear. Resilience to overfitting: Random Forest is less prone to overfitting compared to a single decision tree

Artificial Neural Network: Results /Conclusions/ Analysis



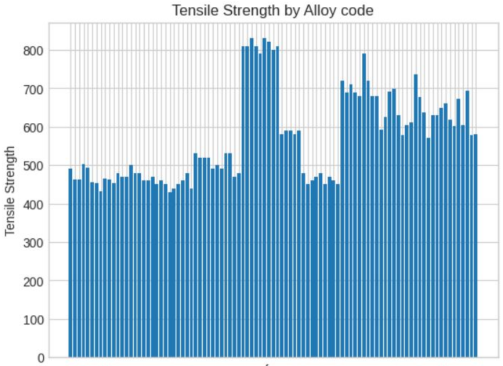
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 1421.07

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 2175.91

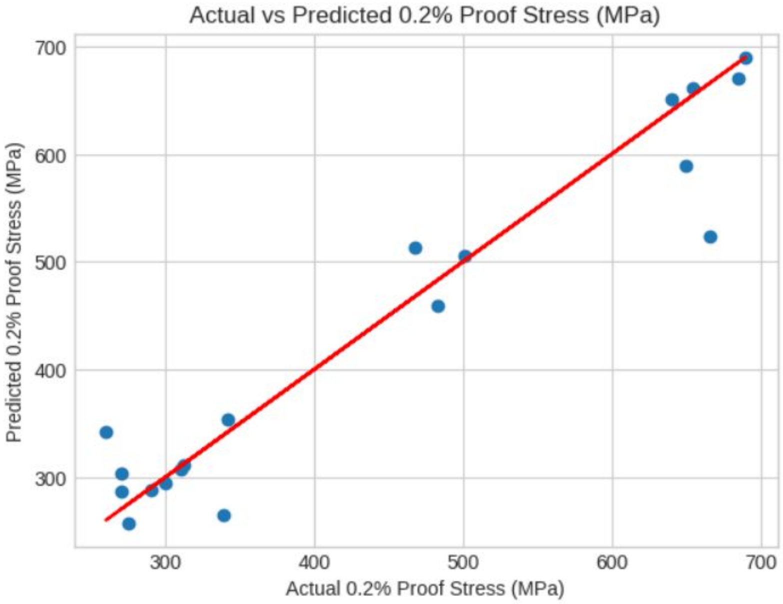
For Target as Elongation :
Mean squared error: 3.55

For Target as Reduction in Area (%) :
Mean squared error: 16.62



ANNs can adapt to different types of data, including high-dimensional data with multiple input features.ANNs generally perform well with large datasets, but our data set is not much large so it reduces it accuracy in prediction . If our alloy dataset is large, ANNs can potentially learn the underlying patterns in the data and provide accurate predictions.

Artificial Neural Network: Results /Conclusions/ Analysis



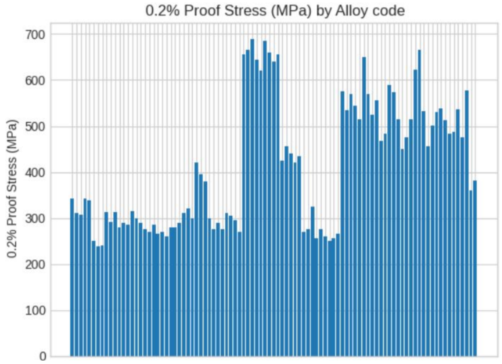
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 1421.07

