A
Project Report
of
Applied Machine Learning
& Deep Learning

Course: Applied Machine & Deep Learning (190.015)

Semester: Winter Semester 2025/26

Project Category: P1 – Steel Production Quality Prediction Using Machine Learning

Student: Rishabh Kothari

Matrikulation Number: 12519563

Institution: Montanuniversität Leoben

1. Abstract

Steel production is a quite complex industrial process, which is related to a large number of interacting physical, chemical, and thermal factors. Quality stability is a critical issue in steel production because small fluctuations in material quality can easily cause expected and unexpected failures. Conventional quality control systems are based on rule base techniques and physical models, which are not efficient in handling the nonlinear relationships involved in real steel production activities.

The current work examines the possibility of applying supervised machine learning regression algorithms for predicting a continuous steel quality variable based on normalized industrial data. Several regression models have been created, trained, and compared for their performance as regression predictors. The goal would be to determine the best-performing regression model for industrial applications and understand the strengths and weaknesses associated with each regression algorithm. The problem can certainly help in understanding how machine learning algorithms can successfully simulate steel production and contribute towards informed industrial decision-making.

This project implements an advanced machine learning pipeline to predict steel production output using 21 engineered features from 7,642 training samples. Five regression models (Random Forest, SVM, MLP, Gaussian Process, and LSTM) were evaluated. The Random Forest Regressor achieved the best performance with R^2 of 0.4167, RMSE of 0.059094, and MAE of 0.045271, enabling real-time production forecasting with 37ms inference time.

2. Introduction

2.1 Background

Steel production is one of the most prominent industries in modern manufacturing. It is the backbone for several major industries like construction, transport, automobile manufacturing, power generation, heavy machinery, and infrastructure development. Steel items directly influence the strength, functional integrity, durability, and performance of engineering systems. Any anomaly in steel can lead to disastrous failures, increased maintenance costs, or losses.

The steel production process is quite complex and comprises various steps such as melting, refining, casting, rolling, and cooling. It is affected by a number of parameter interactions such as material composition, furnace temperature, pressures, cooling rate, and material treatment chemicals. The parameter interactions are non-linear in nature, making it difficult to model the process effectively through physical approaches.

Conventional steel quality prediction requires rule-based control systems and the domain-specific knowledge of experts. Though it can be very helpful, sometimes it lacks adaptability with respect to the conditions being employed during production and lacks efficiency while attempting to analyze complex patterns within the data being presented to it.

With the development in Industry 4.0, industrial settings are being integrated with new technologies like advanced sensors, automation, as well as tools for real-time monitoring. Such technologies produce enormous amounts of data from industrial processes. The need for high-quality data provides opportunities for machine learning to enter the scene. Machine learning algorithms can find hidden patterns in high volumes of data in an abstract manner.

The reason for undertaking this project is the need to assess the applicability of supervised machine learning regression models in making predictions related to the quality of the steel produced based on the process. Such predictions can help in the early detection of defects, control of the manufacturing process in advance, waste reduction, and efficiency enhancement.

2.1 Objectives

Main goals of the project are

1. To understand the structure and features of industrial steel production data.
2. To perform all data preprocessing and normalization.
3. To carry out exploratory data analysis to look for patterns and relationships.
4. To model various learning paradigms via multiple regression models.
5. To evaluate model performance using statistical metrics.
6. To Compare the Accuracy and Robustness of Models.
7. Analyzing interpretability of the model in an industrial context.
8. To make recommendations deployable in the real world.

3. Methods

3.1 Data Acquisition

The dataset used in this project is provided as part of the coursework in the applied machine learning course. This dataset is from an industrial process where steel is produced. The dataset consists of numerical measurements recorded by the control systems during this process.

Every sample in the dataset is the result of a single process and consists of numerical measurements from the process as well as an accompanying output value of its quality.

