# **Predicting Steel Strength: A Regression-based Machine Learning Approach**

Steel is one of the most ubiquitous materials in modern society. Its mass production was one of the primary drivers of the First Industrial Revolution. It’s relative affordability and high strength has made it feasible to build big and strong without breaking the bank.

It is primarily Iron with a mix of other elements known as alloying elements. Combining different alloying elements can result in widely varied properties and as such, depending on the application, an appropriate alloying composition can be chosen. Metallurgists could find use in having a rough idea of the strength of a grade of steel prior to it being manufactured. In this project, A regression model was created that estimates the strength of a steel sample based solely on its alloying elements and temperature.

Steel is a polycrystalline material, meaning it’s made of multiple crystals. Crystals are groups of atoms which have a repeating fundamental structure, known as a unit cell.

Polycrystalline materials are a group of bonded crystals that all point in different directions as shown below:

Chart

Description automatically generated

In a metallurgical setting, crystals are commonly referred to as grains. Each of the enclosed areas in the “polycrystalline” figure are a grain.

A black and white drawing of a polycrystalline

Description automatically generated with low confidence

Adding elements to Iron can change the size and shape of these grains while also resulting in the creation of new phases. The addition of alloying elements can also stretch or compress the crystal lattice of the steel which can provide some benefit. All of these changes can result in improved strength.

# Data

Steel chemistry data was collected from the machine learning data repository,

Graphical user interface, text, application, table

Description automatically generated[Kaggle](https://www.kaggle.com/datasets/rohannemade/mechanical-properties-of-low-alloy-steels?resource=download). It consists of 915 samples of steel each with its respective steel chemistry and strength parameters. A sample of the dataset is shown below:

# 2.0 Data Cleaning

Various features needed to be dropped, Alloy code wasn't useful in this context, neither was Carbon equivalent (Ceq). Columns were then renamed. 0.2% Proof Stress is another name for Yield strength and was renamed as such.

There were no null values however there was one unusually high strength property observation which was dropped. Additionally, the temperatures the samples were pulled ranged from 27ºC to 650ªC. A cut-off of 450ºC was chosen since most steel applications don't reach temperatures that high. 450ºC is still unusually high for a typical engineering application, however there would too much data would be removed if temperatures were set to under 450˚C.

# 3.0 EDA

Table

Description automatically generatedThe first step of EDA was to look for general patterns in the data. A heatmap was created with the correlations of each feature to the four potential target variables:

Temperature is negatively correlated with both Yield and Tensile strength which is expected. The higher the temperature, the weaker a metal gets. Correlations between the other strength variables are all expected as well but the goal is to find relationships between the elements and strength.

The elements that are most influential are Vanadium (V), Molybdenum (Mo), Nickel (Ni) and Manganese (Mn). Surprisingly, Carbon doesn't have a huge role to play in determining strength. There are no elements that contribute negatively to steel strength in a significant way.

The following scatterplots show the relationship between the Yield strength and the weight percent of each element in that sample of steel.

The strength variable, Yield strength, was chosen to be the target variable in this project since it is one of the most important strength parameters and is widely used. It is the value of the applied stress (tension) to the material that would result in permanent deformation. One would want to avoid a low-tensile strength steel in an application that requires strength.

Chart, scatter chart

Description automatically generated

# 4.0 Preprocessing

The remaining data was split into training, test and validation sets with a 0.7:0.2:0.1 split respectively. A validation set was created to assess the model’s performance more robustly. All features, X data, were fit and transformed on X\_train using a Standard Scaler and were transformed on X\_val and X\_test. The y data, target variable, were kept as is.

# 5.0 Modelling

Table

Description automatically generatedPyCaret is a low-code machine learning library that automates the model selection process. It can score various models using k-fold cross-validation and returns a ranked list of the best models. This is very useful in preliminary modelling. Using this feature, the top 3 models were chosen. The top 10 models are shown below:

The Dummy Regressor as seen at the bottom was used as reference and had an MAE and RMSE of 116 and 49 which would make it unable to make accurate predictions, even in this case when rough estimates are required. The CatBoost Regressor, Light Gradient Boosting Machine and Extra Trees Regressor were chosen to be input into a Voting Regressor to be explained later on in this report. The reason why XGBoost was not chosen to be in the ensemble will also be explained further down.

## Explaining Models

The CatBoost Regressor (CAT) is a relatively new machine learning model. This model is an evolution of decision trees and gradient boosting and is best at working with categorical data. In this instance it works well with numeric values as well!

LightGBM (LGBM) and XGBoost are similar models. Where they differ is how their trees grow. In LGBM trees are grown vertically or leaf-wise. XGBoost leaves are grown level-wise. This distinction results in LGBM being faster, but it does tend to overfit.