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 2175.91

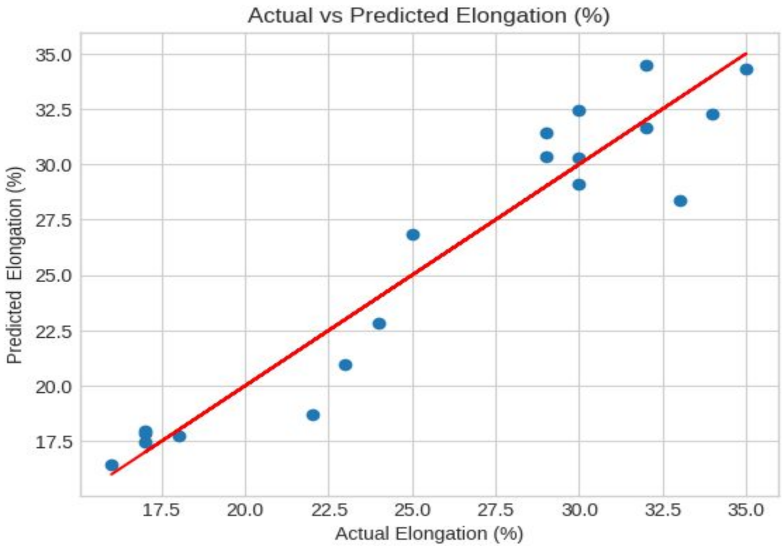
For Target as Elongation :
Mean squared error: 3.55

For Target as Reduction in Area (%) :
Mean squared error: 16.62



ANNs can adapt to different types of data, including high-dimensional data with multiple input features.ANNs generally perform well with large datasets, but our data set is not much large so it reduces it accuracy in prediction . If our alloy dataset is large, ANNs can potentially learn the underlying patterns in the data and provide accurate predictions.

Artificial Neural Network: Results /Conclusions/ Analysis



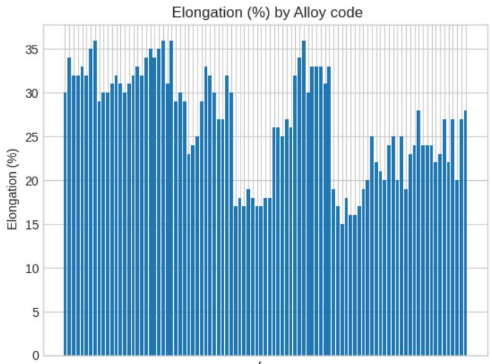
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 1421.07

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 2175.91

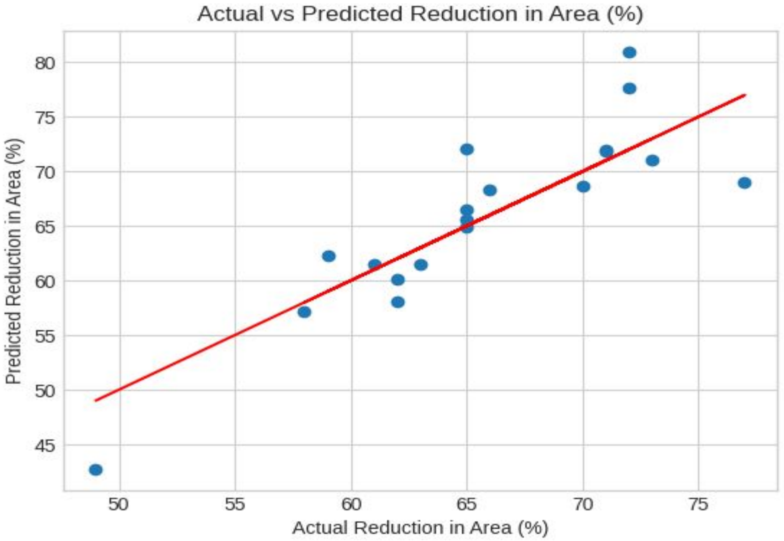
For Target as Elongation :
Mean squared error: 3.55

For Target as Reduction in Area (%) :
Mean squared error: 16.62



ANNs can adapt to different types of data, including high-dimensional data with multiple input features.ANNs generally perform well with large datasets, but our data set is not much large so it reduces it accuracy in prediction . If our alloy dataset is large, ANNs can potentially learn the underlying patterns in the data and provide accurate predictions.

