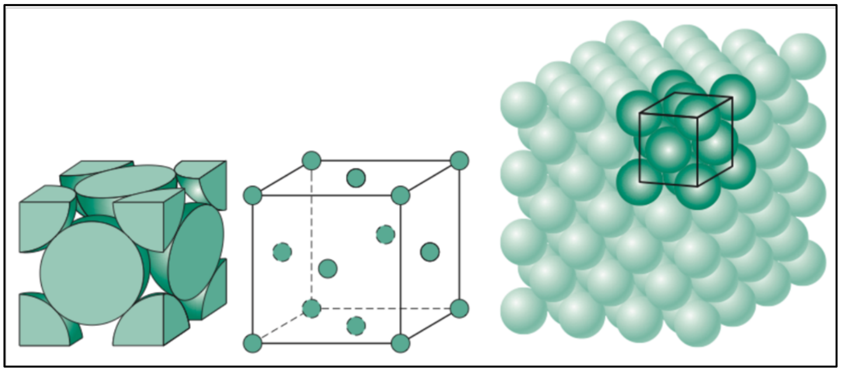
Predicting Steel Strength: A Regression-based Machine Learning Approach

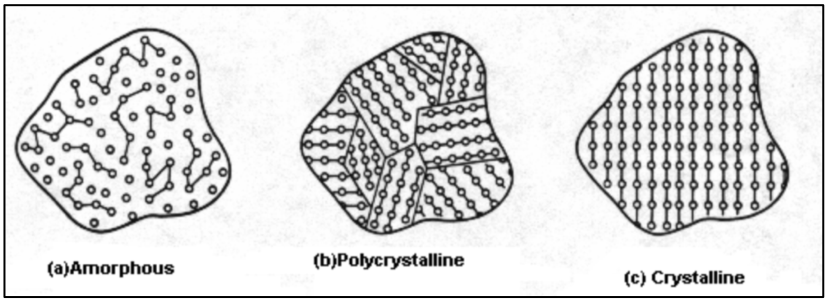
Steel is one of the most ubiquitous materials in modern society. Its mass production was one of the primary drivers of the First Industrial Revolution. Its relative affordability and high strength have made it feasible to build big and strong without breaking the bank.

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:



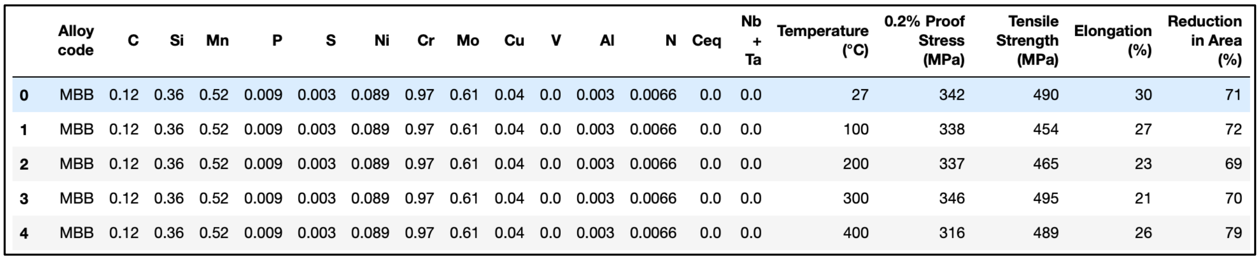
In a metallurgical setting, crystals are commonly referred to as grains. Each of the enclosed areas in the "polycrystalline" figure are a grain.



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, Kaggle. It consists of 915 samples of steel each with its respective steel chemistry and strength parameters. A sample of the dataset is shown below:



	Alloy code	C	Si	Mn	P	S	Ni	Cr	Mo	Cu	V	Al	N	Ceq	Nb + Ta	Temperature (°C)	0.2% Proof Stress (MPa)	Tensile Strength (MPa)	Elongation (%)	Reduction in Area (%)
0	MBB	0.12	0.36	0.52	0.009	0.003	0.089	0.97	0.61	0.04	0.0	0.003	0.0066	0.0	0.0	27	342	490	30	71
1	MBB	0.12	0.36	0.52	0.009	0.003	0.089	0.97	0.61	0.04	0.0	0.003	0.0066	0.0	0.0	100	338	454	27	72
2	MBB	0.12	0.36	0.52	0.009	0.003	0.089	0.97	0.61	0.04	0.0	0.003	0.0066	0.0	0.0	200	337	465	23	69
3	MBB	0.12	0.36	0.52	0.009	0.003	0.089	0.97	0.61	0.04	0.0	0.003	0.0066	0.0	0.0	300	346	495	21	70
4	MBB	0.12	0.36	0.52	0.009	0.003	0.089	0.97	0.61	0.04	0.0	0.003	0.0066	0.0	0.0	400	316	489	26	79

