Machine Learning: Material Stiffness Predictor

Donaven Lobo

HW 2

ME 8813
Dr. Yan Wang | Dr. Surya Kalidindi
Georgia Institute of Technology
North Avenue, Atlanta, GA 30332

Table of Contents

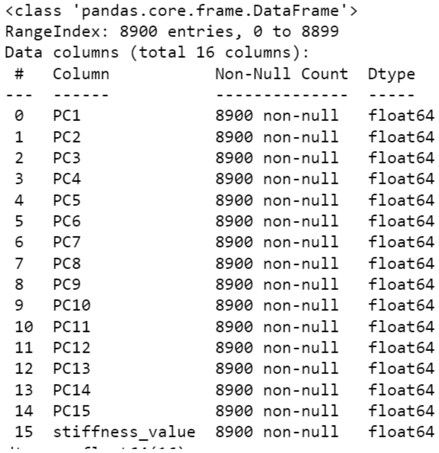
1. Introduction & Background

Predicting the mechanical properties of materials, such as stiffness, is essential for material selection and design in engineering. This study aims to develop machine learning models to predict the stiffness of materials based on their properties and composition. The dataset contains samples with features representing physical, chemical, and structural characteristics of the materials.

The objective is to compare different machine learning approaches, including linear regression, Gaussian process regression, and neural networks, to determine the most effective model for stiffness prediction. The models will be assessed for accuracy, generalization ability, and computational efficiency, with the findings contributing to advancements in predictive modeling for material science.

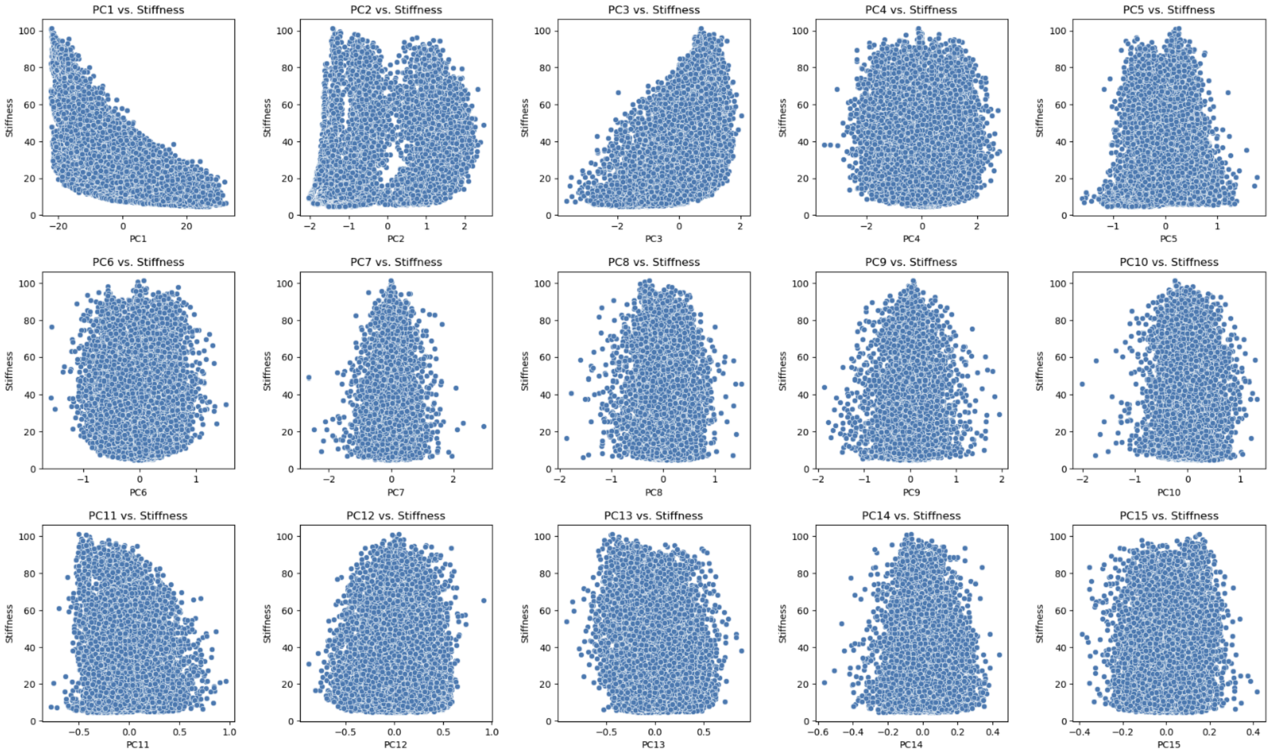
2. Describing & Visualizing the Dataset (EDA)

The dataset is made up of 8900 samples, where each sample has 15 features that capture important aspects of the materials' structure. Alongside these features, there are stiffness values predicted through finite element analysis for each sample. These features are arranged in a way that suggests they have varying levels of influence on the predicted stiffness, with the expectation that some features are more informative than others. As seen in the figure on the right, the dataset also has a perfect fill-rate with no null values.

