Evaluation of Weld Quality Using Machine Learning

Guilherme Mertens de Andrade, Mateus José de Sousa Goto, and João Pedro Regazzi

CentraleSupélec, Artificial Intelligence Department

October 24, 2024

1 Introduction

In this project, we are evaluating the quality of welds across various types of welding methods and material compositions. To perform this evaluation, we carried out comprehensive data preprocessing and analysis to clean and structure the dataset (link to dataset). This step allowed us to identify key patterns and inconsistencies in the data. Based on this analysis, we selected the most appropriate methods for our evaluation, applying a combination of supervised and semi-supervised learning techniques, alongside a Generative Adversarial Network (GAN), to enhance prediction accuracy and model robustness.

2 Data Analysis

2.1 Data Collection

The data required for this project was gathered from the Materials Algorithms Project Data Library. The dataset is provided by the Phase Transformations and Complex Properties Research Group at the University of Cambridge. The data was initially in a .data format, which contained some errors that had to be corrected manually before being converted to CSV.

2.2 Target Selection

As suggested by Tracey Cool's research [2], we chose the labels *yield strength* and *ultimate tensile strength* as potential predictors for assessing weld quality, as these are two key metallurgical metrics commonly used to evaluate the strength and durability of materials. Given their high correlation, we opted to predict only one of them and excluded the other. Including one as a feature to predict the other would have led to artificially high performance, as the model could easily learn the relationship between the two variables.

2.3 Data Features

The dataset includes features related to the chemical composition of the weld, process parameters, and the resulting weld properties. It contains both categorical and numerical data, with varying scales, and includes an ID that can provide additional information.

2.4 Dataset Split

The dataset was initially split into training, validation, and testing sets to prevent data leakage. This ensures proper model training, hyperparameter tuning, and unbiased evaluation while maintaining consistent statistical distribution of chemical compositions and mechanical properties across all sets

Set	Size of X
Xtrain	494
Xvalid	248
Xtest	248

Table 1: Sizes of Training, Validation, and Testing Sets

2.5 Explanatory Analysis

We conducted an exploratory data analysis to assess missing data patterns, their percentages, and the distribution of categorical variables. Key findings include:

- Carbon, Silicon, Manganese, Sulfur, and Phosphorus have almost no missing values.
- Nickel, Chromium, Molybdenum, Vanadium, and Copper have significant missing values at the beginning of the dataset and exhibit strong correlations.
- Cobalt and Tungsten have a high percentage of missing data and are also correlated.
- Oxygen, Titanium, Nitrogen, Aluminium, Boron, and Niobium are highly correlated with more dispersed missing values.
- Tin, Arsenic, and Antimony show similar patterns of correlation and missing data.
- Current and Voltage are correlated, as expected.
- Yield Strength, Ultimate Tensile Strength, Elongation, and Reduction of Area also show strong correlations.
- The final set of features contains a high amount of missing values, which might justify dropping them.

Figure 10 includes a plot visualizing the distribution and correlation of missing values, showing that most variables exhibit a Missing At Random (MAR) pattern due to these correlations.

2.6 Categorical Variables

We transformed the ID column into categorical variables. Overall, the dataset is highly imbalanced, with several features having a higher proportion of missing values than others with low cardinality. For features with high cardinality, one dominant category tends to represent a large portion of the data.

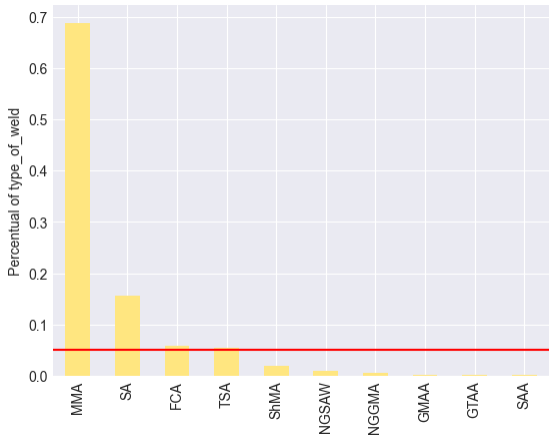

Figure 1: Type of Weld Distribution

2.7 Numerical Variables

We performed a detailed analysis of the numerical features by visualizing their distributions, comparing them to normal distributions, and using boxplots to illustrate data concentration, including medians and quartiles. This allowed us to identify outliers and skewed distributions, which are particularly important for understanding potential anomalies and trends in the dataset.