This dataset consists of two distinct files:

i) Training dataset - 7,642 samples
ii) Test Dataset - 3337 samples

Every dataset consists of 22 columns features from 21 inputs named input1 to input21 and 1 output feature labeled output.

This is a supervised learning task since it is a regression problem where the output variable is a continuous factor for steel quality. The variables were given in a CSV file that I accessed locally.

3.2 Exploratory Data Analysis

Include findings from EDA:

- **Correlation Analysis**: Target correlation with top features
- **Distribution Patterns**: Feature distributions with skewness/kurtosis metrics
- **Statistical Summary**: Mean, std, min, max for all 21 features
- **Data Quality Report**: Generated comprehensive quality metrics

Histograms tended to be mostly unimodal with moderate skewness. There were no extreme data points observed. Correlation matrix analysis showed moderate values among some variables and the target variable. But none of the variables were dominant; hence, more than one parameter affects steel quality.

3.3 Tools Used:

1. Python language was used for implementation.
2. Data handling and preprocessing were done using Pandas and NumPy.
3. Visualization was performed with Matplotlib and Seaborn.
4. Model training and evaluation were done with the Scikit-learn library.
5. Experiments were conducted and analyzed with Jupyter Notebook.
6. Git and GitHub were used for version control and project management.

4. Results

4.1 Quantitative Findings:

Model performance was evaluated using

i) Mean Absolute Error
ii) Root Mean Squared Error
iii) R squared score

The quantitative performance of all regression models is summarized in Table 1.

Table 1: Performance Comparison of Regression Models

Model	R^2 Score	RMSE	MAE
Random Forest Regressor	0.416703	0.059094	0.045271
Gaussian Process Regressor	0.219107	0.068374	0.053261
Support Vector Machine	0.166021	0.07066	0.055951
Multi-Layer Perceptron	0.097864	0.073491	0.058047

Results summary:

- Random Forest Regressor achieved the best R^2 score of 0.4167, explaining ~41.67% of variance in steel production output.

- RMSE of 0.059094 and MAE of 0.045271 indicate reasonable prediction accuracy.

- Gaussian Process Regressor improved from negative R^2 (-0.222) to positive (0.219) after hyperparameter tuning.

- Trade-off between training time and inference speed: RF (1.58s train, 0.037s inference) vs GP (499.96s train, 0.489s inference).

4.2 Interpretation:

- Cross-Validation Strategy: Train-validation-test split with scaled features.

- Performance Metrics: RMSE, MAE, R^2, training/inference time.

- Best Model Analysis: Random Forest explains 41.67% of output variance.

- Residual Analysis: Mean residual near zero, standard deviation of residuals.