Artificial Neural Network: Results /Conclusions/ Analysis



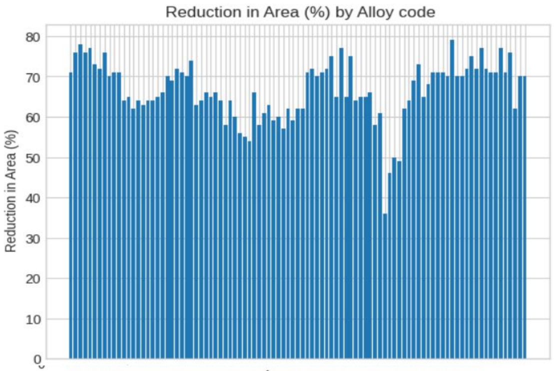
The Mean Squared Error(MSE), measures how close a regression line is to a set of data points. MSE tends to zero is super best for our model.

For Target as Tensile Strength :
Mean squared error: 1421.07

For Target as 0.2% Proof Stress (MPa) :
Mean squared error: 2175.91

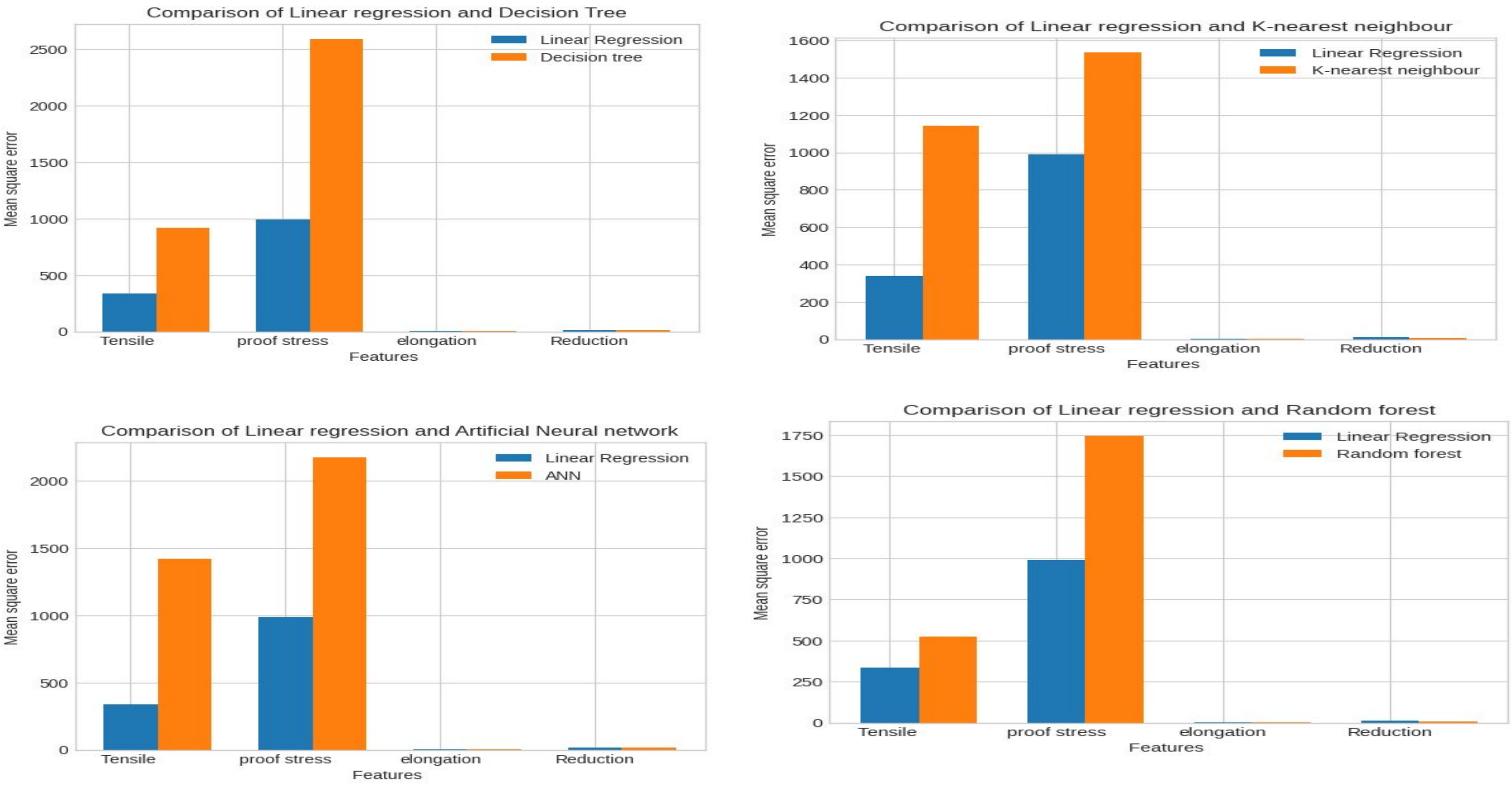
For Target as Elongation :
Mean squared error: 3.55

