**Exploring the Composition-Property Relationship in Industrial Steels**

Group Members:

Hadi Bourji, 1449130

Hariharan Annadurai, 2214695

Matt Weigl, 2063513

**Abstract:**

Herein we provide an investigation into the relationship between the chemical compositions and mechanical properties—namely tensile strength—of industrially produced steels for a dataset of over 60,000 data points. Our investigation is of machine learning methods learned in our course with increasing complexity: linear regression, random forest, and neural network. All our methods are analyzed with mean absolute error (MAE) as to provide a comparison of model performance for statistics like mean, median, and standard deviation. The latter, more complex, methods include feature selection, which tells us which chemical components are most important to tensile strength. Both methods highlighted the role of titanium, among other transition metal alloying elements, which is to be expected and is in line with known metallurgy. Interestingly, both methods also highlight sulfur, which is a required addition in steel production, but not for purposes of enhancing mechanical properties.

**Background**:

The riddle of steel has been a secret and question of the metallurgists for millennia before the Industrial Revolution.  Indeed, even seen in the mythology of Conan The Barbarian, this riddle is called into question through Conan—played by the adventurous Arnold Schwarzenegger—in devotion to his God—Crom.  This riddle of steel and Crom—portrayed as the ultimate power of the time—is eventually conquered, yet upheld philosophically, which extends beyond this work.  Through the secrets of metallurgy, the Industrial Revolution blossomed, not through mythology, not through enigma, but through the early concepts of solid-state chemistry—of Bravais lattices and interstitial sites.  Indeed, early steel as we know it has been explained through the lattices of iron with interstitial carbon, and structural steel to this day is still largely iron based—which we shall explore later and further in our dataset.  But what of the non-structural steel of today?  In our contemporary era, steel like many other alloys have found their footing in the industrial environments through specific demands from industry.  For instance, stainless steel.  How does “stainless steel” look to our human eyes?  Glossy and clean- but chemically, what does that mean?  It is chemically quantified by the additions of chromium!  With the complexity that exists today, many additions occur in material development.

Our history has been divided into many, but a “Steel Age” never existed.  Perhaps it may be by later historians, but the contribution of steel will never be discounted by those with their eyes open.  In our world, steel is still valuable internationally.  It’s still being said that the measure of GDP is coupled with steel production and consumption.  Debatable as this is in 2023, as advantageous AI advances may be, the advances of software industries and AI are still coupled to our industries of physical production.  This directly begs the question: can machine learning methods help elucidate the structure-property relationship in steel?  If so, what are the most important factors to those mechanical properties?  Herein we shall investigate the relationship between the chemical composition in our dataset to the target variable of tensile strength.  Since the developments of the early 19th and 20th century, metallurgy moved from a chemical to a mechanical engineering discipline, and the major advancements of steel production have been quantified in their change in mechanical properties at the source of manufacturing.  Thus, this proves a good time to look back and reflect on that material which we owe so much of our infrastructure to with the tools and methods of our current age.

**Dataset**:

The dataset is composed of 60,000+ data points and obtained from the literature [1]. The dataset is composed of features including melting and processing conditions, chemical composition, and mechanical properties. To prevent the project from getting too bloated, we removed melt and processing conditions from the scope of investigation. Additionally, the scope of mechanical property investigation was narrowed from yield and tensile strength and elongation down to just tensile strength. As will be explained further in the Methods section, this was done due to the scaling of feature selection importance. Tensile strength was selected due to its importance and being the most extensively illustrative of the three properties in stress-strain relationships. Finally, we introduce our dataset in terms of statistics and **Figure 1**: The dataset can be described with a mean of 455.77, a standard deviation of 123.43, a minimum of 180.0, a maximum of 1319.0 and a median of 454.0.