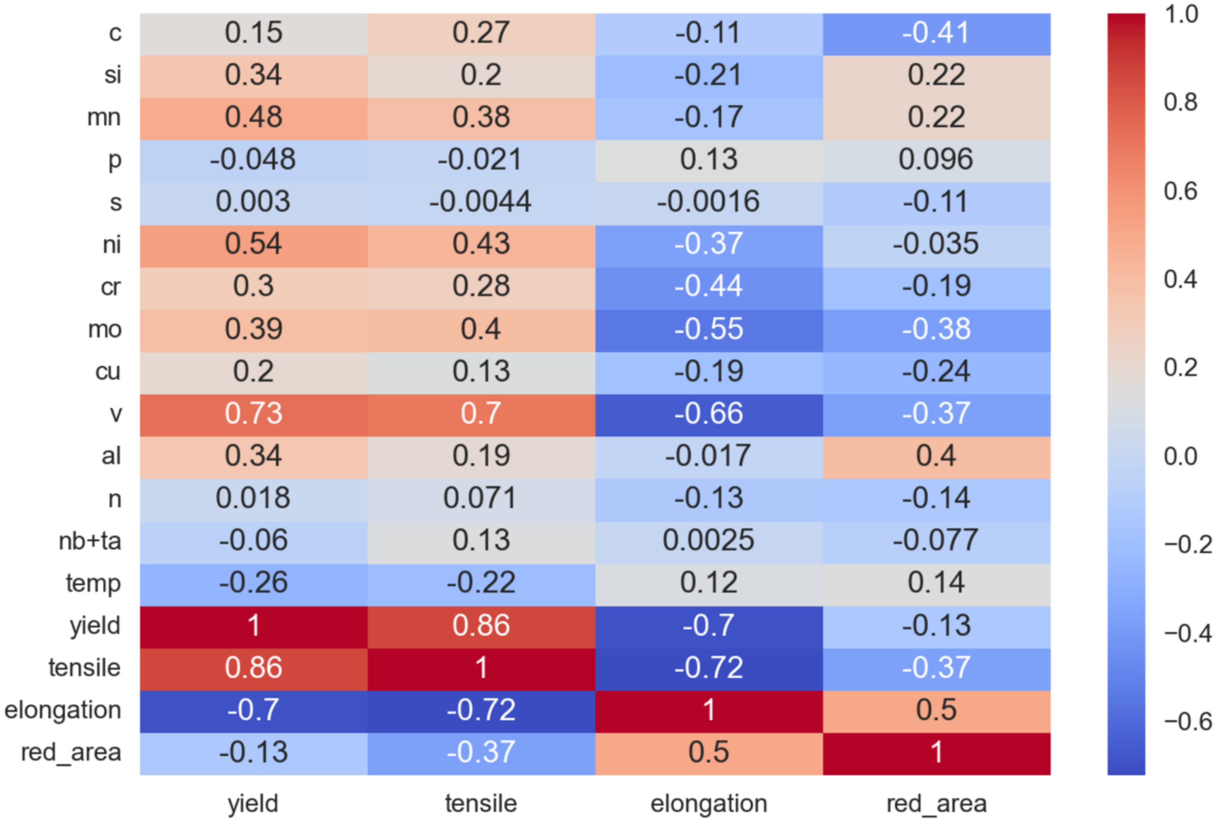
Data Cleaning & Wrangling

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 650˚C. 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 too much data would have to be removed if temperatures were set to under 450˚C.

EDA

The 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:

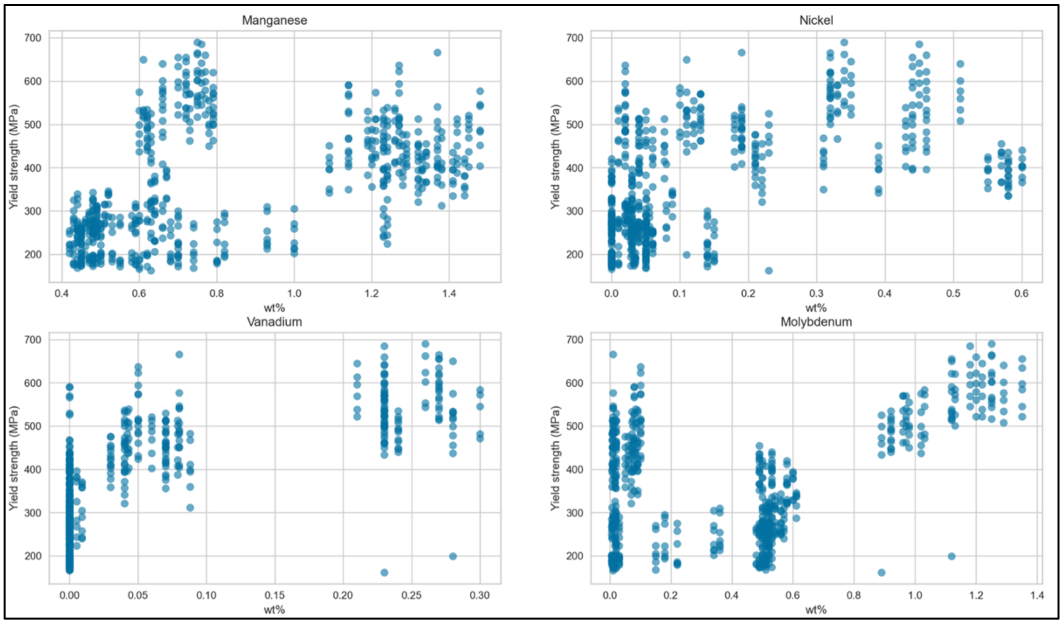


Temperature is negatively correlated with both Yield and Tensile strength which is expected. The higher the temperature, the weaker a metal gets. Correlations between the other strength variables are all expected as well but the goal is to find relationships between the elements and strength.

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 in this dataset 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.