A picture containing diagram

Description automatically generated

Diagram

Description automatically generated

Extra Trees (XT) models are also an ensemble decision tree model like Random Forests. The differentiating factor is that decision trees in an XT model are trained on the entire dataset unlike the decision trees in Random Forests that are trained on bootstrapped samples. Nodes are also split randomly unlike in Random Forests where they are split optimally according to a selection criterion. Since there is no heavy calculation required when splitting, XT are much faster.

## 5.2 Feature Importance

Here are the graphs displaying the importance of each feature to the model’s predictions.

**Chart, scatter chart

Description automatically generatedCatBoost Regressor (CAT):**

**Extra Trees (XT):**

Chart, scatter chart

Description automatically generated

**Light Gradient Boosting Machine (LGBM):**

Chart, scatter chart

Description automatically generated

**XGBoost (XGB):**

A graph with a number of lines

Description automatically generated

Feature importance by themselves don’t explain whether a feature negatively or positively the predictions from the model themselves. It explains magnitude of importance but not direction of its influence. However, this relationship can be deduced using the correlation plot.\

There are a few common elements that are major contributors to these 3 models, Vanadium (v), Manganese (mn) and Nickel (ni). Temperature is also a major contributor. An interesting finding is that XT and CAT rely heavily on the top 3 or 4 elements to make their decision. On the other hand, LGBM takes a more democratic approach and weighs the information of the other elements as well. Vanadium weight is also quite high in XT and CAT but has second to last importance in LGBM. These models are taking quite a different approach. Temperature has the highest feature importance in the LGBM model and therefore its contribution should be limited which will be explained below.

As it can be seen, Vanadium is the element which contributes most to Yield strength in XT and CAT. Most samples didn't contain this element (367) as it can be seen in the histogram on the next page.

Even though XGBoost had a more favourable MAE compared to XT, the issue was that the model was over reliant on Vanadium. This could result in reduced generalizability in the model. Most samples in this dataset do not contain any Vanadium. Therefore, predictions on a set of samples that have a different distribution of Vanadium will not be accurate.

The LGBM model also places a large importance on a different feature, temperature. Although the goal is to minimize this feature’s influence on the model, it does also place a fair importance on important elements such a Ni, Mn and Mo.

A picture containing text, screenshot, line, plot

Description automatically generated

Vanadium (V), Nickel (Ni) and Manganese (Mn) all contribute to increased strength in these samples. They reduce the grain size of the steel and/or form new phases in the steel matrix that reduce movement of dislocations [1][2][3].

Temperature plays a crucial role in reducing Yield strength. It plays quite a big factor in the LGBM model especially. The impact is high but is negative as it can be seen from the correlation plot. This is because an increase in temperature makes the movement of dislocations in most metals much easier since atoms are physically moving more. This ease of dislocation movement in higher temperatures causes metals to show less resistance to stress.

## 5.3 Hyperparameter Tuning

A table of numbers and symbols

Description automatically generated

The metrics of the default and tuned models can be seen above. Default models were first trained on the training set, and tested on the training and validation sets to set a benchmark. A new instance of the model was then hyperparameter tuned via cross-validation. The resulting tuned model was trained on the training set and again tested on the test and validation sets. MAE, MSE, RMSE and R2 were calculated and summarized in the table above. One column represents the performance of the default models and the other represents the tuned model for each of the three models.

Depth, learning\_rate and iterations were all used as hyperparmeters during CAT tuning since altering any of these can mitigate overfitting. A randomized search was performed on all models. As it can be seen, the tuned CAT performed very similarly to the default model. The default model was chosen to be included in the ensemble since its performance, specifically its RMSE and MAE, was marginally better on cross-validation.

Learning\_rate, max\_depth, n\_estimators and num\_leaves were used as hyperparameters in LGBM tuning. Once again, the model did overfit the training data but changing hyperparameters did make a slight improvement in performance. Therefore, the tuned model was chosen to be included in the ensemble.

n\_estimators, min\_samples\_split, min\_samples\_leaf and max\_depth were the hyperparameters chosen when tuning ExtraTrees. The same issue, overfitting, arose as with the other models but the tuned model performed slightly better and so it was included in the ensemble model. Even though this regressor did overfit completely on the training data, its model performance on the evaluation sets was still excellent. It also placed importance on the elements V, Mo, Mn and Ni which were crucial to the model predictability.

To conclude this section, all the models did overfit, but they also performed extremely well on the training and validation set as well as during cross-validation. Further refinement wouldn’t be necessary in this case since absolute accuracy wouldn’t bring much benefit in this use case.