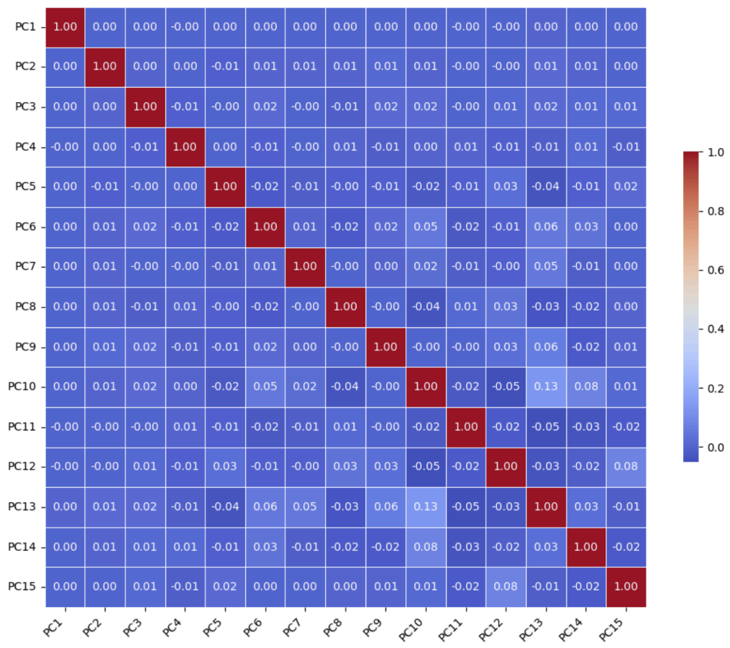


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8900 entries, 0 to 8899
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   PC1              8900 non-null   float64
 1   PC2              8900 non-null   float64
 2   PC3              8900 non-null   float64
 3   PC4              8900 non-null   float64
 4   PC5              8900 non-null   float64
 5   PC6              8900 non-null   float64
 6   PC7              8900 non-null   float64
 7   PC8              8900 non-null   float64
 8   PC9              8900 non-null   float64
 9   PC10             8900 non-null   float64
 10  PC11             8900 non-null   float64
 11  PC12             8900 non-null   float64
 12  PC13             8900 non-null   float64
 13  PC14             8900 non-null   float64
 14  PC15             8900 non-null   float64
 15  stiffness_value  8900 non-null   float64
```

Visual analysis through scatter plots shows diverse relationships between each feature and the material's stiffness, with some features showing a clear connection, while others do not. This initial exploration helps to anticipate the complexity of modeling the relationship between the features and stiffness, highlighting the need for careful selection of modeling techniques that can capture these intricate relationships. The scatter plot grid is depicted below.



Finally, a correlation heat map was applied to all the features of the dataset to investigate the relationships between them and to see if any could be omitted due to high correlation. There wasn't any high correlation between feature parameters as seen below.



	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15
PC1	1.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.00	-0.00	0.00	0.00	0.00
PC2	0.00	1.00	0.00	0.00	-0.01	0.01	0.01	0.01	0.01	0.01	-0.00	-0.00	0.01	0.01	0.00
PC3	0.00	0.00	1.00	-0.01	-0.00	0.02	-0.00	-0.01	0.02	0.02	-0.00	0.01	0.02	0.01	0.01
PC4	-0.00	0.00	-0.01	1.00	0.00	-0.01	-0.00	0.01	-0.01	0.00	0.01	-0.01	-0.01	0.01	-0.01
PC5	0.00	-0.01	-0.00	0.00	1.00	-0.02	-0.01	-0.00	-0.01	-0.02	-0.01	0.03	-0.04	-0.01	0.02
PC6	0.00	0.01	0.02	-0.01	-0.02	1.00	0.01	-0.02	0.02	0.05	-0.02	-0.01	0.06	0.03	0.00
PC7	0.00	0.01	-0.00	-0.00	-0.01	0.01	1.00	-0.00	0.00	0.02	-0.01	-0.00	0.05	-0.01	0.00
PC8	0.00	0.01	-0.01	0.01	-0.00	-0.02	-0.00	1.00	-0.00	-0.04	0.01	0.03	-0.03	-0.02	0.00
PC9	0.00	0.01	0.02	-0.01	-0.01	0.02	0.00	-0.00	1.00	-0.00	-0.00	0.03	0.06	-0.02	0.01
PC10	0.00	0.01	0.02	0.00	-0.02	0.05	0.02	-0.04	-0.00	1.00	-0.02	-0.05	0.13	0.08	0.01
PC11	-0.00	-0.00	-0.00	0.01	-0.01	-0.02	-0.01	0.01	-0.00	-0.02	1.00	-0.02	-0.05	-0.03	-0.02
PC12	-0.00	-0.00	0.01	-0.01	0.03	-0.01	-0.00	0.03	0.03	-0.05	-0.02	1.00	-0.03	-0.02	0.08
PC13	0.00	0.01	0.02	-0.01	-0.04	0.06	0.05	-0.03	0.06	0.13	-0.05	-0.03	1.00	0.03	-0.01
PC14	0.00	0.01	0.01	0.01	-0.01	0.03	-0.01	-0.02	-0.02	0.08	-0.03	-0.02	0.03	1.00	-0.02
PC15	0.00	0.00	0.01	-0.01	0.02	0.00	0.00	0.00	0.01	0.01	-0.02	0.08	-0.01	-0.02	1.00

3. Linear Model: Stochastic Gradient Descent (SGD) Regressor

The Stochastic Gradient Descent (SGD) Regressor was chosen for its efficiency in handling large datasets and its flexibility in being able to accommodate different types of loss functions and penalties, making it suitable for various regression tasks. Additionally, I have been using it in my research and wanted to explore this model a bit further.