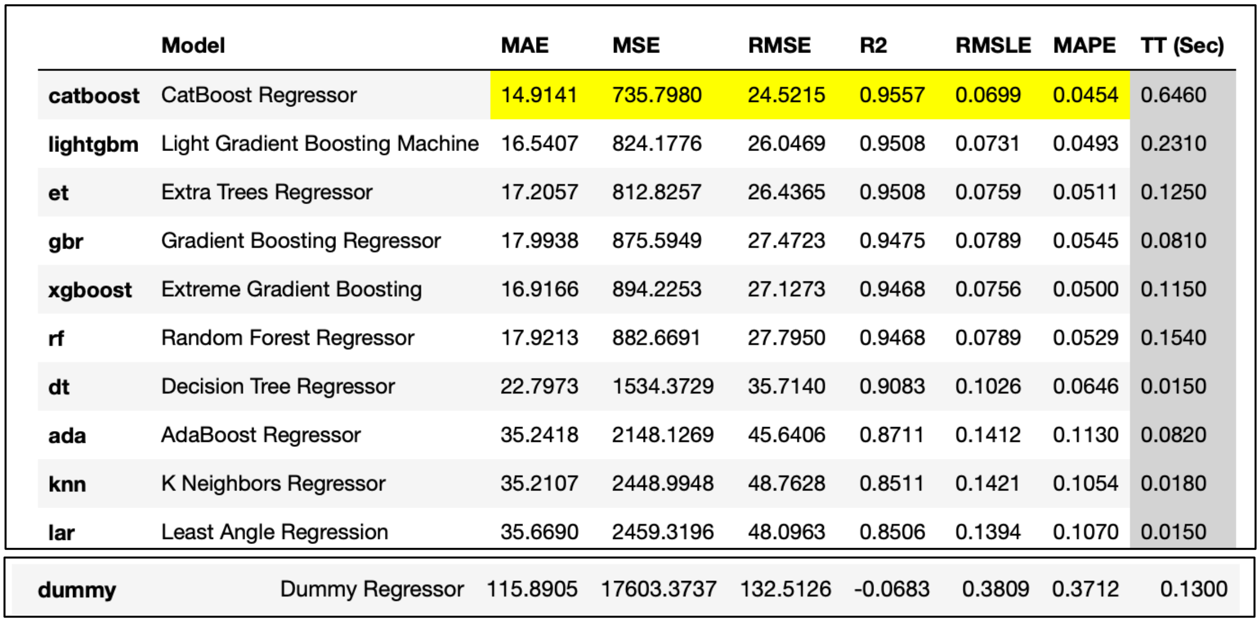
Preprocessing

The remaining data was split into training, test and validation sets with a 7:2:1 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.

Modelling

PyCaret 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:



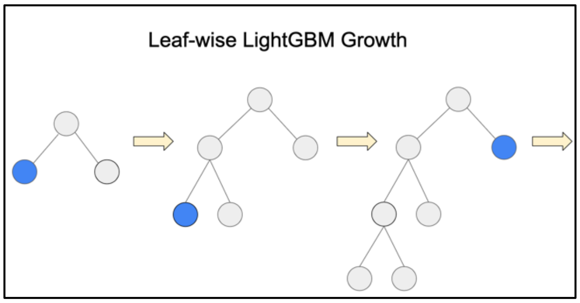
	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	14.9141	735.7980	24.5215	0.9557	0.0699	0.0454	0.6460
lightgbm	Light Gradient Boosting Machine	16.5407	824.1776	26.0469	0.9508	0.0731	0.0493	0.2310
et	Extra Trees Regressor	17.2057	812.8257	26.4365	0.9508	0.0759	0.0511	0.1250
gbr	Gradient Boosting Regressor	17.9938	875.5949	27.4723	0.9475	0.0789	0.0545	0.0810
xgboost	Extreme Gradient Boosting	16.9166	894.2253	27.1273	0.9468	0.0756	0.0500	0.1150
rf	Random Forest Regressor	17.9213	882.6691	27.7950	0.9468	0.0789	0.0529	0.1540
dt	Decision Tree Regressor	22.7973	1534.3729	35.7140	0.9083	0.1026	0.0646	0.0150
ada	AdaBoost Regressor	35.2418	2148.1269	45.6406	0.8711	0.1412	0.1130	0.0820
knn	K Neighbors Regressor	35.2107	2448.9948	48.7628	0.8511	0.1421	0.1054	0.0180
lar	Least Angle Regression	35.6690	2459.3196	48.0963	0.8506	0.1394	0.1070	0.0150
dummy	Dummy Regressor	115.8905	17603.3737	132.5126	-0.0683	0.3809	0.3712	0.1300

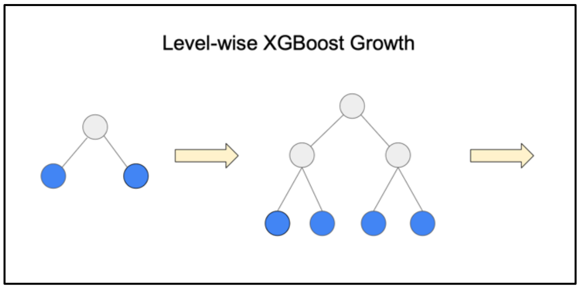
The Dummy Regressor as seen at the bottom was used as reference and had an MAE and RMSE of 116 and 49 which would render it unable to make accurate predictions, even in this case when rough estimates are required. The CatBoost Regressor, Light Gradient Boosting Machine and Extra Trees Regressor were chosen to be input into a Voting Regressor to be explained later on in this report. 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.