# 6.0 Final Model

As mentioned above, a Voting Regressor was chosen to combine all models into what is known as an ensemble model. The advantage of using an ensemble model is its diversity. Incorrect predictions from an individual estimator are normalized by predictions from the others therefore increasing accuracy. Ensemble models are also more robust since each estimator might excel at predicting certain patterns in the dataset. When combined, they lead to improved performance versus each individual model. In this meta-model, a weighted average of each model's predictions is used to form a final prediction. The algorithm to determine these weights is shown below:

The most accurate weights for the CatBoost Regressor, Light Gradient Boosting Machine and Extra Trees Regressor had optimum weights of 0.3, 0.6 and 0.1 respectively. The final metrics are shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Weight | 0.3 | 0.6 | 0.1 | = 1.0 |

A table with numbers and lines

Description automatically generated

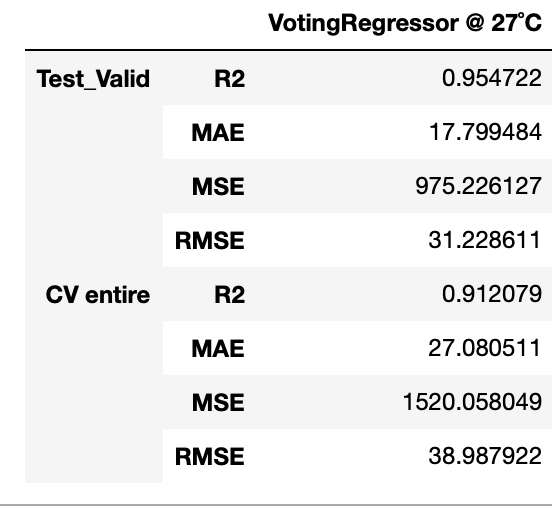
LGBM’s predictions contributes 60% to the ensemble model and is beneficial since it gives a much more balanced importance on the most influential elements. It does rely on temperature quite a bit but as it can be seen, it still performs exceptionally. Extra Trees doesn’t seem to contribute much and might be because it overfits. Default CatBoost performs better than the Voting Regressor. However, the Voting Regressor will be chosen as the final model due to its versatility as stated above.

In this business use case, the model’s performance would be best judged using both MAE and RMSE. This is because the data itself This is because metallurgists would only require a rough estimate of steel performance. The Voting Regressor scored an MAE of ~14.2 MPa and an RMSE of ~30 MPa. This means that the Voting Regressor’s predictions are on average, ~14.2 MPa away from the true strengths. The R2 was 0.96. The ensemble model describes 96% of the variance in the dataset. Considering the mean Yield strength from this data is 361 MPa, this model would excel at providing rough strength estimates.

Another evaluation was done on a subset of the data at a temperature of 27˚C, around room temperature.

To do this, all observations recorded at 27˚C were indexed. Using this index, new X and y datasets were created. These new sets were also removed of any training data.

To reiterate, the resulting dataset was comprised exclusively of test and validation data recorded at 27˚C. It consisted of 25 observations. The model was scored on this data and was cross validated on all the data (including the training data) that was recorded at 27˚C. The results are shown below:



When scored on the new test and validation data, the model still performed quite well. It’s MAE was ~27 MPa which is similar to the MAE obtained from training on the data from all temperatures even though the R2 did decrease to 0.91. However, the CV MAE when subsetting for 27˚C was higher than the CV MAE when the data from all temperatures was included (~27 MPa vs. ~14 MPa).

# 7.0 Conclusion

The ensemble model is excellent in predicting steel strength. Surprisingly, data on the samples' microstructure resulting from its heat treatment were not needed in this analysis. A limitation to this model is that the data is probably representative of a certain set of steel samples and may not be generalizable to other steel with different chemistries and heat treatments. However, this was taken into consideration which is why the ensemble Voting Regressor model was chosen even though its performance was slightly lower than that of the default CatBoost model. Additionally, the inclusion of temperature in this analysis might not be useful in most cases however it did perform very well on data observed at 27˚C.The regressor that weighed temperature most heavily, LGBM, has the lowest weightage in the final model which is also beneficial to the model’s performance using other data.

# 8.0 Sources

*[1] Applications of vanadium in the steel industry. (2021). Vanadium, 267–332. https://doi.org/10.1016/b978-0-12-818898-9.00011-5*

*[2] Applications of vanadium in the steel industry. (2021). Vanadium, 267–332. https://doi.org/10.1016/b978-0-12-818898-9.00011-5*

*[3] Kaar, S., Krizan, D., Schneider, R., Béal, C., Sommitsch, C. (2019). Effect of manganese on the structure-properties relationship of cold rolled AHSS treated by a quenching and partitioning process. Metals, 9(10), 1122. https://doi.org/10.3390/met9101122*