For Target as Reduction in Area (%) :
Mean squared error: 16.62



ANNs can adapt to different types of data, including high-dimensional data with multiple input features.ANNs generally perform well with large datasets, but our data set is not much large so it reduces it accuracy in prediction . If our alloy dataset is large, ANNs can potentially learn the underlying patterns in the data and provide accurate predictions.

Results : Comparison of different ML model using barplot, Linear regression is giving **minimum mse**



Different Hyper-Tuning Parameters

Hyperparameter tuning is critical for **optimizing** model performance in machine learning. Hyperparameters, set before training, control model behavior and performance. Example :

Learning rate in machine learning controls the step size for parameter adjustments during training. It impacts learning speed and optimal solution attainment, making it a crucial hyperparameter to tune for model performance.

Regularization parameters control the strength of regularization in machine learning models to prevent overfitting. Higher regularization strengths increase regularization but may impact model performance, making proper tuning important.

And other parameters can be used as Number of epochs, Batch size,Activation function etc , now we see them model wise parameters -

Linear Regression:

- Regularization strength: The strength of regularization to prevent overfitting, such as L1 (Lasso) or L2 (Ridge) regularization.
- Learning rate: The step size used for updating the model parameters during training.

Random Forest:

- Number of trees: The number of decision trees to be included in the random forest ensemble.
- Maximum depth: The maximum depth of each decision tree in the random forest.
- Number of features to consider for splitting: The number of features randomly selected for splitting at each node in a decision tree.

Different Hyper-Tuning Parameters

K-Nearest Neighbors (KNN):

- Number of neighbors (k): The number of nearest neighbors to consider when making predictions.
- Distance metric: The metric used to measure the distance between data points, such as Euclidean, Manhattan, or Minkowski distance.
- Weighting scheme: The scheme used to weight the contribution of neighbors, such as uniform weights or distance-based weights.

Decision Tree:

- Maximum depth: The maximum depth of the decision tree.
- Split criterion: The criterion used for splitting at each node, such as Gini impurity or entropy.
- Minimum samples for split: The minimum number of samples required at a node to consider splitting.

Artificial Neural Network (ANN):

- Number of layers: The number of hidden layers in the neural network.
- Number of neurons per layer: The number of neurons in each hidden layer.
- Activation functions: The activation functions used in each layer, such as sigmoid, tanh, or ReLU.
- Learning rate: The step size used for updating the model parameters during training.
- Regularization strength: The strength of regularization to prevent overfitting, such as L1 or L2 regularization.