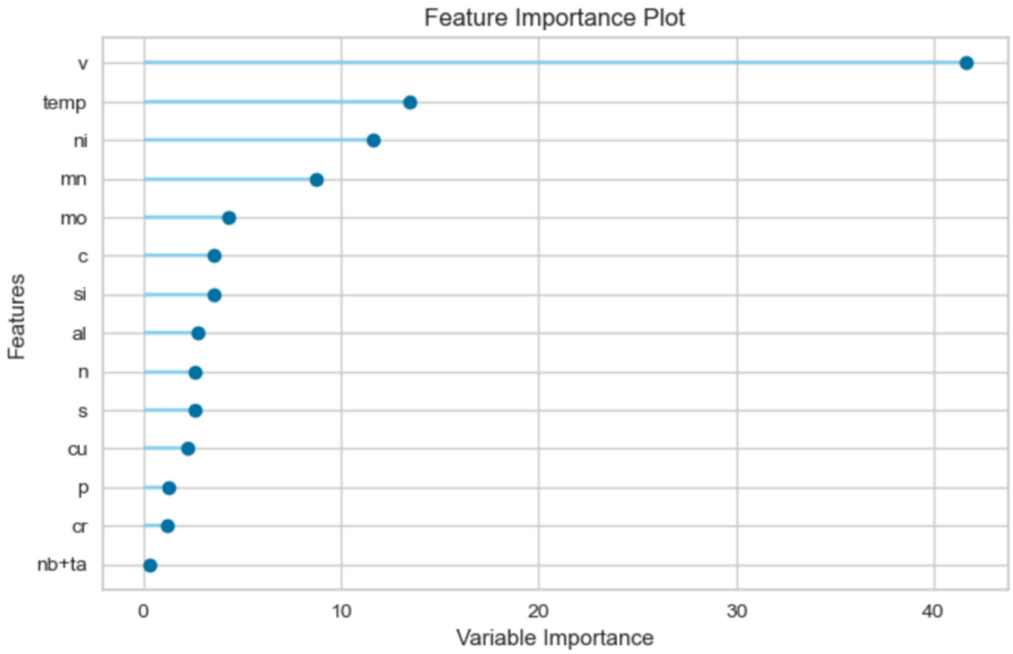


Extra Trees (XT) models are also an ensemble decision tree model like Random Forests. The differentiating factor is that decision trees in an XT model are trained on the entire dataset unlike the decision trees in Random Forests that are trained on bootstrapped samples. Nodes are also split randomly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

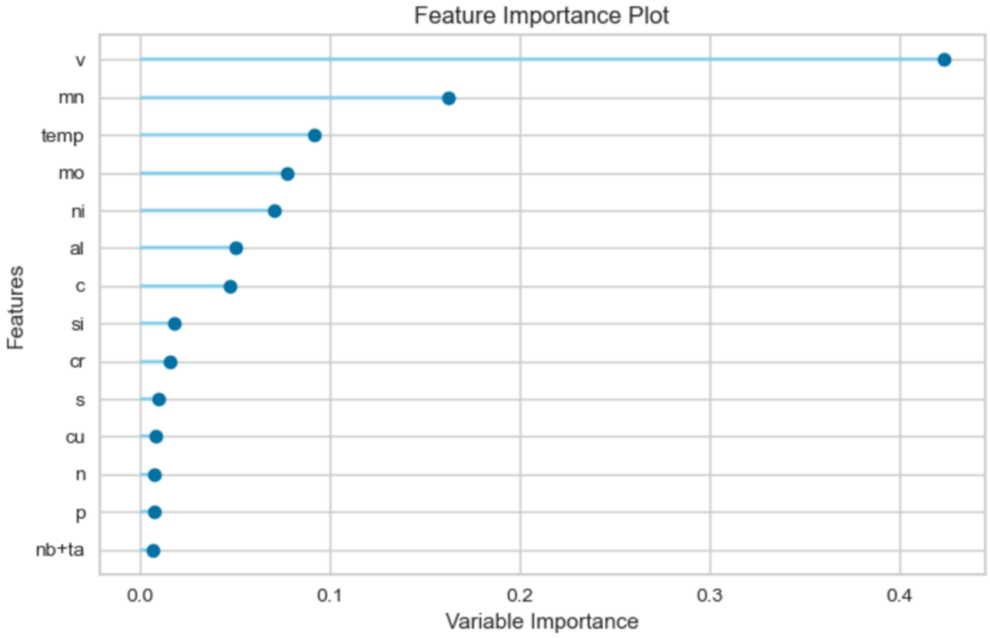
Feature Importance

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

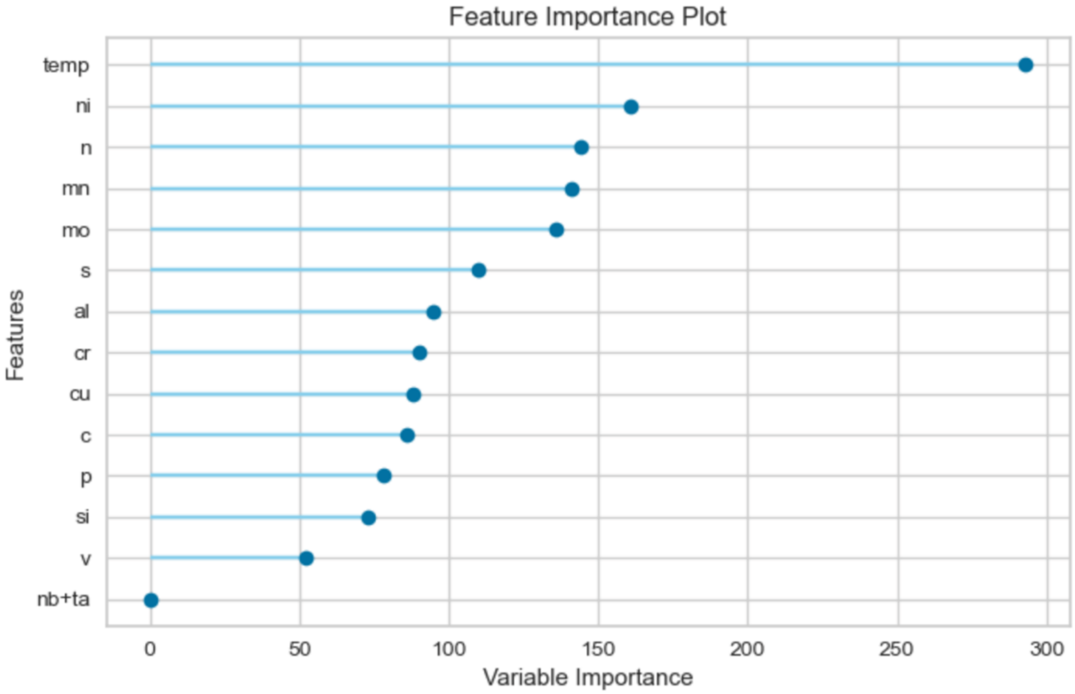
CatBoost Regressor (CAT):