A blue graph with numbers

Description automatically generated

**Figure 1**: *A histogram of the dataset is constructed to help understand the magnitude of our MAE.*

**Linear Regression**

Linear regression is the simplest of methods used for regression analysis. The relationship between the features and the targets are described with a straight line. As for other machine learning systems, once the model is trained extrapolation can be used to explore areas outside the training or inputted dataset. In the context of linear regression, the required findings are simply the slope and intercept of that linear system. For the purposes of our analysis linear regression was used as a baseline for the more complex methods. The linear regression was performed using Scikit Learn based on methods learned in the course and the MAE value of 50.6 [2]. In the context of the previously addressed statistics, this value is quite acceptable and a pleasant surprise. Below, as we introduce the more advanced methods, we shall lower that value and introduce which features are most important for tensile strength in steel.

**Random Forest Regressor**

The Random Forest Regressor is a machine learning algorithm used for regression tasks. It belongs to the ensemble learning family, combining the strengths of multiple decision trees to enhance predictive accuracy. It's an ensemble of decision trees as it builds multiple decision trees during training and merges their predictions to obtain a more robust model. Each decision tree is a base model which learns from the data independently. Randomness is introduced into the algorithm by training each tree on a random subset of features. This helps to decorrelate the trees and reduces overfitting. The algorithm uses bootstrapped sampling, meaning each tree is trained on a random subset of the dataset with replacement. This introduces diversity among the trees. 80% of the data was used to train the model and the rest was used for testing it. The random state of all the models we tested was set to 0 for reproducibility.

*Hyperparameters Tuning*

Hyperparameters are settings or values that are not learned from the data during the training process of a machine learning model. In other words, they are set prior to training and are used to control the learning process. Hyperparameters in this model include the numberof decision trees, depth of each tree, and number of features selected for bootstrapping. We optimized all these hyperparameters to build a robust model. During tuning, one of the hyper parameters is iterated over when other hyperparameters are set to default options. For instance when optimizing the number of estimators, depth of trees was set to None and number of features selected for bootstrapping was sqrt(number of features) since the dataset is huge.

Out of bag(OOB) error score was used to tune the hyperparameter: number of trees or estimators. Optimum number was identified to be 40, per **Figure 2a**. The OOB score provides an unbiased estimate of the model's performance on unseen data, similar to a validation set. It is useful for assessing how well the Random Forest generalizes to new data without the need for a separate validation set. Depth of tree was optimized with the help of grid search and the maximum depth was identified to be 20. When we ran the model with default options the depth was about 40 which may overfit the model **Figure 2b**.

A green bar graph with numbers

Description automatically generatedA graph of a number of trees

Description automatically generated

**Figure 2**: *(a, Right)* *Visualization of* h*yperparameter tuning via out-of-bag error score. (b, Left) Analysis of tree depth was used further to elucidate hyperparameter selection.*

Number of features selected for bootstrapping: As a rule of thumb since we had a huge dataset we choose sqrt of total number of features for bootstrapping. With these values for hyperparameters  and split ratio for test and train, the MAE on the test was about 17.22. We identified  the important features using the feature importance attribute and the top 10 features were taken to see the model performance, **Figure 3**.

With these top features we trained and tested the model which gives a  marginal effect on MAE(17.25, on test set). This indicates that the left out features are unimportant for the model training. After that, we performed PCA for dimension reduction and combined it with the model with optimum hyperparameters(optimized hyperparameters again using the same methods mentioned above). Including PCA did not improve the model performance by any margin. The MAE obtained was 17.73. Since it's not a classifier we don't have metrics like F1score or recall score to compare the model performance hence we used MAE as metric for consistent comparison. Generally, this method gave a threefold reduction in MAE compared to linear regression.

A graph of blue rectangular bars

Description automatically generated with medium confidence

**Figure 3:** *Feature selection of chemical components in steel per Random Forest Method.*

**Neural Network:**

The dataset was loaded and PCA was performed to find how few components we can use while maintaining a good amount of the information in the training set. The explained variance ratio was calculated for ‘n\_components’ values ranging from one to twenty. The EVR value seemed to even out with the number of components equal to seven, seen in **Figure 4**. The information from a seven component PCA fitting was then split with 80% of the set going towards training and 20% going to testing.