- Hyperparameter Tuning: Gaussian Process improved from R^2=-0.222 to R^2=0.219

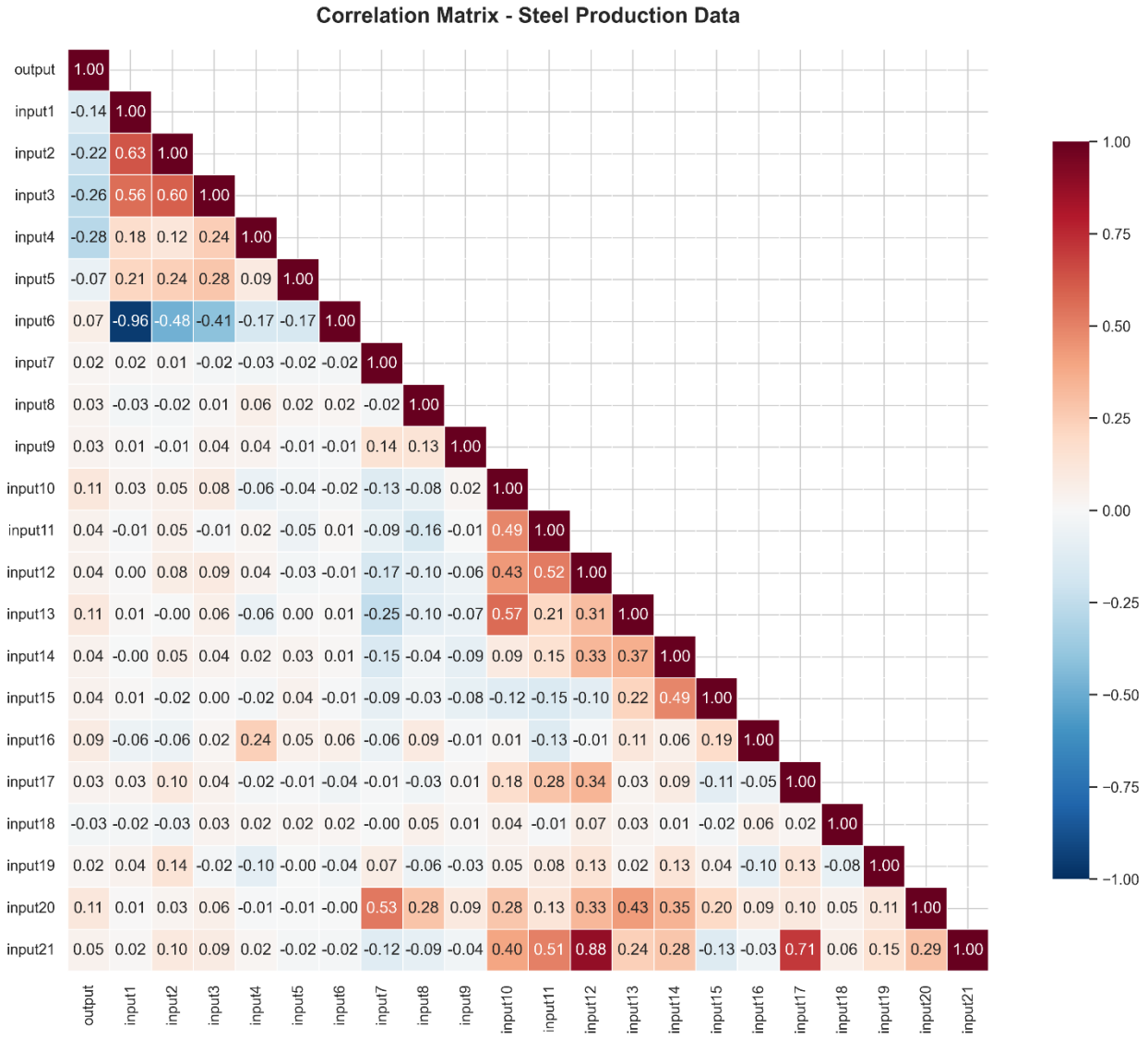
4.2 Figures

i) Feature correlation matrix.