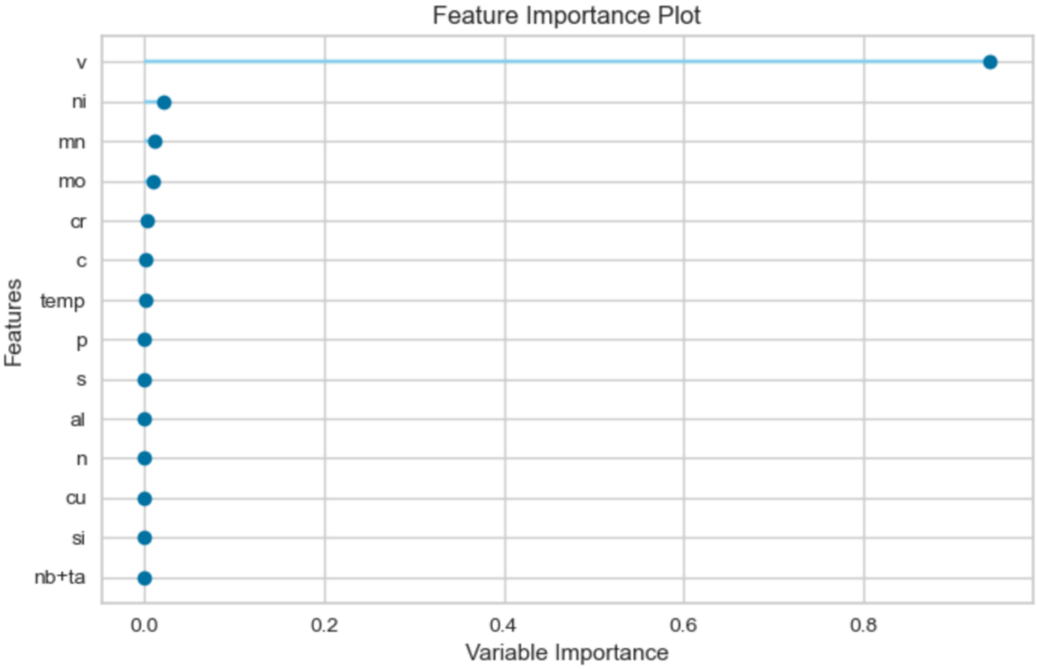
Extra Trees (XT):



Light Gradient Boosting Machine (LGBM):



XGBoost (XGB):



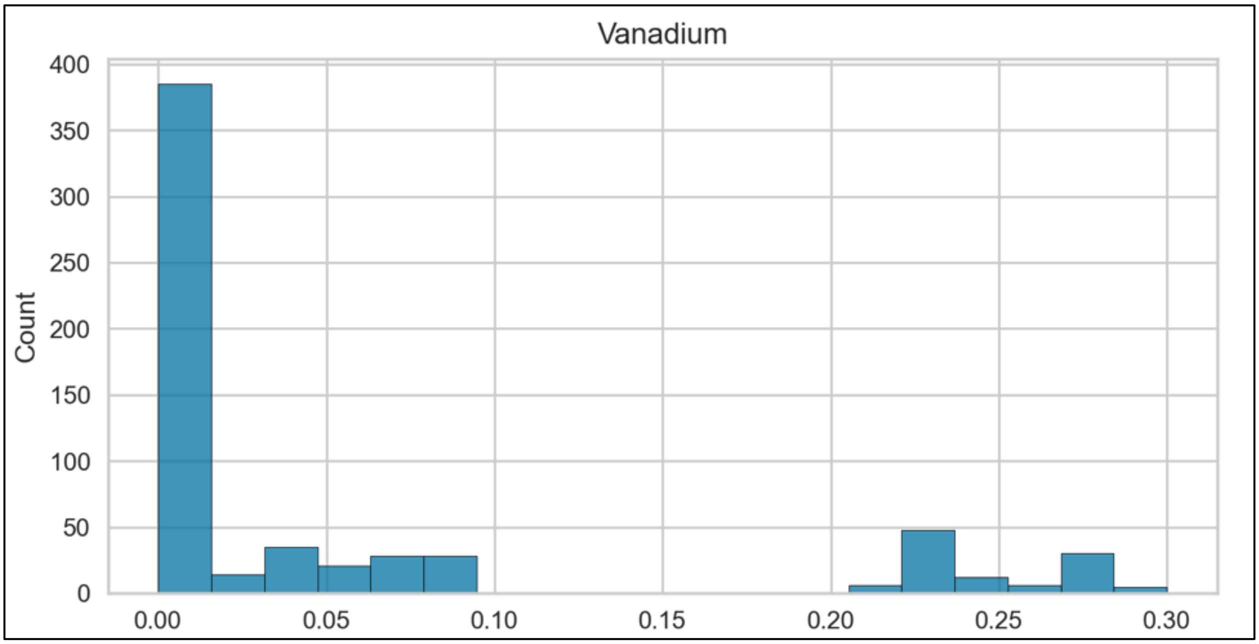
Feature importances by themselves don’t explain whether a feature negatively or positively influences model predictions. It explains magnitude of importance but not direction of its influence. However, this relationship can be deduced using the correlation plot.