Linear regression - Hypertunning

- Splitted our data set before training into 0.1, 0.2, 0.3 and 0.4 and changes random state from 42 to 123 respectively, for target value as Tensile strength.
- Uses 10% for testing and 90% for training
  With Random state = 42
  Mean squared error: 410.92
  Coefficient of determination (R^2): 0.98
  With Random state = 123
  Mean squared error: 225.27
  Coefficient of determination (R^2): 0.96
- Uses 30% for testing and 70% for training
  With Random state = 42
  Mean squared error: 345.98
  Coefficient of determination (R^2): 0.98
  With Random state = 123
  Mean squared error: 298.69
  Coefficient of determination (R^2): 0.97
- Uses 40% for testing and 60% for training
  With Random state = 42
  Mean squared error: 295.64
  Coefficient of determination (R^2): 0.98
  With Random state = 123
  Mean squared error: 229.34
  Coefficient of determination (R^2): 0.98

Linear regression - Hypertunning

- Figured out that, using 123 as random states, reduces Mean squared error but Decreases accuracy littlebit, as compared to random state as 42.
- We are getting best R2 score and less mean score error for 40/60 split into testing and training (0.98, 295.64),with random state as 42.
- Performed hyperparameter tuning for regularization strength (alpha) using **Lasso regression**, from this we get best Regularization strength (alpha) ,alpha = 0.1 .It gives **:**

  Best_alpha is : 0.1

  Mean Squared Error: 497.3495

  Coefficient of determination (R^2): 0.98

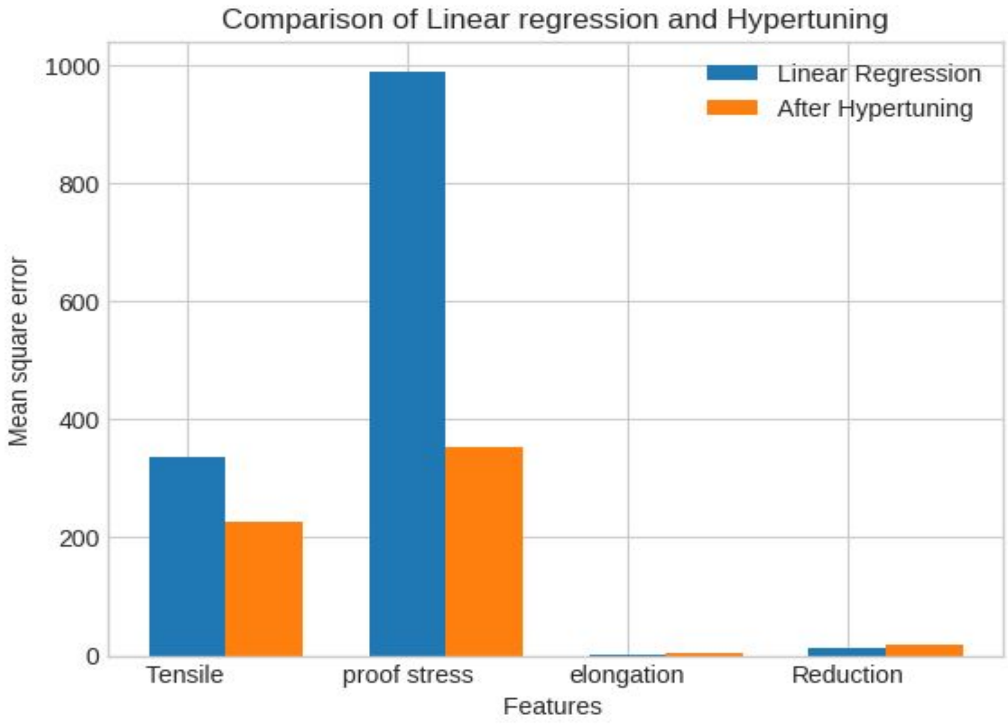
- With Ridge regression we get alpha , mse and R2 as -

  Best_alpha is : 0.1

  Mean Squared Error: 591.0285

  Coefficient of determination (R^2): 0.97

Linear regression after Hypertuning



- Tensile strength from 336.82 → 225.27
- 0.2% yield strength from 989.00 → 353.01