Most of the data displays skewed distributions, especially the chemical elements from sulfur to tungsten, which exhibit unusual patterns. We need to be cautious when handling outliers, as some extreme values, such as those observed in sulfur, may not be true outliers but rather part of a shared pattern with other features, such as phosphorus, which shows similar behavior.

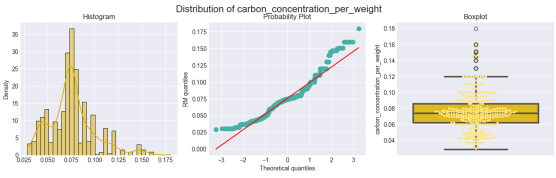

Figure 2: Boxplot of the Distribution of Carbon Concentration per Weight

3 Data Preprocessing

3.1 Categorical Features

3.1.1 Missing Labels

We started by separating the categorical and numerical variables, then proceeded to do the data imputation on these variables. We also added the new "Missing" category, so that the model can capture the reason for the missing instance in case it exists. Afterwards, we checked the new distribution of some categorical variables that had missing data.

3.1.2 Categorical Encoding

Categorical variables require proper preprocessing to ensure they are suitable for the model. In this case, label encoding is not ideal because the categorical variables lack ordinal relationships. For example, categories like standard colors don't imply a hierarchy, which is also true for our variables.

Therefore, we applied one-hot encoding to the categorical labels `weld_id`, `current`, `ac_or_dc`, `voltage`, and `type_of_weld`. This method converts each category into a binary vector, allowing the models to interpret categorical features without imposing any ordinal structure. However, this transformation significantly increased the number of columns in the dataset. With only about 1600 rows, this increase in dimensionality could introduce the risk of overfitting.

To address this, we used UMAP (Uniform Manifold Approximation and Projection), a dimensionality reduction technique that preserves the local structure of the data while reducing its dimensionality.

PCA has limitations with one-hot-encoded data, as it treats binary variables as continuous, concentrating variance in only a few components, which can reduce interpretability. Additionally, PCA struggles with sparse data, making techniques like UMAP or t-SNE better suited for handling it. We also considered Decision Tree Encoder but didn't use it due to the limited number of non-null target values.

3.2 Data Imputation

Following this initial analysis, we determined that the missing values were predominantly Missing At Random (MAR) or Missing Not At Random (MNAR). Based on the reference work by Tracey Cool in *Design of Steel Weld Deposits* [1], we adopted different imputation strategies depending on the type of feature. For instance, missing values related to Phosphorus (P) and Sulphur (S) were handled by mean imputation, while missing values for Manganese (Mn) and Nickel (Ni) were set to zero. These choices were informed by the nature of the data and domain-specific insights, as such concentrations often behave differently in relation to the weld quality.

Following the initial analysis, we determined that the missing values were primarily Missing At Random (MAR) or Missing Not At Random (MNAR). Based on insights from Tracey Cool's *Design of Steel Weld Deposits* [1], we applied different imputation strategies depending on the feature. For example, missing values for Phosphorus (P) and Sulfur (S) were imputed with the mean, while Manganese (Mn) and Nickel (Ni) were set to zero. These decisions were guided by domain-specific knowledge—Phosphorus is always present in welds, so missing values likely indicate measurement failures, whereas Manganese is rarely detected, meaning Nans likely reflect undetected concentrations.

For features like Elongation and Reduction of Area, which result from the overall welding process, KNN imputation was applied. We also experimented with simpler methods, such as imputing all values as zero, mean, mode, or using KNN for all features to assess their effectiveness.

Handling missing values was a critical part of data preprocessing, necessitating the evaluation of multiple strategies.

3.3 Transformation of variables

Our next step was to address the asymmetry in the variable distributions to enhance model training. We applied the Yeo-Johnson transformation to make the variables more symmetrically distributed.

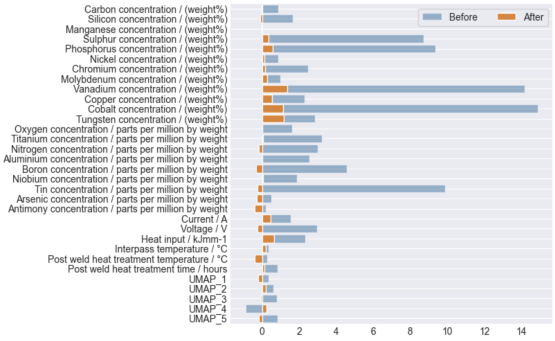

Figure 3: Skewness before and after Yeo-Johnson Transformation.