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) – evident 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.

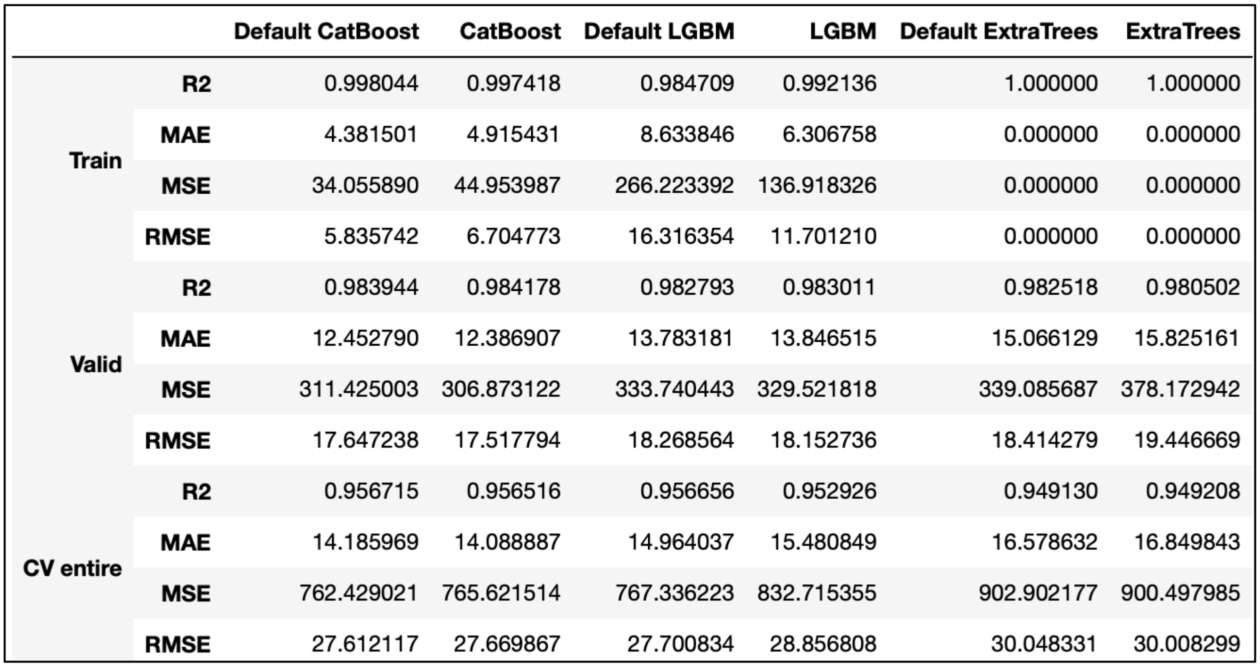
The LGBM model also places a large importance on a different feature, temperature. Although the goal is to minimize this feature’s influence on the model, the model does also place importance on elements such as Ni, Mn and Mo.



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][3].

Temperature plays a crucial role in reducing Yield strength. It is a big factor in the LGBM model especially. The impact is high but is negative as it seen in 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.

Hyperparameter Tuning



		Default CatBoost	CatBoost	Default LGBM	LGBM	Default ExtraTrees	ExtraTrees
Train	R2	0.998044	0.997418	0.984709	0.992136	1.000000	1.000000
	MAE	4.381501	4.915431	8.633846	6.306758	0.000000	0.000000
	MSE	34.055890	44.953987	266.223392	136.918326	0.000000	0.000000
	RMSE	5.835742	6.704773	16.316354	11.701210	0.000000	0.000000
Valid	R2	0.983944	0.984178	0.982793	0.983011	0.982518	0.980502
	MAE	12.452790	12.386907	13.783181	13.846515	15.066129	15.825161
	MSE	311.425003	306.873122	333.740443	329.521818	339.085687	378.172942
	RMSE	17.647238	17.517794	18.268564	18.152736	18.414279	19.446669
CV entire	R2	0.956715	0.956516	0.956656	0.952926	0.949130	0.949208
	MAE	14.185969	14.088887	14.964037	15.480849	16.578632	16.849843
	MSE	762.429021	765.621514	767.336223	832.715355	902.902177	900.497985
	RMSE	27.612117	27.669867	27.700834	28.856808	30.048331	30.008299

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 R^2 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 evident, 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.

