Alloy Analyzer: Predictive Modeling of Casting Properties

Project Overview:
The application allows users to load steel alloy data, train models to predict various properties, and evaluate model performance based on Mean Squared Error (MSE). It provides a user-friendly interface to interact with data, visualize errors, and load trained models for predictions.

Regression Model and GMDH Algorithm.
Regression Model (Version 1): In this context, regression models could refer to any model used to predict a quantitative outcome. Typically, regression models are evaluated using MSE, which measures the average of the squares of the errors—that is, the average squared difference between the observed actual outcomes and those predicted by the model.

GMDH Algorithm (Version 2 + Streamlit app): The GMDH is a **type of neural network that automates the modeling process** to identify the most predictive variables and determine the structure of the model itself. This approach is particularly advantageous for complex, multivariate datasets where relationships between variables might be nonlinear.
Unlike traditional regression models that often require manual selection of interactions and transformations, the GMDH algorithm automates these decisions, making it highly effective for discovering underlying patterns in complex data without overfitting. This results in potentially higher predictive accuracy and model robustness.

Comparison to Regression Models
Traditional regression models might provide a straightforward, interpretable framework for prediction but can fall short in handling nonlinear relationships unless explicitly modeled. GMDH, with its self-organizing capabilities, excels in scenarios where the data relationships are complex and not well-understood beforehand, potentially outperforming simpler regression models in such contexts.

In this project was used Joblib

Efficiency in Serialization
Joblib is particularly efficient at serializing Python objects that contain large data, especially numpy arrays. This makes it ideal for saving machine learning models that often involve large amounts of data and model parameters.

Joblib provides a simple interface for saving and reloading Python objects, which simplifies the process of model persistence. This allows you to save your trained models to disk with minimal code and then load them later as needed without retraining.T his can be particularly advantageous when dealing with complex models such as those generated by the GMDH algorithm.
By saving your trained models using Joblib, you make it possible to reuse these models for predictions on new data without the need to retrain.

The Results_data folder in this project stores serialized models (and possibly other outputs like scalers) that have been trained on my data.
If you have a new dataset and you need to make predictions without any changes to the model structure or retraining, you can simply load the pre-trained model from the ***Results_data*** folder and use it to make predictions. This is particularly useful for operationalizing your machine learning workflow in a production environment.
If you are developing new models or experimenting with different machine learning algorithms, you can use the stored models as benchmarks.

Regression Models	GMDH Algorithm
Regression models, especially linear regression, assume a linear relationship between the input variables and the target. They might underperform if the underlying data structure is complex or non-linear.	GMDH is a type of neural network algorithm that effectively models non-linear and complex relationships through its polynomial structure. It constructs an iterative network that can capture more intricate patterns in the data.
These models can include interactions but typically require manual specification of which interactions to include.	Automatically considers combinations of inputs through its polynomial terms, effectively learning which interactions are most predictive of the outcome.
Can be prone to overfitting especially if too many predictors are used, or they can be too simplistic, missing out on underlying patterns.	Uses a self-organizing approach, where it starts with a simple model and increases complexity only when beneficial.
The training times of the GMDH layers increase as the model complexity grows, with times ranging from 0.06 seconds for the first layer to around 1.16 seconds for the most complex layers.	
The MSEs for different elements using regression are significantly higher than those achieved by the GMDH model. MSE for w: 1.510297792372733 MSE for al: 0.26714664308621694 MSE for ti: 0.12480798374420149 The coefficients show the influence of each feature on the target variable, but the higher MSE indicates that the model does not predict as accurately as the GMDH model.	The MSE values provided for the GMDH algorithm are extremely low, ranging from 2.46397767903223×10 $2.46397767903223\times10^{-5}$ to $3.561976679748538\times10^{-6}$ $3.561976679748538\times10^{-6}$ These values are indicative of very high model accuracy on the testing dataset.
The regression model shows much higher	The GMDH algorithm outperforms the

MSE values, indicating lower prediction accuracy.	traditional regression model significantly in terms of MSE. The MSEs for GMDH are near zero, which suggests almost perfect prediction accuracy on the test data used.
For simpler, quicker models where interpretability is more critical, or computational resources are a constraint, traditional regression might still be preferred despite its lower accuracy.	If the goal is to maximize prediction accuracy and the dataset complexity justifies a more sophisticated approach, GMDH is clearly the better choice. It provides much lower MSEs, indicating better performance on the test set.

Conclusion

In this project, the GMDH algorithm is used as an advanced alternative to traditional regression models, offering enhanced automation in model selection and the ability to handle complex, nonlinear data efficiently. This makes it particularly suitable for predicting properties of steel alloys where such complexities are present.