Metallurgica Hackathon 2025

PREDICTING ELECTRICAL CONDUCTIVITY (%IACS) OF ALLOYS

REPORT

1. Dataset Overview and Initial Observations

1.1 Dataset Description

The dataset consists of **1600 entries** in the training set and an almost **350 entries** in the test set . It includes **41 columns**, categorized as follows:

- **Features:**
  - **Alloy formula**: A string representing the chemical composition (e.g., Cu-6Ni-1Si-0.5Al-0.15Mg-0.1Cr).
  - **Alloy class**: Categorical variable indicating alloy type (e.g., Cu low alloyed, Cu-Ni-Si alloys, Cu-Ti alloys, Cu-Be alloys).
  - **Elemental Composition**: 27 numerical columns representing weight fractions of elements (e.g. Cu ,Al , Ag etc.).
  - **Processing Conditions**: 7 columns, including Tss (K), tss (h), CR reduction (%), Aging, Tag (K), tag (h), and Secondary thermo-mechanical process.
- **Target Variable**: Electrical conductivity (%IACS), a continuous variable representing the alloy's electrical conductivity.
- **Additional Properties**: Hardness (HV), Yield strength (MPa), and Ultimate tensile strength (MPa).

1.2 Dataset Problems

1.2.1 Missing Values:

- Several columns contain missing data, with varying degrees of severity:
  - Yield strength (MPa): Missing in ~90% of rows (160 non-null out of 1600).
  - Ultimate tensile strength (MPa): Missing in ~85% of rows (247 non-null out of 1600).
  - Hardness (HV): Missing in ~11% of rows (1425 non-null).
  - Tss (K): Missing in 42 rows (~2.6%).
  - tss (h): Missing in 68 rows (~4.3%).
  - Tag (K): Missing in 59 rows (~3.7%).
  - tag (h): Missing in 85 rows (~5.3%).
  - Secondary thermo-mechanical process: Missing in 28 rows (~1.8%).
- The target variable, Electrical conductivity (%IACS), has 2 missing values (1598 non-null).

1.2.2 High Dimensionality:

- With 27 elemental composition columns and additional processing parameters, the dataset is high-dimensional, increasing the risk of overfitting and computational complexity.

1.2.3 Sparse Data:

- Many elemental composition columns (e.g., Hf, La, Mn, Mo, Nb, Nd, Pb, Pr) are predominantly zero, indicating sparse usage of these elements across alloys.

1.2.4 Categorical Variables:

- Columns like Alloy class, Aging, and Secondary thermo-mechanical process are categorical and require encoding for machine learning models.

1.2.5 Target Distribution:

- The electrical conductivity values typically vary widely depending on alloy composition and processing, potentially requiring careful model selection to handle non-linear relationships.

2. Data Preprocessing and Problem Handling

2.1 Handling Missing Values in the Target

- We dropped rows where the target variable (Electrical conductivity (%IACS)) is missing, reducing the training set from 1600 to **1598 rows**. This ensures a complete target dataset for supervised learning.

2.2 Dropping High-Missing-Value Columns

- Columns with more than 80% missing values (Yield strength (MPa) and Ultimate tensile strength (MPa)) are removed entirely. This is a reasonable choice, as imputing such sparse columns could introduce significant noise, and their correlation with the target is not assessed due to limited data.

2.3 Preprocessing Pipeline

- The remaining missing values in numerical and categorical columns are addressed using a ColumnTransformer:
  - **Numerical Columns**: Missing values are imputed with the mean using **SimpleImputer(strategy='mean'),** followed by standardization with **StandardScaler()** to normalize the data.
  - **Categorical Columns**: Missing values are filled with the string 'missing' using **SimpleImputer(strategy='constant'),** followed by one-hot encoding with **OneHotEncoder(handle_unknown='ignore')** to convert categories into binary features.

2.4 Column Selection

- We dropped drops ID and Alloy formula from the feature set:
  - ID is irrelevant for prediction.
  - Alloy formula is redundant since its information is already captured in the elemental composition columns.
- The remaining features are split into:
  - **Categorical**: Alloy class, Aging, Secondary thermo-mechanical process.
  - **Numerical**: 33 columns (27 elemental + 6 processing parameters).

---

3. Feature Engineering

The notebook implements a custom engineer_features function to create new features that might enhance the model's ability to predict electrical conductivity. These features leverage domain knowledge about metallurgy and conductivity:

3.1 Cu_Al_ratio:

- Formula: $Cu/(Al + 0.001)$
- Rationale: Copper (Cu) is highly conductive, while aluminum (Al) is less so. This ratio captures the relative dominance of Cu over Al, with a small constant (0.001) added to avoid division by zero.

3.2 conductive_elements_sum:

- Formula: $Cu + Ag$
- Rationale: Both copper (Cu) and silver (Ag) are highly conductive elements. Summing their contributions highlights their combined effect on conductivity. (Note: The code checks for Au (gold), but it’s not present in the dataset, so only Cu and Ag are used.)

3.3 total_alloying:

- Formula: Sum of all numerical columns (elemental compositions + processing parameters).
- Rationale: This represents the total influence of alloying elements and processing conditions, potentially capturing overall complexity or dilution of pure copper.