Parameter Selection:-

Loss Function: The choice of loss function was determined based on the nature of the regression problem. For example, 'squared_loss' was considered for its simplicity in ordinary least squares regression, while 'huber' was evaluated for its robustness to outliers.

Regularization (Penalty): L2 ('ridge'), L1 ('lasso'), and Elastic Net penalties were explored to prevent overfitting and to handle multicollinearity. The L1 penalty was particularly useful for feature selection due to its ability to produce sparse solutions.

Learning Rate and Initial Learning Rate (eta0): Different learning rate schedules ('constant', 'optimal', 'invscaling', 'adaptive') and initial learning rates were tested to ensure convergence and to control the step size in the gradient descent.

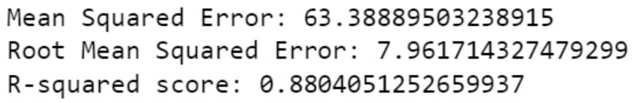
Alpha: The regularization strength was tuned to balance the trade-off between fitting the data well and keeping the model complexity low.

Hyperparameter Tuning:-

Bayesian Optimization was employed to systematically search for the optimal hyperparameters. This method was chosen for its efficiency in exploring the parameter space and its ability to incorporate prior knowledge about the hyperparameters, as well as it being susceptible to overfitting. Additionally this search employed a 5-fold cross validation during training.

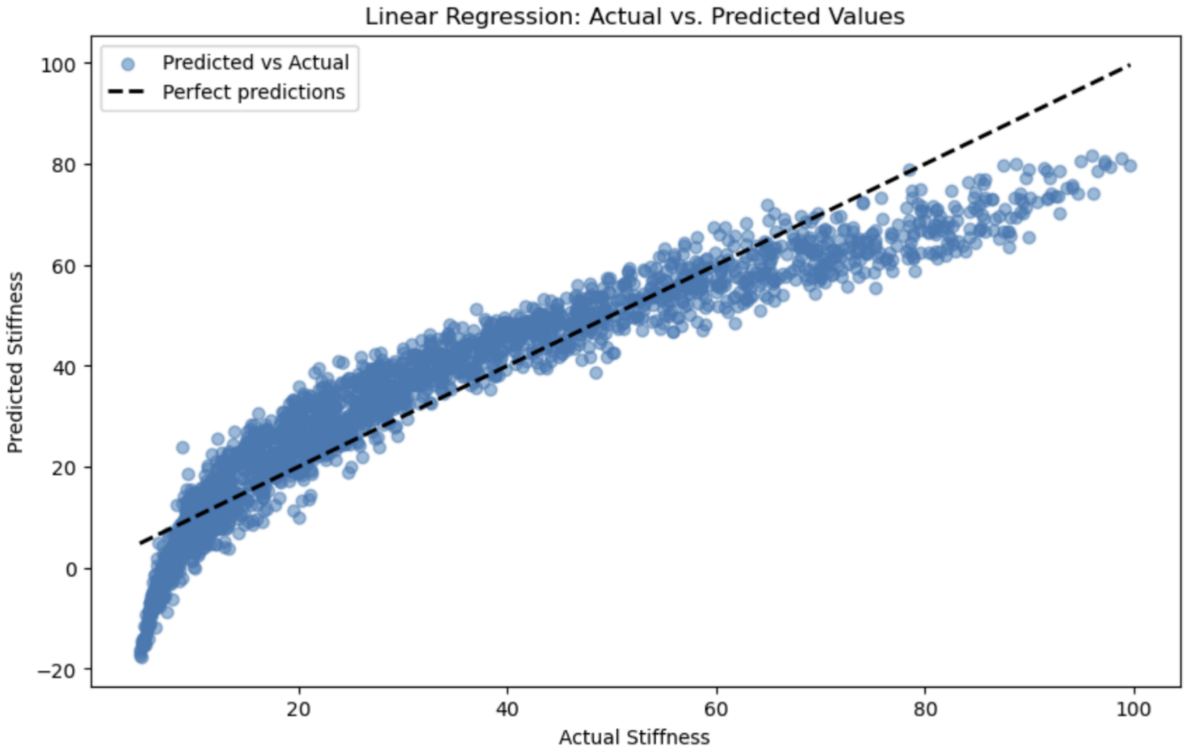
Results:-

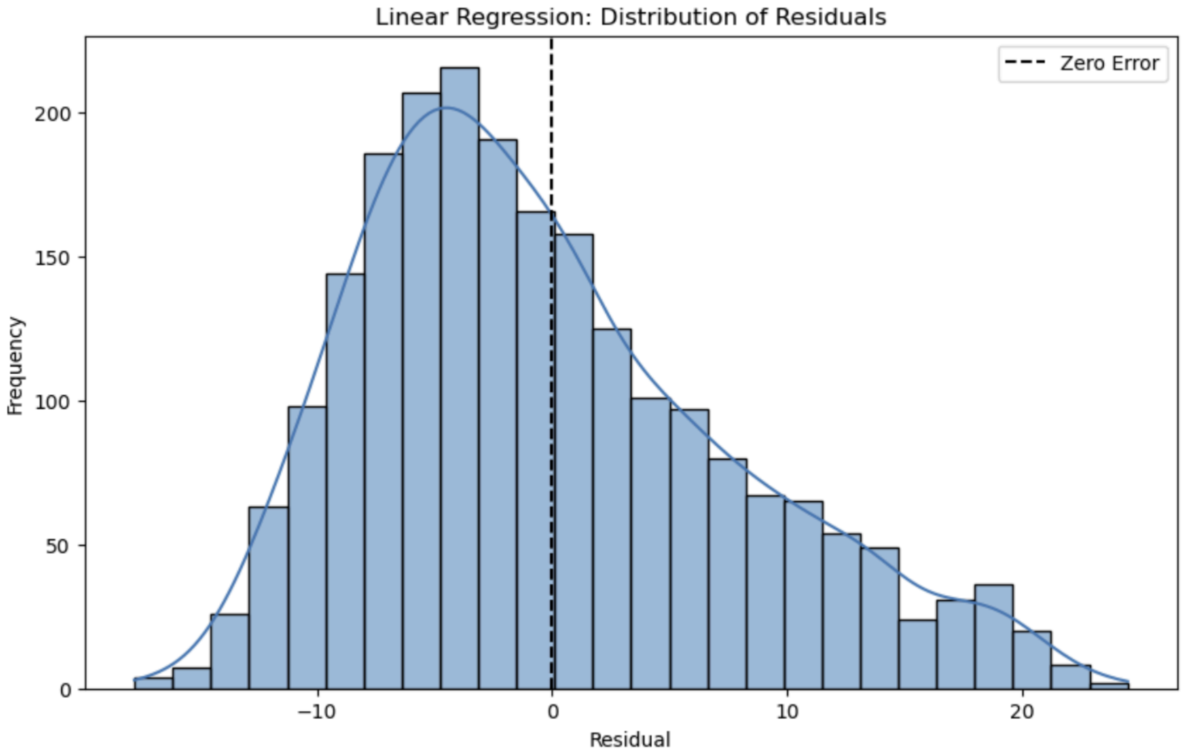
The model was evaluated using Mean Squared Error (MSE), Root Mean Square Error (RMSE), and R-squared metrics on a held-out test set. These model performances can be seen below.



```
Mean Squared Error: 63.38889503238915
Root Mean Squared Error: 7.961714327479299
R-squared score: 0.8804051252659937