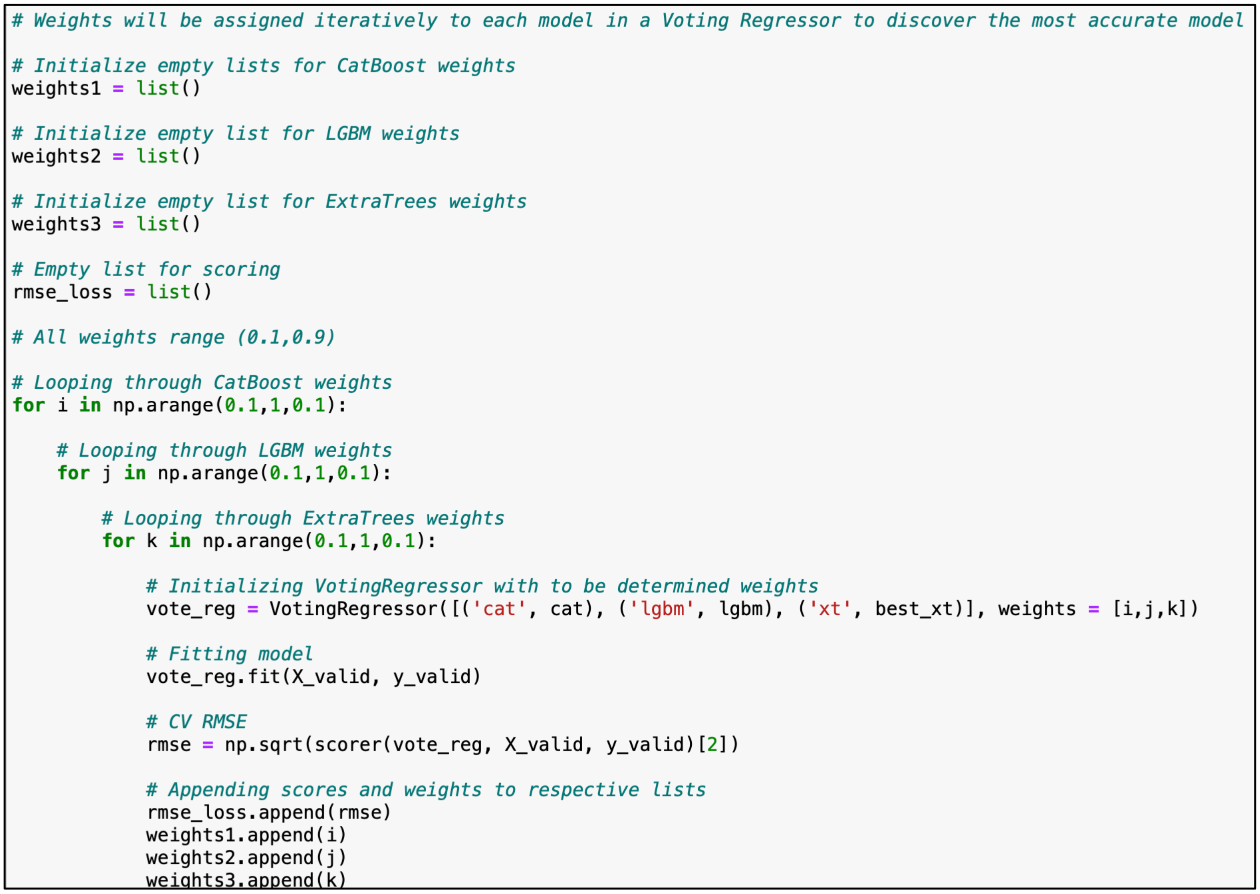
There wasn't much of a boost in performance between the tuned and untuned LGBM, therefore the default was chosen.

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 business use case.

Final Model

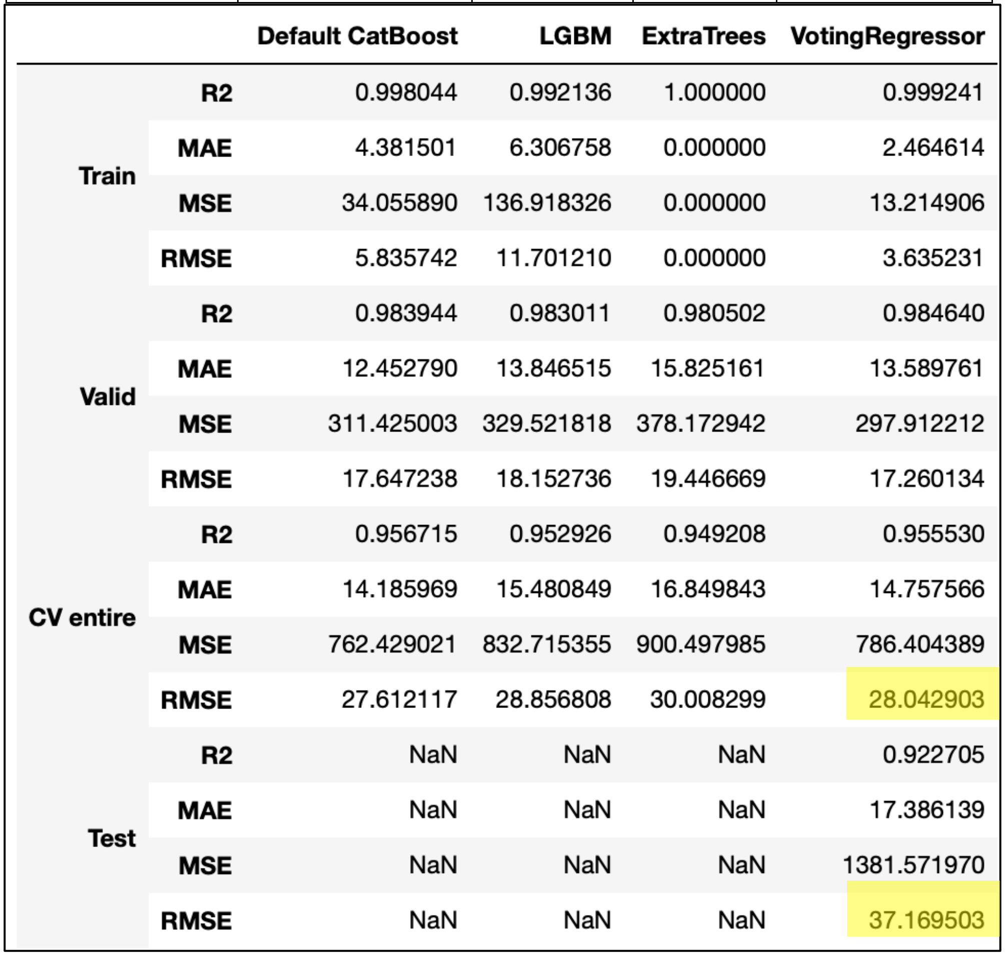
As mentioned above, a Voting Regressor was chosen to combine all models into what is known as an ensemble model. The advantage of using an ensemble model is its diversity. Incorrect predictions from an individual estimator are normalized by predictions from the others thereby increasing accuracy. Ensemble 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:



```
# Weights will be assigned iteratively to each model in a Voting Regressor to discover the most accurate model

# Initialize empty lists for CatBoost weights
weights1 = list()

# Initialize empty list for LGBM weights
weights2 = list()

# Initialize empty list for ExtraTrees weights
weights3 = list()

# Empty list for scoring
rmse_loss = list()

# All weights range (0.1,0.9)

# Looping through CatBoost weights
for i in np.arange(0.1,1,0.1):

    # Looping through LGBM weights
    for j in np.arange(0.1,1,0.1):

        # Looping through ExtraTrees weights
        for k in np.arange(0.1,1,0.1):

            # Initializing VotingRegressor with to be determined weights
            vote_reg = VotingRegressor([('cat', cat), ('lgbm', lgbm), ('xt', best_xt)], weights = [i,j,k])

            # Fitting model
            vote_reg.fit(X_valid, y_valid)

            # CV RMSE
            rmse = np.sqrt(scorer(vote_reg, X_valid, y_valid)[2])

            # Appending scores and weights to respective lists
            rmse_loss.append(rmse)
            weights1.append(i)
            weights2.append(j)
            weights3.append(k)
```