3.4 thermal_factor:

- Formula: $Tss(K) * np.log1p(tss(h))$
- Rationale: Combines solution treatment temperature and time (log-transformed to handle skewness) into a single feature, reflecting the thermal processing impact on microstructure and conductivity.

3.5 aging_factor:

- Formula: $Tag(K) * np.log1p(tag(h))$
- Rationale: Similar to thermal_factor, this combines aging temperature and time, capturing the effect of aging treatment on precipitation and conductivity.

3.6 hardness_yield_ratio :

- Formula: Hardness $(HV)/(Yieldstrength(MPa) + 0.001)$
- Rationale: Intended to capture the relationship between hardness and strength, which might indirectly relate to conductivity. However, since Yield strength (MPa) was dropped due to high missingness, this feature is not computed.

These engineered features are applied to both the training and test datasets, increasing the feature set from 36 (after dropping ID, Alloy formula, and high-missing columns) to **41 features**.

---

4. Modeling Approach

4.1 Best Model: CatBoostRegressor

- We used a CatBoostRegressor within a Pipeline that includes the preprocessing steps. Key parameters:
  - **iterations**: 500 (number of boosting iterations).
  - **learning_rate**: 0.03 (step size for gradient descent).
  - **depth**: 3 (maximum depth of trees, controlling model complexity).
  - **loss_function**: 'MAE' (Mean Absolute Error), aligning with the evaluation metric.
  - **verbose**: 100 (prints training progress every 100 iterations).
  - **random_seed**: 42 (for reproducibility).
- **Why CatBoost?**
  - CatBoost is chosen for its ability to handle categorical features natively (though one-hot encoding is still used here), robustness to missing data, and strong performance on regression tasks with complex interactions.

4.2 Other Model Exploration

- We also used a StackingRegressor with multiple base models, including:
  - RandomForestRegressor (n_estimators=200)
  - GradientBoostingRegressor (n_estimators=200)
  - XGBRegressor (n_estimators=200, learning_rate=0.05)
  - LGBMRegressor (n_estimators=200, learning_rate=0.05)
  - Ridge (alpha=1.0)
  - Lasso (alpha=0.01)
  - ElasticNet (alpha=0.01, l1_ratio=0.5)
- Final estimator: Ridge
- We hoped to get better resluts but we failed and it could not beat our Catboostregressor model.

4.3 Training and Validation

- The dataset is split into training (80%) and validation (20%) sets using train_test_split (test_size=0.2).
- The model is trained on the training set, and performance is evaluated on the validation set using **MAE**:
  - Validation MAE: **13.3832**.
- The final model is retrained on the entire training dataset (train_features) for submission predictions.

4.4 Pipeline Structure

- **Preprocessor**: Applies imputation and scaling/encoding as described earlier.
- **Model**: CatBoostRegressor with the specified hyperparameters.

- This pipeline ensures consistent preprocessing and modeling across training and test data.

---

5. Evaluation and Submission

- **Validation MAE**: 13.3832, indicating the model's average prediction error on the validation set.
- **Test Predictions**: Generated on the test set (test_features) using the fully trained model.
- **Submission File**: A CSV file (submission.csv) is created with columns ID and Electrical conductivity (%IACS), containing predictions for the test set.

---

6. Analysis of Results and Potential Improvements

6.1 Strengths

1. **Effective Missing Value Handling**: Dropping sparse columns and imputing others ensures a robust dataset.
2. **Feature Engineering**: The new features leverage domain knowledge (e.g., Cu’s role in conductivity, thermal processing effects), potentially improving model performance.
3. **Model Choice**: CatBoost is well-suited for this task due to its handling of non-linear relationships and robustness.

6.2 Weaknesses and Challenges

1. **Small Validation Set**: A 1% validation set (16 samples) is insufficient for reliable performance estimation, risking overfitting to the training data.
2. **Limited Correlation Analysis**: We computed correlations but does not deeply explore feature importance or target distribution, which could guide feature selection.
3. **Feature Redundancy**: total_alloying sums all numerical columns, including processing parameters, which may introduce noise rather than signal.
4. **Hyperparameter Tuning**: The CatBoost parameters are fixed without optimization (e.g., via GridSearchCV), potentially leaving performance on the table.

6.3 Potential Improvements

1. **Increase Validation Size**: Use a 20-30% validation split for more reliable evaluation.
2. **Feature Selection**: Use feature importance from CatBoost or correlation analysis to drop irrelevant features (e.g., sparse elements like Hf, La).
3. **Hyperparameter Tuning**: Implement GridSearchCV or RandomizedSearchCV to optimize iterations, learning_rate, and depth.
4. **Target Transformation**: Analyze the target distribution and apply transformations (e.g., log) if skewed.
5. **Ensemble Revival**: Revisit the stacking approach with tuned base models for potential performance gains.

---

7. Conclusion

We effectively tackled the competition by addressing data challenges (missing values, high dimensionality) through preprocessing and feature engineering, then applying a robust **CatBoost** model to predict electrical conductivity.

We got **MAE** of **13.3832** on the validation set and on submitting we achieved a **MAE** score of **13.3770.**

The engineered features and preprocessing pipeline demonstrate thoughtful handling of metallurgical data, while the choice of CatBoost aligns with the task’s complexity. With further tuning and validation refinements, the solution could be enhanced for better leaderboard performance.