```

The performance of the model was visualized through Actual vs. Predicted Values to see how predictions aligned to their true values. Residual plots were also created to assess the distribution of errors. These can be seen below.





4. Gaussian Process Regression:

Gaussian Process Regression (GPR) is the next model selected for its ability to provide a probabilistic approach to regression. It offers not only predictions but also a measure of uncertainty in those predictions. This is particularly valuable in scenarios where understanding the confidence in predictions is as important as the predictions themselves.

Kernel Selection:

RBF Kernel: The Radial Basis Function (RBF) kernel was chosen for its flexibility and the assumption of smoothness it imposes on the function being modeled. The RBF kernel is a common choice for GPR due to its property of mapping inputs into an infinite-dimensional space, allowing for the modeling of complex relationships.

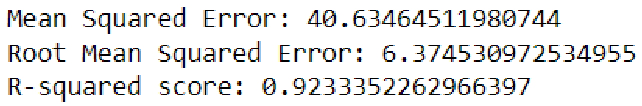
Simplification Decision: Initially, a combination of kernels including a Constant Kernel and White Kernel was considered to account for a non-zero mean and noise in the data, respectively. However, to reduce computational complexity and improve optimization speed, the decision was made to use only the RBF kernel in the final model.

Hyperparameter Tuning:

Bayesian Optimization was used again to determine the optimal length scale of the RBF kernel. This method was chosen for its efficiency in exploring the hyperparameter space and its ability to incorporate prior knowledge into the search. A 5-fold cross-validation approach was used during the optimization process to ensure that the model generalizes well to unseen data.