The weights corresponding to the highest RMSE used almost exclusively the predictions from XT which could pose a problem with data that XT performs poorly. Therefore, the weights picked for CAT, LGBM and XT were 0.4, 0.1 and 0.5 respectively. The final metrics are shown below:

Weight	0.4	0.1	0.5	= 1.0



		Default CatBoost	LGBM	ExtraTrees	VotingRegressor
Train	R2	0.998044	0.992136	1.000000	0.999241
	MAE	4.381501	6.306758	0.000000	2.464614
	MSE	34.055890	136.918326	0.000000	13.214906
	RMSE	5.835742	11.701210	0.000000	3.635231
Valid	R2	0.983944	0.983011	0.980502	0.984640
	MAE	12.452790	13.846515	15.825161	13.589761
	MSE	311.425003	329.521818	378.172942	297.912212
	RMSE	17.647238	18.152736	19.446669	17.260134
CV entire	R2	0.956715	0.952926	0.949208	0.955530
	MAE	14.185969	15.480849	16.849843	14.757566
	MSE	762.429021	832.715355	900.497985	786.404389
	RMSE	27.612117	28.856808	30.008299	28.042903
Test	R2	NaN	NaN	NaN	0.922705
	MAE	NaN	NaN	NaN	17.386139
	MSE	NaN	NaN	NaN	1381.571970
	RMSE	NaN	NaN	NaN	37.169503

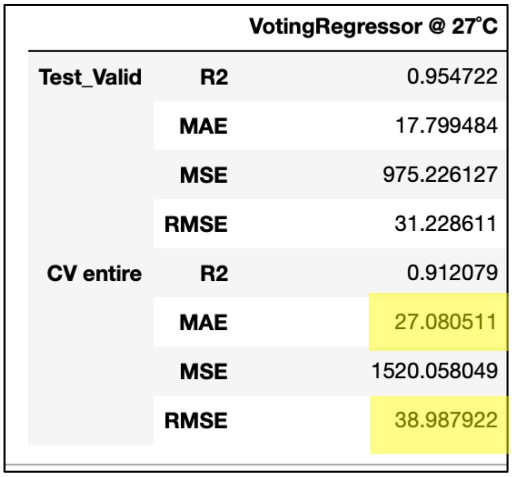
Both XT and CAT weigh a high importance on the elements that are the most correlated with steel strength. They also don't place as much importance on temperature which, as mentioned, isn't a very important feature. This ties in to the highest contributing features of LGBM. Even though it places a high importance on LGBM, it does weigh the other elements as important as well more equally than XT or CAT which is beneficial since XT's predictions are very reliant on Vanadium and less so for the other elements. Accuracy using these models could be an issue for datasets that don’t contain as much Vanadium especially because many of the samples do not contain Vanadium.

In this business use case, the model’s performance would be best judged using both MAE and RMSE. This is because metallurgists would only require a rough estimate of steel performance using this model. The Voting Regressor scored an MAE of ~14 MPa and an RMSE of ~28 MPa during cross-validation. This means that the Voting Regressor’s predictions are on average, ~14 MPa away from the true strengths. The R^2 was 0.96 meaning 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 cleared 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:



		VotingRegressor @ 27˚C
Test_Valid	R2	0.954722
	MAE	17.799484
	MSE	975.226127
	RMSE	31.228611
CV entire	R2	0.912079
	MAE	27.080511
	MSE	1520.058049
	RMSE	38.987922

When scored on the new test and validation data, the model still performed quite well. Its MAE was ~27 MPa which is similar to the MAE obtained from training on the data from all temperatures even though the R^2 did decrease to 0.91. However, the CV

MAE on this subset was higher than the CV MAE when the data from all temperatures was included (~27 MPa vs. ~14 MPa).

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 of 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. This is especially true because the model is over-eliant on on Vanadium. 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 but 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.

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] *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*