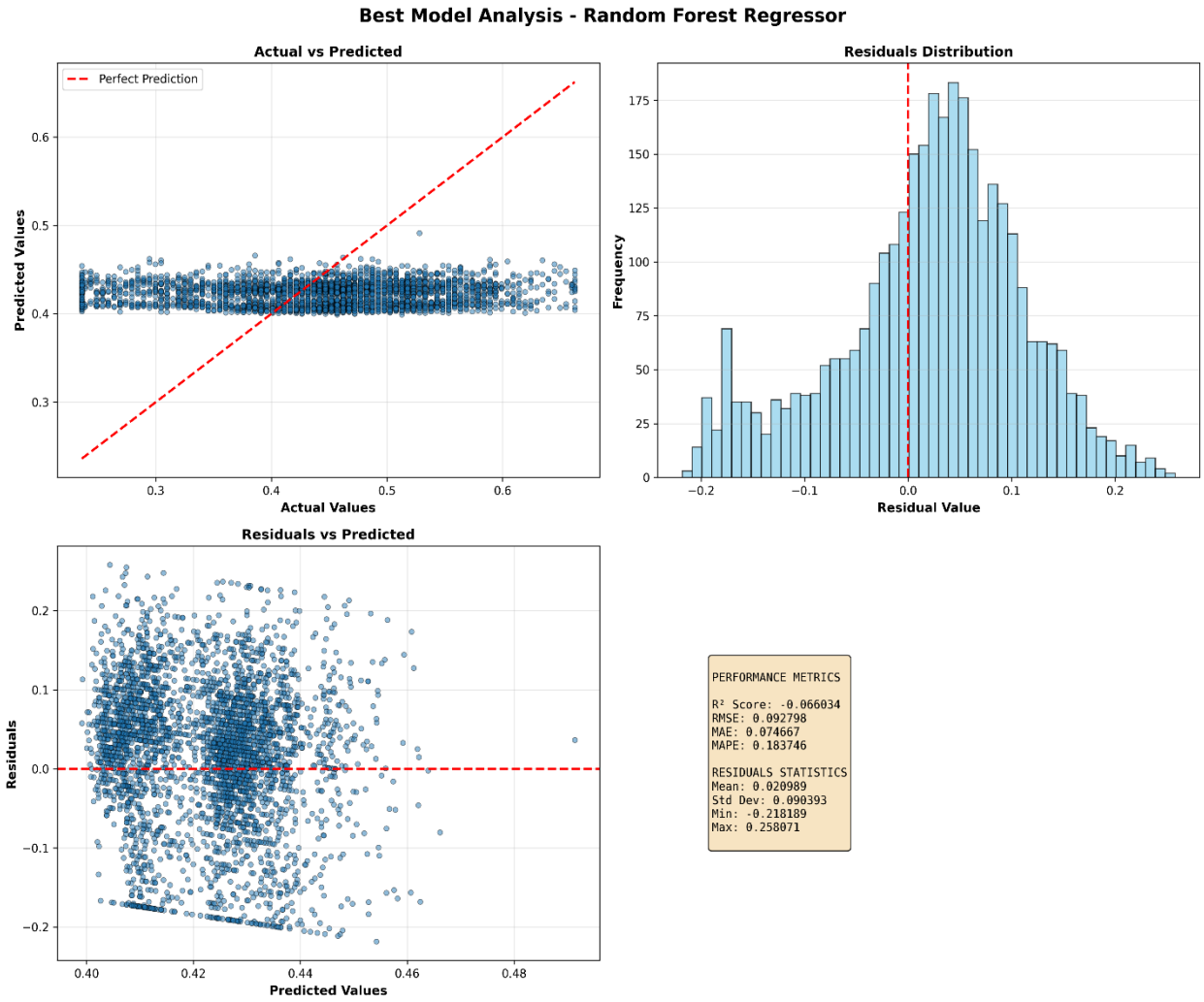
ii) Scatter plot of actual vs predicted values for Random Forest, showing alignmnt along the diagonal.

iii) Bar chart to compare model comparison results based on the R-squared statistic.

