# Coding Challenge Analysis and Summary

## Website

We have provided an interface where you can try your own compositions, and it will predict the yield strength of your custom steel:

<https://cc-landing-plum.vercel.app/>

## Data Preprocessing

### Choices Made

* **Missing Values**: Composition values were retrieved from each steel's unique ID (the first column in the database) and converted from weight percent to atom percent. Missing values in elongation were handled by removing rows labeled "Unknown." This approach ensures the retention of valuable data by focusing only on complete and reliable records. Steel properties were later predicted using machine learning algorithms, including k-nearest neighbors.
* **Normalization**: Applied Z-score standardization to all numerical features (steel compositions). This ensures equal contribution from all features and is critical for distance-based models such as k-nearest neighbors. Z-score was preferred over Min-Max scaling because it retains the original data distribution and is less sensitive to outliers.
* **Categorical Encoding**: Used label encoding to classify steels into three categories: fragile, medium strength, and strong, based on elongation percentages. Label encoding was chosen over one-hot encoding to avoid introducing unnecessary dimensionality and to maintain a simple classification structure.

## Regression Analysis

### Algorithms Tested

* **Linear Regression** - Simple baseline model used for comparison purposes. Assumes a linear relationship between features and target variables.
* **Ridge Regression** - Incorporates L2 regularization to penalize large coefficients, improving stability in cases of multicollinearity among features
* **Lasso Regression** - Uses L1 regularization to shrink irrelevant feature coefficients to zero, effectively performing feature selection.
* **Random Forest Regression** - An ensemble learning method that aggregates predictions from multiple decision trees, reducing overfitting and improving robustness.
* **Decision Tree Regression** - Captures non-linear relationships in the data but is prone to overfitting without regularization.

### Rationale

### Baseline Testing - Linear regression provides a benchmark to compare the performance of more advanced algorithms.

### Regularization - Ridge and Lasso help address overfitting and multicollinearity issues while enhancing generalization performance.

### Ensemble Methods - Random Forest combines multiple models to improve stability and reduce variance in predictions.

### Non-Linear Models - Decision Tree Regression models non-linear data effectively, making it suitable for datasets with complex relationships.

### Hyper-Parameters Chosen

Random Forest's *n\_estimators=100* was chosen to ensure sufficient diversity in the ensemble without excessive computational cost. For Decision Trees, *max\_depth=10* was set to prevent overfitting. These values were chosen to improve performance and reduce overfitting. We also consistently used *RANDOM\_STATE=42* to ensure reproducible results across all algorithms tested.

### Evaluation Metrics

* **Mean Squared Error (MSE)** – A comprehensive measure that penalizes larger errors, helping to identify models with high prediction accuracy.
* **R-squared (R²)** - Indicates the proportion of variance in the dependent variable that is predictable from the independent variables. A higher R² implies better explanatory power.

### Results (Key Graphs)

**A graph of different colored bars

Description automatically generated with medium confidence**

**MSE Comparison**: *Comparison of Mean Squared Error (MSE) for different regression models.*

**A graph of different colored squares

Description automatically generated**

**R² Comparison**: *Comparison of R² scores for different regression models.*

### Regression Metrics Summary

| Algorithm | MSE | R² |
| --- | --- | --- |
| Linear Regression | 70000 | 0.45 |
| Ridge Regression | 68000 | 0.50 |
| Lasso Regression | 67000 | 0.52 |
| Random Forest | 12000 | 0.90 |
| Decision Tree | 20000 | 0.85 |

### Observations

**Best Performance**:

* **Random Forest Regression** stands out clearly with the highest R² scores and lowest MSE values for both tensile and yield strength. This strongly reaffirms its suitability for modelling these properties.
* **Decision Tree Regression** also performs reasonably well but exhibits slightly lower R² and higher MSE than Random Forest, indicating less consistency.

**Linear, Ridge, and Lasso Regression**:

* These models have noticeably poorer performance. The negative R² values for yield strength suggest that these models are unsuitable for this particular target property, as they fail to outperform even a simple mean-based prediction.

**Model Choice Impact**:

* The clear separation in the MSE graph emphasizes the value of ensemble methods like Random Forest for capturing complex relationships - especially when there are so many different elements to be considered - and reducing prediction errors.

## Classification Analysis

### Algorithms Tested

* **k-Nearest Neighbors (k-NN)** - Utilized k=5 for nearest neighbor calculation. This algorithm captures local patterns by classifying based on the majority label among the k-nearest neighbors. Effective for smaller datasets where local patterns in the data are essential for classification.
* **Random Forest Classifier** - An ensemble learning method combining predictions from multiple decision trees to enhance reliability and reduce overfitting. Provides a robust solution by aggregating predictions from multiple models, which reduces variance and handles class imbalances better.

### ****Logistic Regression**** - A probabilistic approach to binary and multi-class classification, offering high interpretability. Acts as a baseline with its interpretable nature, providing insights into class separability in the dataset.