A graph with a blue line

Description automatically generated

**Figure 4**: *EVR vs. Number of Components for PCA selection in further study.*

We then test the hyperparameters of the neural network. We begin by testing a variety of regularization parameters, from 10-1 to 10-6. We trained a model with 50 nodes in one hidden layer for 1000 steps with each of these regularization parameters and found the model with a regularization parameter of 10-1 produced the lowest testing error. Now that the optimal regularization parameter has been found, we use it to train models of differing size. We train models with one, two, or three hidden layers each containing 50, 100, 150, and 200 nodes. Out of these twelve models, the model with the highest accuracy was the model with three hidden layers, each containing 100 nodes.

A graph with a line

Description automatically generatedA bar graph with blue lines

Description automatically generated

**Figure 5**: *Data visualization for hyperparameter selection.*

Now that the hyperparameters have been determined we train a neural network for 50,000 steps to determine how long the model should be trained for before overfitting becomes an issue. We save the testing error at each step, and then calculate the average testing error every 1000 steps. We find the testing error reaches a minimum of 24.455 at 36,000 steps into the training. **Figure 6** below visualizes the training and testing error during the training for the dataset with PCA applied.

A graph of a training and testing

Description automatically generated

**Figure 6**: *Visualization of training and testing loss.*

Next, we want to compare the accuracy of the set with PCA applied to it, to the accuracy of the original training set with and without feature selection. To do this we began by training on the original dataset and determined the minimum testing error to be 19.841 at step 15,000 of the training. We then perform feature selection using this model and calculate the importance mean of each feature. The four features that have a relatively small importance value are then removed from the dataset, and the training is run on this smaller dataset. During the training of this model with feature selection, the testing error was minimized at step 7000 with an error of 19.380 indicating a slight increase in model accuracy when we removed the features found to be least important. Again, this reduction in MAE compared to linear regression is of similar order to the random forest results, while underperforming slightly to the latter.

A graph of a number of periodic table

Description automatically generated with medium confidence

**Figure 7**: *Feature selection for neural network method*.

**Conclusion:**

First, the reduction in MAE values observed in the more complex random forest and neural network methods compared to linear regression is expected and somewhat indicative of successful investigations. Also, importantly, is somewhat similar results in the feature selection- particularly the prevalence of titanium. Titanium is known to improve tensile strength due to microstructure and grain size control. Titanium forms precipitates with carbon and nitrogen, which enhances mechanical properties by strengthening the grain boundary [3]. Additionally, other transition metals are prevalent in both feature selection outputs, as one would expect. Of interest to both outputs is the performance of sulfur. Traditionally, sulfur is merely added along with phosphorous and silicon as “residual” elements rather than alloying [3]. This distinction is important because the addition of residual elements is for deoxidation purposes following industry standards as opposed to mechanical improvements. Thus, a natural extension of this study would be to further explore the significance of sulfur in industrial steels as we have seen in our results.

**Contributions**:

Matt Weigl: Project concept, project management, literature research, data acquisition, data cleaning, code (linear regression), writing (aside from random forest and neural network), editing.

Hadi Bourji: Code (reading data, splitting train/test, finding optimal components for PCA, neural networks), visualization for neural network section, general data statistics, writing (neural network section)

**References**:

1. Guo, S., Yu, J., Liu, X., Wang, C., & Jiang, Q. (2019). A predicting model for properties of steel using the industrial big data based on machine learning. *Computational Materials Science*, *160*, 95–104. https://doi.org/10.1016/j.commatsci.2018.12.056
2. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html>
3. Campbell, F. C. (2008). *Elements of metallurgy and Engineering Alloys*. ASM International.