Symmetrically distributed variables are often preferred in statistical and machine learning models because they help prevent the model from being overly influenced by extreme values or skewed data. Correcting for asymmetry allows the model to better capture underlying data patterns, improving the accuracy and reliability of predictions.

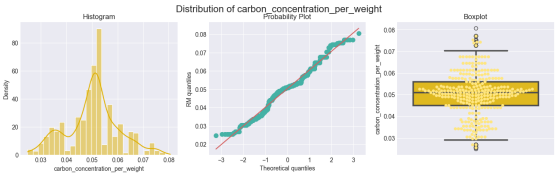

Figure 4: Boxplot of the Distribution of Carbon Concentration per Weight after Transformation

After applying the transformation, we visualized the distributions. For instance, the original asymmetry in the distribution of Carbon (as shown in Figure 2) was reduced, though the distribution is still not perfectly normal (see Figure 4)). However, it now tends to be more symmetrical.

We also noticed that after the transformations that the variables still have outliers. However, treating these outliers may represent a loss of information. We will therefore choose not to treat these outliers, as removing them could mean a huge loss of information.

3.4 Feature Scaling

As we preferred to keep the outliers, we then used the RobustScaler, which is a feature scaler robust to outliers. This Scaler removes the median and scales the data according to the quantile range.

To verify that the feature scaling process worked, we plotted once again the boxplot of the *Distribution of Carbon Concentration per Weight* variable and verified that most of the samples were between -1 and 1, with the outliers having a module greater than 2.

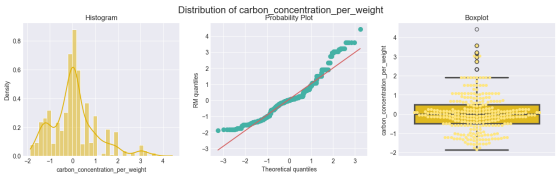

Figure 5: Boxplot of the Distribution of Carbon Concentration per Weight after Robust Scaling

4 Supervised Learning Models

After completing all preprocessing steps, we conducted a comprehensive evaluation of various machine learning models and different imputation strategies to identify the best performing models. Our target variable was *Yield Strength*, and we dropped the other target variable due to its high correlation with Yield Strength. As this was a supervised learning task, we also removed all rows where the *Yield Strength* was missing, which reduced the dataset size by nearly half. We used the python library *PyCaret* to easily train different models.

Table 2 summarizes the performance of the supervised regression models evaluated on our custom imputed dataset:

Table 2: Performance of Supervised Regression Models

Model	MSE	RMSE	R^2
et	1803.4	41.7	0.7855
lightgbm	2086.1	44.9	0.7504
gbr	2102.4	45.3	0.7455
xgboost	2212.7	46.5	0.7315
rf	2483.9	49.3	0.7012
ada	2904.8	53.6	0.6598
knn	3006.9	54.1	0.6373
lasso	3593.8	59.6	0.5792
llar	3593.7	59.6	0.5792
br	3588.8	59.6	0.5762
ridge	3623.2	59.9	0.5701
lr	3627.9	59.9	0.5694
lar	3627.9	59.9	0.5694
huber	3830.9	61.5	0.5491
en	4054.6	63.2	0.5366
par	4205.9	64.2	0.5118
omp	5207.6	70.8	0.4228
dt	4853.7	69.0	0.4133
dummy	9090.2	94.5	-0.0284

In addition, we experimented with model blending to assess whether combining predictions could improve performance. The blended models produced results similar to those of individual models. We tested these models using different imputed datasets from the earlier steps, and our custom imputation yielded the best accuracy, demonstrating a solid understanding of the dataset's characteristics.

Table 3: Imputation Methods and Corresponding R^2 Values

Imputation Method	R^2 Value
Mean	0.81
Constant (0)	0.82
KNN	0.81
Iterative Imputer	0.79
Custom	0.82

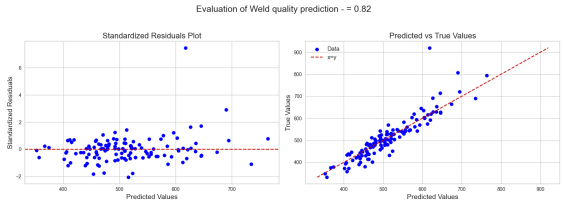

Figure 6: Prediction of Weld Quality using Supervised Learning and our Custom Imputed Dataset.