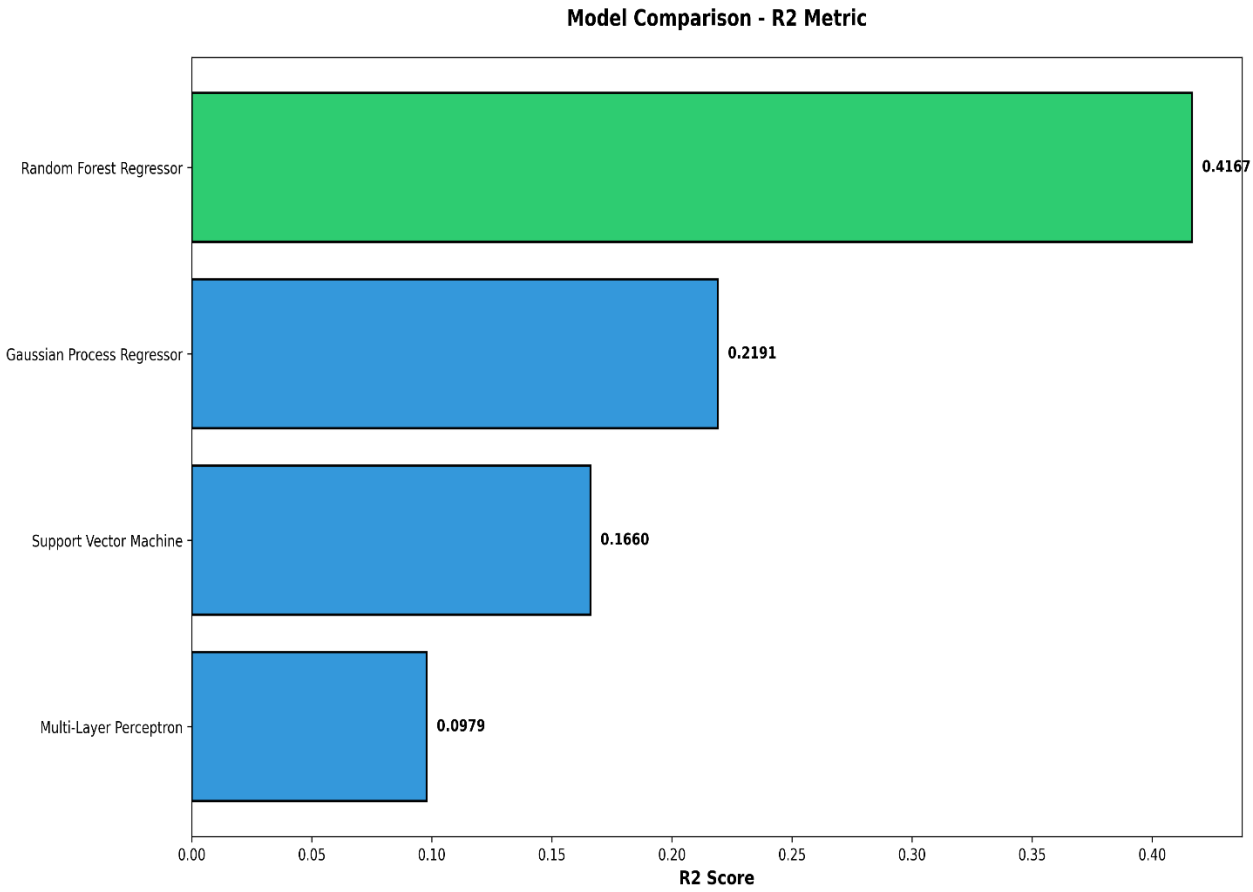
iv) Model performance comparison.