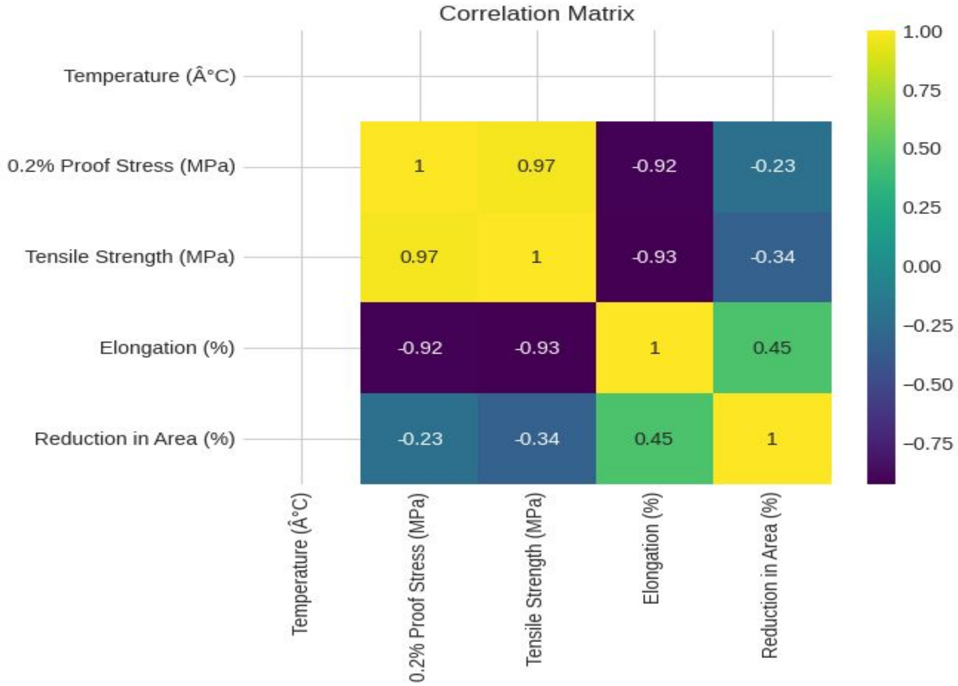
Final Conclusion

- After applying nine machine learning algorithms on my dataset like K-NN, Linear regression, Naive bayes, ANN, SVM, RNN, decision tree, Random forest. I observed that linear regression is giving me best results and best predicted values, followed by KNN and Random forest, i got very good R-squared value.
- Other machine learning model like A
- After performing HyperTuning parameters i am able to reduce Mean square error to some extent from 336.82 to 225.27 .

Google colab link

[DC(Design_of_AHSS_using_ML)_B21MT035.ipynb - Colaboratory (google.com)]

Thank you


Correlation Matrix
Temperature (Â°C)
0.2% Proof Stress (MPa)
Tensile Strength (MPa)
Elongation (%)
Reduction in Area (%)
1
0.97
-0.92
-0.23
0.97
1
-0.93
-0.34
-0.92
-0.93
1
0.45
-0.23
-0.34
0.45
1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

References -

Numpy , pandas , scikit-learn from -
https://towardsdatascience.com/numpy-and-pandas-for-data-scientists-2be4a093b4b5
https://www.analyticsvidhya.com/blog/2021/08/complete-guide-on-how-to-learn-scikit-learn-for-data-science/
https://www.w3schools.com/python/numpy/default.asp
Matplot learn from -
https://www.w3schools.com/python/matplotlib_pie_charts.asp
K-NN
https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
ANN
https://towardsdatascience.com/an-introduction-to-artificial-neural-networks-5d2e108ff2c3
Linear regression
https://www.javatpoint.com/linear-regression-in-machine-learning
Random Forest
https://www.javatpoint.com/machine-learning-random-forest-algorithm
Hyper-parameter optimization
(133) Hyperparameter optimization - YouTube
Analyticsvidhya : https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Java point : https://www.javatpoint.com/machine-learning-random-forest-algorithm

Nptel:
https://www.youtube.com/watch?v=_M-nDb0MIa4&list=PLyqSpQzTE6M-SISTunGRBRiZk7opYBf_K&ab_channel=NPTEL-NOCIITM