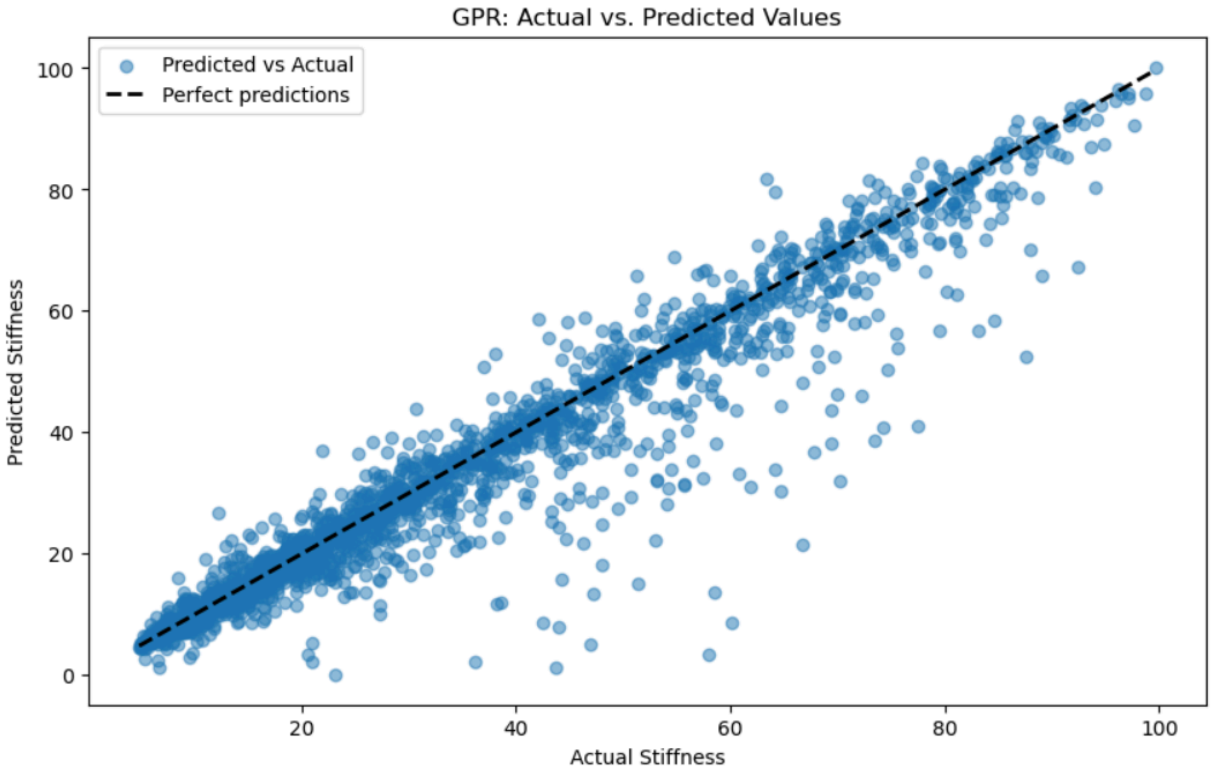
Results:-

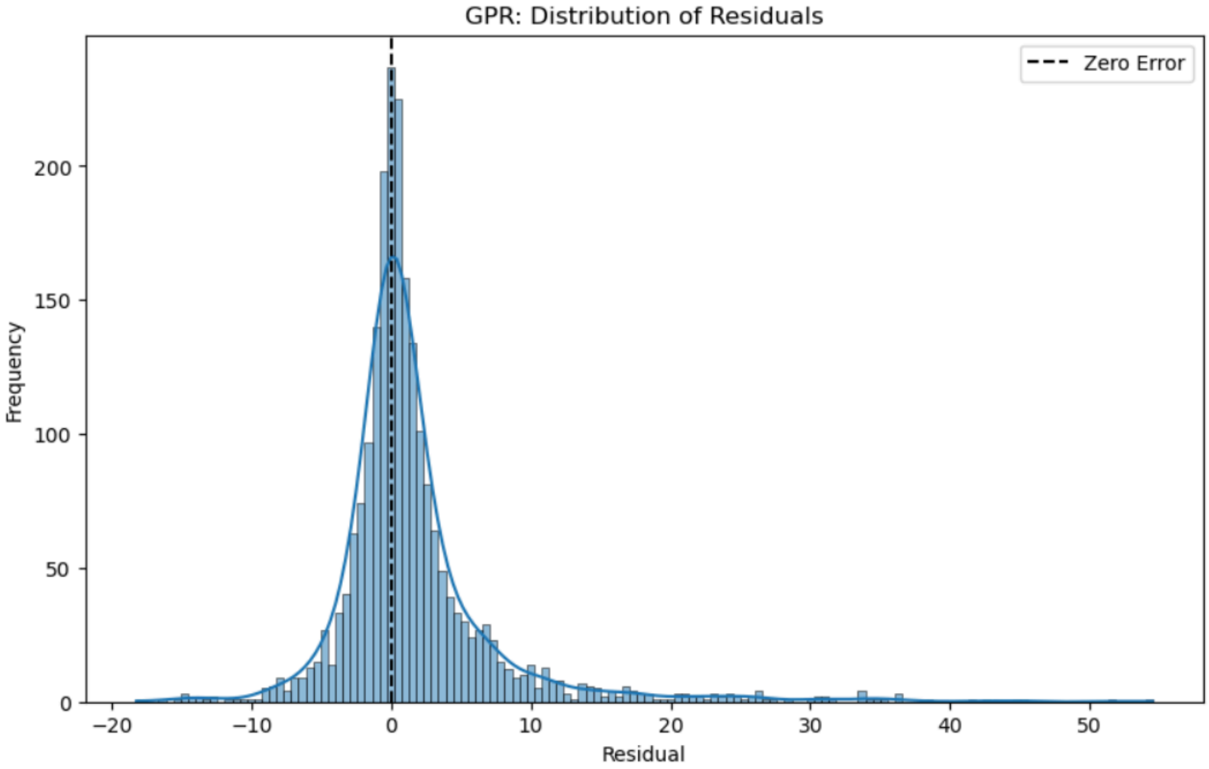
The model was evaluated using Mean Squared Error (MSE), Root Mean Square Error (RMSE), and R-squared metrics on a held-out test set. These model performances can be seen below.