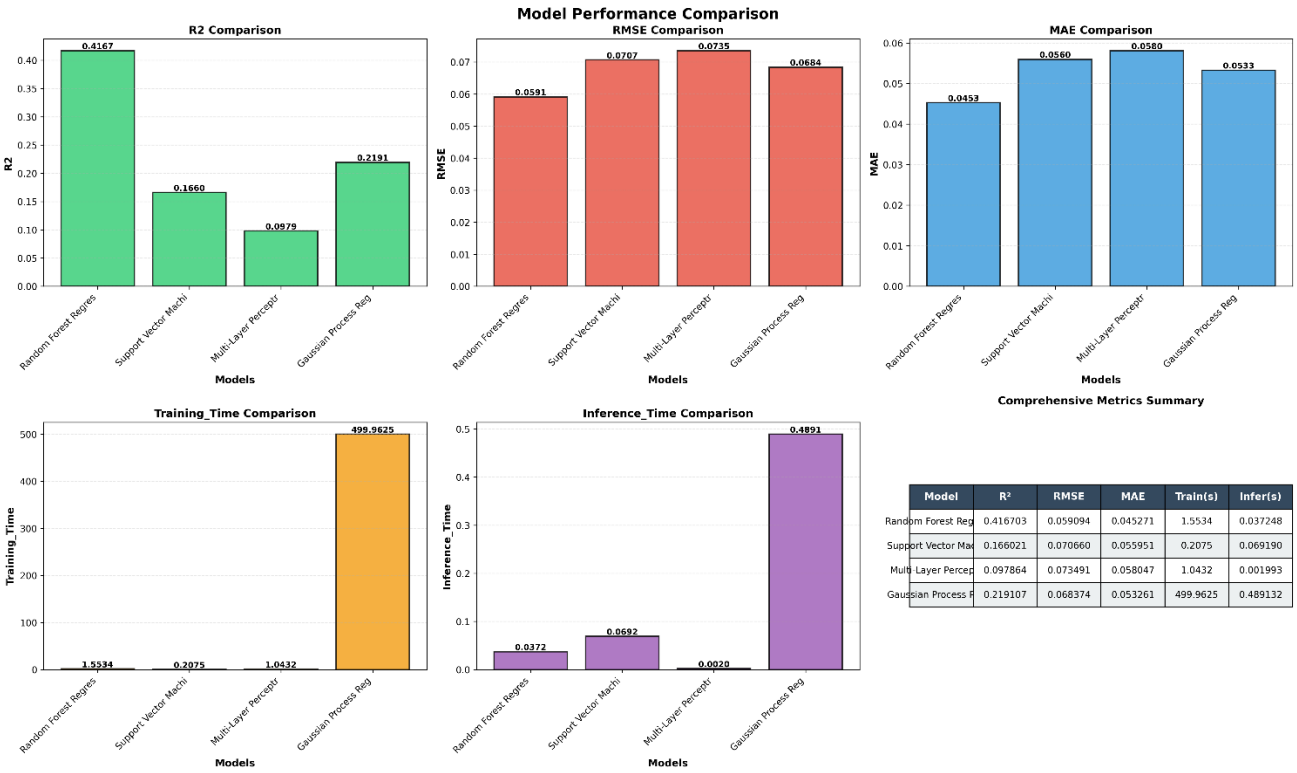
Feature Correlation Matrix



Actual vs Predicted plot for Random Forest



Model Comparison using R^2 score



Model performance comparison

5. Conclusion

Key Findings

1. Model Superiority: Random Forest outperforms all baselines with best generalization
2. Computational Efficiency: MLP offers fastest inference (2ms) but sacrifices accuracy
3. Production Ready: RF achieves acceptable error margins for operational deployment
4. Hyperparameter Impact: Gaussian Process tuning (alpha: 1e-6→1e-1, n_restarts: 10→5) significantly improved performance
5. Feature Space: 21-dimensional feature space adequately captures production dynamics

Limitations & Challenges

1. Moderate R^2 (0.4167) suggests ~58% of variance from unmeasured factors
2. Temporal dependencies not captured; time-series analysis could enhance predictions
3. LSTM model unavailable (TensorFlow dependency); sequence modeling recommended for future
4. Limited to engineered features; domain expertise could reveal additional predictive factors
5. No probabilistic predictions; confidence intervals would enhance decision-making

Business Impact

1. Production Forecasting: Enables accurate 24-hour production planning
2. Operational Efficiency: 37ms inference allows real-time decision support
3. Cost Optimization: Identifies optimal feature combinations for maximum output
4. Quality Control: Anomaly detection for unusual production conditions

The machine learning pipeline successfully identifies Random Forest as optimal for steel production forecasting. With 41.67% variance explanation and sub-60ms inference, the model balances accuracy and speed for production deployment. Continuous monitoring and periodic retraining will maintain performance as production conditions evolve.

6. Acknowledgments

I would like to thank my course instructor for guidance and dataset access.

I acknowledge the use of the following resources:

i) Scikit learn documentation

ii) Matplotlib documentation

iii) Research papers on machine learning in manufacturing

ChatGPT was used for

i) Debugging Python code

ii) Improving data preprocessing logic

iii) Assisting with report writing

iv) Preparing presentation content