5 Semi-supervised Learning Models

In this section, we explore semi-supervised learning, an approach that leverages unlabeled targets to enhance the training of our model. Initially, we conducted an analysis to assess the data distribution between the labeled and unlabeled instances, revealing a significant discrepancy. This difference indicates that valuable information is being overlooked when the unlabeled data is excluded, suggesting potential performance gains by incorporating these data points into the training process.

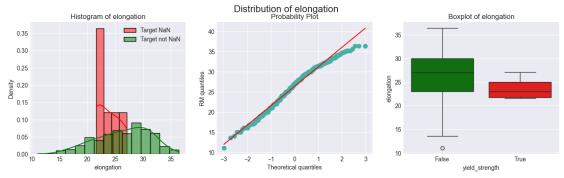

Figure 7: Difference of Distribution between instances with not NaNs and Nans target.

5.1 GAN Model

Although GANs are primarily known for applications in computer vision, they can be adapted for semi-supervised learning in regression tasks with tabular data. In this approach, the discriminator is modified to both distinguish between real and synthetic samples and to predict continuous values for the labeled data.

During training, the discriminator learns from both labeled and unlabeled data. The generator produces synthetic data that the discriminator tries to differentiate from real data, forcing it to learn richer representations. Even when some samples lack labels, the task of distinguishing real from fake data encourages the model to extract useful patterns from the underlying data distribution. This helps the model to better predict continuous values for labeled data, as the discriminator refines its understanding of the real data characteristics.

The discriminator has two objectives: accurately predict the regression values for labeled data, and distinguish between real and synthetic samples. This dual task helps the network to leverage unlabeled data effectively, as the discriminator indirectly learns information about the data distribution from these samples, enhancing its overall predictive capabilities.

Since it's a deep learning approach, and we had a small amount of data, we could only reach a R^2 of 0.82.

5.2 Self-Learning Approach

We also experimented with a self-learning approach, which leverages the data to improve model performance without requiring explicit supervision. In self-learning, a model is initially trained on a labeled dataset. Subsequently, it generates pseudo-labels for the unlabeled data based on its predictions. These pseudo-labeled examples are then incorporated back into the training set, allowing the model to refine its understanding and enhance its performance iteratively.

For this approach, we selected the Extra Trees Regressor, as it had demonstrated the best performance in our initial analysis. Using this method, we achieved our highest R^2 score of 0.84.

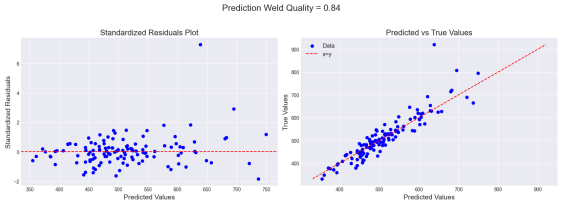

Figure 8: Prediction of Weld Quality using Semi-supervised Learning and our Custom Imputed Dataset.

5.3 VIME

After extensive research into current approaches, we identified a method similar to GANs for semi-supervised learning that utilizes null values to improve the model's understanding of the dataset. This method is called VIME [3].

VIME consists of two components: self-supervised learning and semi-supervised learning. In the self-supervised phase, VIME defines two pretext tasks: feature vector estimation and mask vector estimation. It involves corrupting samples by masking some features and replacing them with noise, then training an encoder to reconstruct the original sample and identify the mask. This helps the encoder learn feature correlations and generate informative representations.

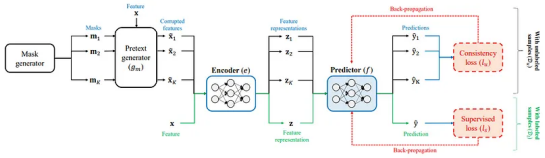

Figure 9: Semi-supervised Model Structure of VIME.

In the semi-supervised phase, VIME leverages the trained encoder to create multiple augmented samples for each data point by masking and imputing different features. A predictive model is then trained using a supervised loss on labeled data and a consistency loss on the augmented samples to encourage robustness.

Despite our efforts, we struggled to implement VIME effectively for our regression task. The method was originally designed for classification, so adapting it required modifications to the loss functions, which made it difficult to achieve convergence. However, we included it here due to its promising potential for semi-supervised learning.