```
Mean Squared Error: 40.63464511980744
Root Mean Squared Error: 6.374530972534955
R-squared score: 0.9233352262966397
```

The performance of the model was visualized through Actual vs. Predicted Values to see how predictions aligned to their true values. Residual plots were also created to assess the distribution of errors. These can be seen below.





4. Artificial Neural Network (ANN):

Next is an exploration of the application of Artificial Neural Networks (ANNs) for predicting material stiffness. Various network architectures are investigated, with adjustments made to the number of layers and neurons to identify the most effective model. Additionally, the issue of overfitting is addressed through the implementation of techniques such as regularization, dropout layers, batch normalization, and early stopping.

<u>*General Parameters:*</u>

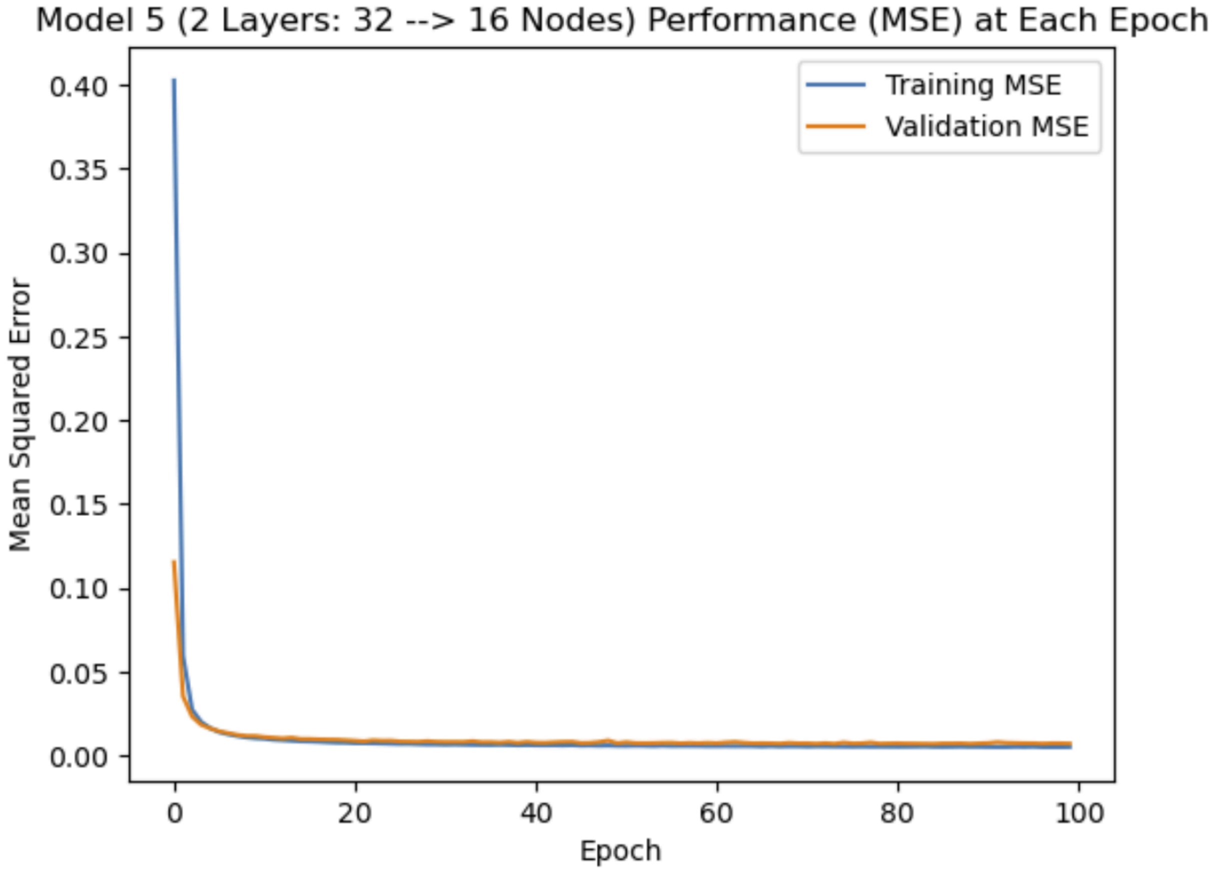
Learning Rate Optimizer: Adam Optimizer
Activation Function: Gaussian Error Linear Unit (GeLU)
Loss Function: Mean Squared Error (MSE)
Training Time: 100 Epochs
Validation Split: 20% for validation

During the exploration of model architecture, seven distinct models were created to assess the impact of varying neuron counts and layer configurations on test error. The first three models consisted of a single hidden layer, each with 16, 32, and 64 neurons, respectively. Subsequent models, 4 through 6, featured two hidden layers, experimenting with combinations of 16, 32, and 64 neurons. From these, Model 5, which comprised two layers with 32 and 16 neurons, emerged as the best-performing architecture based on test error.