### Evaluation Metrics

* **Accuracy**: Measures the proportion of correctly classified instances across all strength categories.
* **Classification Report**: Provides precision, recall, and F1-scores for each class, offering a detailed performance breakdown.
* **Confusion Matrix**: Visualizes predictions and highlights misclassifications across fragile, medium, and strong categories.

### Results (Key Graphs)

A graph of confusion matrix

Description automatically generated

**Confusion Matrix (k-NN)**: k-NN struggles with imbalanced data, misclassifying most incorrectly predicted samples as strong and failing to identify fragile steels.

.A graph with blue squares and white text

Description automatically generated

**Confusion Matrix (Random Forest)**: Random Forest achieves the highest accuracy but misclassifies all fragile samples as medium or strong.

A diagram of a logistic regression

Description automatically generated

**Confusion Matrix (Logistic Regression)**: Logistic Regression correctly classifies fragile steels, and has some slight inaccuracies in either direction with medium and strong steels.

### Observations

* **Best Performer** – Logistic Regression is the only model to correctly classify fragile samples, which demonstrates its ability in handling minority classes, despite misclassifying a few medium samples as strong and vice versa. Random forest successfully classified the same number of samples (56) but misclassified all fragile samples as medium or strong. It exhibited higher precision for strong samples but lacked the sensitivity to detect fragile instances. While Random Forest offers overall balance, Logistic Regression's ability to classify fragile steels makes it a better choice for applications where detecting all categories of strength - particularly fragile samples - is critical.
* **k-NN Limitations** - k-NN consistently underperformed, misclassifying the most samples (8 vs 6 in Random Forest and Logistic Regression, showing it struggles with imbalanced datasets and is overly sensitive to noise in the data.

## Optimized Compositions

### Graphs

**A graph with blue dots

Description automatically generated**

**Optimized Compositions: Ni vs Co**: A scatter plot of optimized compositions, highlighting the Nickel (Ni) vs. Cobalt (Co) content. This visualization emphasizes the successful optimization process to minimize the usage of both Ni and Co while meeting the high strength criteria.

**A diagram of red and green dots

Description automatically generatedNickel Content vs. Predicted Strengths**: A scatter plot displaying the relationship between Nickel content and predicted tensile and yield strengths. The graph demonstrates that lower Nickel content does not significantly compromise mechanical properties, supporting cost-effective material designs.

**A chart of red and green dots

Description automatically generatedCobalt Content vs. Predicted Strengths**: A scatter plot illustrating the effect of Cobalt content on predicted tensile and yield strengths. The results confirm that strong mechanical properties are achievable even with reduced Cobalt content, minimizing reliance on expensive elements.

### Observations

* **Key Features** - The optimization process identifies compositions that maintain high tensile and yield strengths while minimizing Nickel and Cobalt contents. This balance is essential for producing cost-effective and sustainable steel alternatives.
* **Patterns Observed**:
  + Nickel: While the optimized compositions reduced Nickel content to as low as 0.5% in some iterations, its strength contributions were consistent across various percentages. This suggests room for further reductions without significant losses in performance.
  + Cobalt: The optimized compositions consistently retained Cobalt around or slightly above 1%, with similar mechanical performance trends, suggesting it is a critical yet controllable element in these alloys.
  + *Mechanical Properties:*
    - The scatter plots reveal minimal impact of increasing Nickel or Cobalt percentages on yield strength, which remains within 1500–1650 MPa across compositions.However, the tensile strength does increase as Nickel and Cobalt content increases. With Cobalt, tensile strength shows a slight positive trend above 0.5%, with strengths consistently exceeding 1750 MPa. Below this percentage, tensile strength drops to around 1600 MPa.With Nickel, tensile strength increases significantly beyond 1.25%, rising from a maximum of 2050 MPa below this threshold to approximately 2300 MPa above it.
* **Top Candidates** - The top-performing compositions achieve tensile strengths and yield strengths above 1500 MPa with Nickel above 1.25% and Cobalt above 0.5%. These compositions reflect the ideal balance between performance and material costs.

## Conclusion

The optimization process demonstrates how strategic machine learning choices—spanning data preprocessing, regression, classification, and material property analysis—enable the identification of cost-effective and high-performance steel compositions.

* Data Preprocessing - Techniques such as Z-score normalization and label encoding ensured that models were fed standardized, relevant inputs, crucial for model performance.
* Regression - Random Forest Regression emerged as the best predictor for tensile and yield strength, achieving the lowest error rates and the highest explanatory power due to its ability to capture complex relationships.
* Classification - Logistic Regression was the most suitable for handling imbalanced data, accurately classifying fragile samples.
* Optimization Insights - The results emphasize that high mechanical performance is achievable even with reduced Nickel and Cobalt contents. Optimized compositions strike a balance between strength and material costs, highlighting sustainability without sacrificing performance.