6 Conclusion

In conclusion, the most significant aspect of this work was our methodology. We adhered to best practices to avoid data leakage, using robust techniques for data treatment such as Yeo-Johnson transformations, and employed

cross-validation tools to thoroughly test our models. Additionally, we applied and studied semi-supervised learning methods, enhancing the overall performance and generalization of our models. This rigorous approach ensured the reliability and validity of our results.

7 Figures

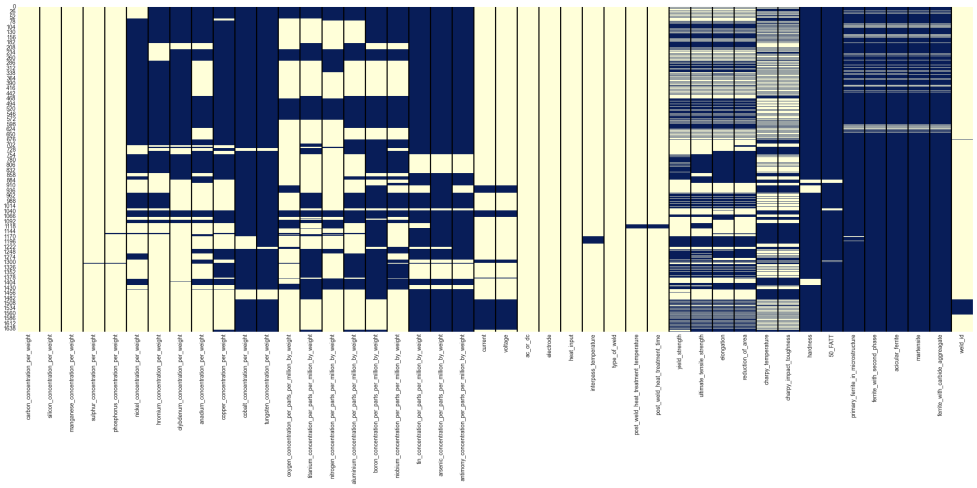

Figure 10: Investigation of NaN values in the dataset.

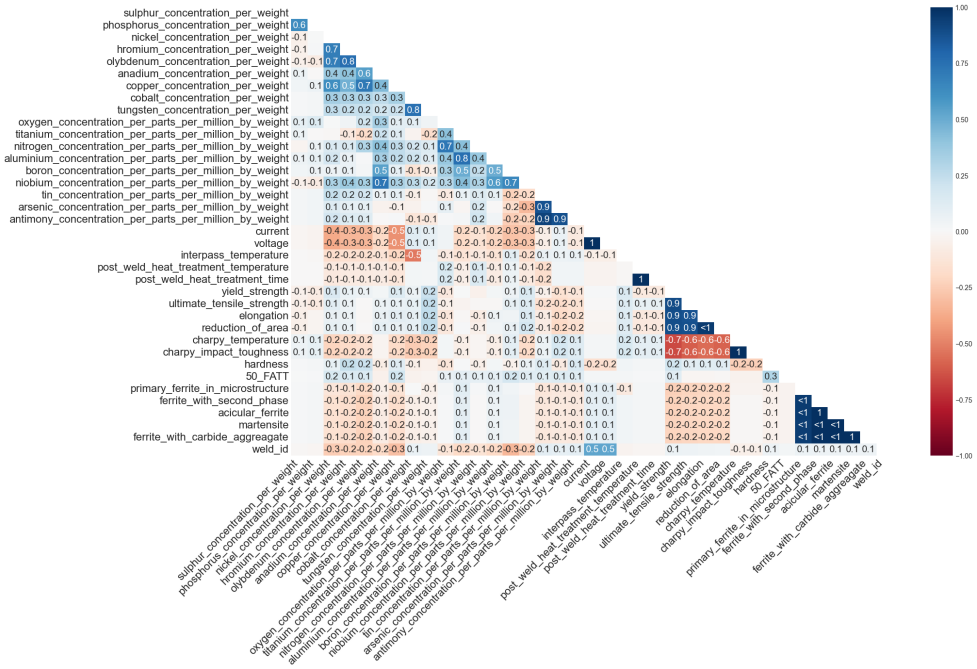

Figure 11: heatmap of the dataset.

References

[1] Tracey Cool. *Design of Steel Weld Deposits*. PhD thesis, University of Cambridge, 1996.

[2] Tracey Cool, H. K. D. H. Bhadeshia, and David J. C. MacKay. Materials science and engineering. *Materials Science and Engineering*, 1997.

[3] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self- and semi-supervised learning to tabular domain. *arXiv preprint arXiv:2106.07438*, 2021.