To address the challenge of overfitting, Model 7 was developed, building on the architecture of Model 5. This model incorporated several strategies to mitigate overfitting:

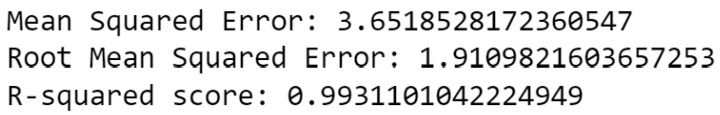
- An **L2 regularizer** was applied to the layers to penalize large weights, thereby discouraging complexity.
- **Dropout layers** were introduced between the hidden layers to randomly drop a proportion of neurons during training, enhancing the model's generalization ability.
- **Batch normalization** was added following the dropout layers to normalize the activations and improve training stability.
- **Early stopping** was implemented to halt training when the validation loss ceased to improve, thus preventing the model from fitting too closely to the training data.

This however didn’t result in a lower test loss than model 5. Therefore model 5 was chosen as the best NN model to represent the data. You can see model 5 training history below.



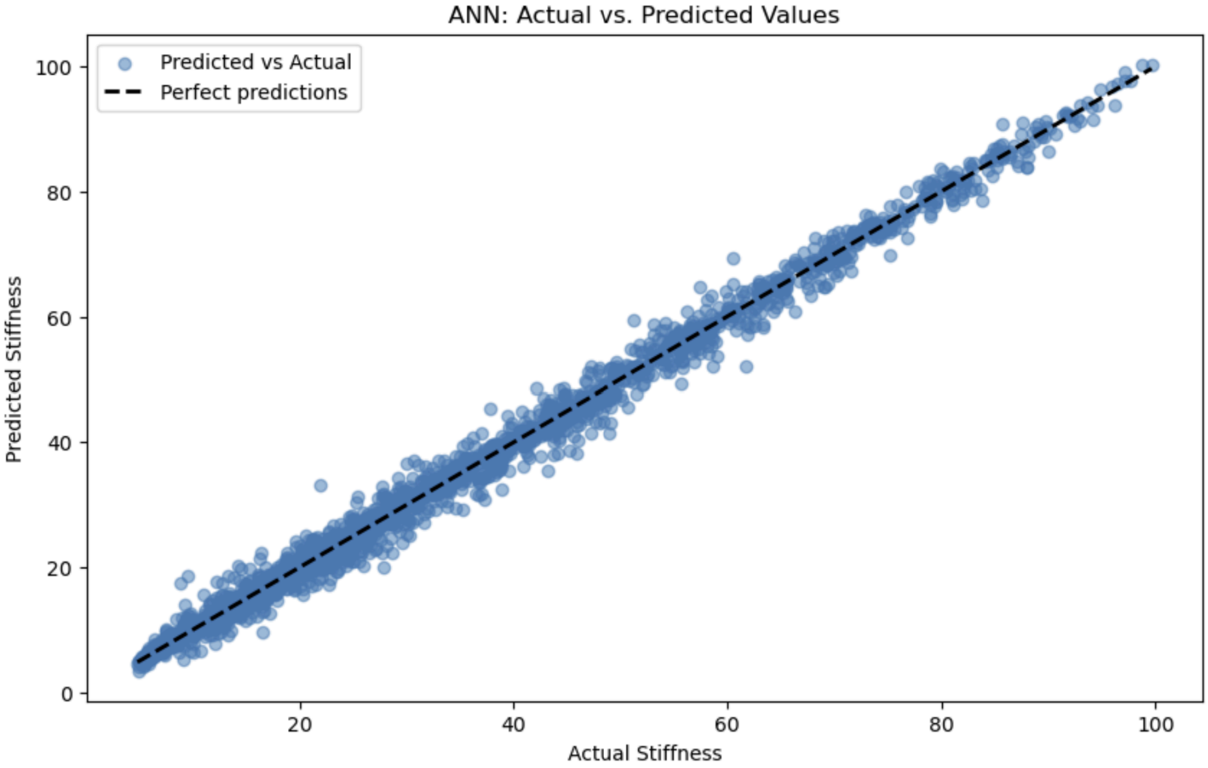
Results:-

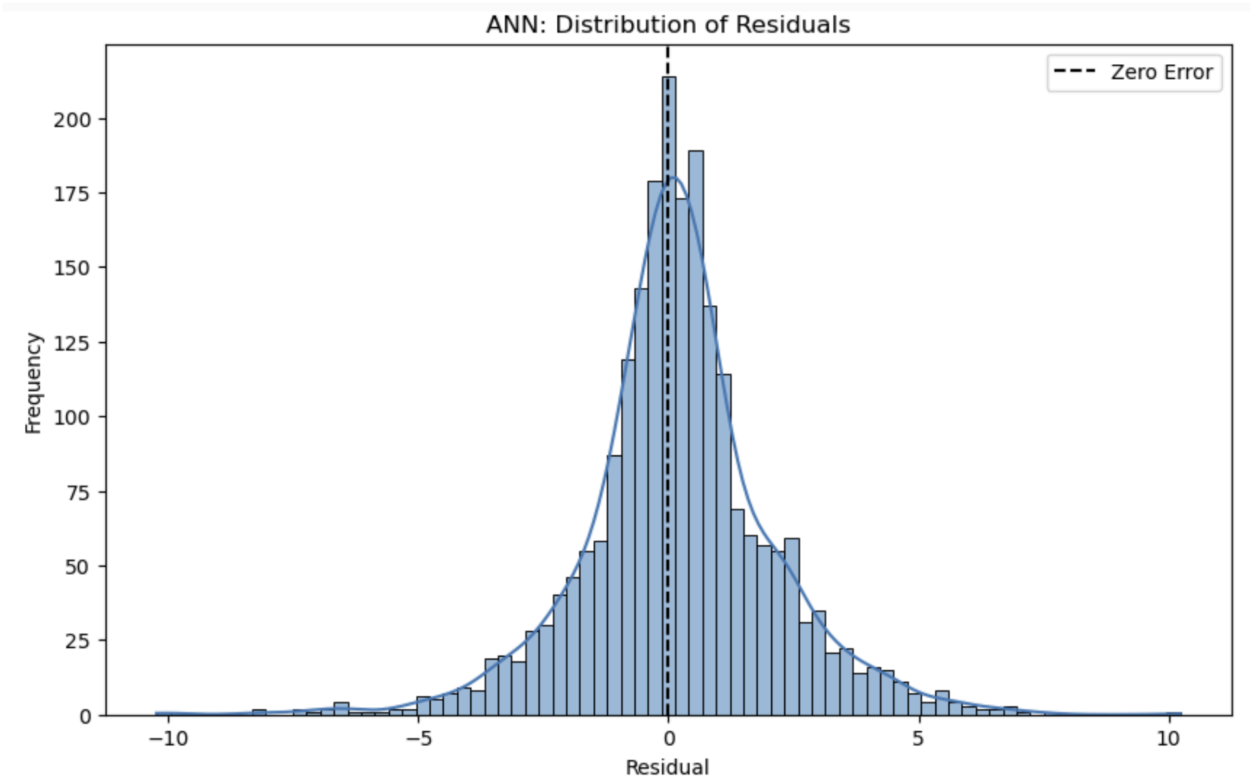
The model was evaluated using Mean Squared Error (MSE), Root Mean Square Error (RMSE), and R-squared metrics on a held-out test set. These model performances can be seen below.



```
Mean Squared Error: 3.6518528172360547
Root Mean Squared Error: 1.9109821603657253
R-squared score: 0.9931101042224949
```

The performance of the model was visualized through Actual vs. Predicted Values to see how predictions aligned to their true values. Residual plots were also created to assess the distribution of errors. These can be seen below.





4. Conclusion:

In this investigation, three machine learning models were developed and evaluated for their ability to predict material stiffness. The Stochastic Gradient Descent (SGD) Regressor served as the initial model, demonstrating a test loss (MSE) of 63.4. Subsequent improvement was achieved with the Gaussian Process Regression model, which reduced the test loss to 40.6. In the development of Artificial Neural Networks (ANNs), where a two-layer architecture (32 --> 16 neurons) emerged as the most effective, the test loss was further lowered to 3.65. The progressive enhancement in model performance, as evidenced by the decreasing test loss and supported by the visualizations, underscores the importance of model selection and architectural refinement in predictive modeling for each particular scenario, in this case: material science.