Steel Production Data Analysis

Name: Haoyang Li

Student Number: m12520419

1 Abstract

This project presents a complete machine learning pipeline for predicting a quality-related output variable in steel production processes. The pipeline integrates systematic data preprocessing, feature normalization, model training, hyperparameter optimization, and comparative evaluation. Four regression models—Random Forest, Support Vector Regression (SVR), Multi-Layer Perceptron (MLP), and Gaussian Process Regression (GPR)—were implemented and evaluated under a unified experimental framework. Model selection was based on validation performance, and final evaluation was conducted on an independent test set using multiple statistical metrics. The results demonstrate the strengths and limitations of different regression paradigms in modeling nonlinear industrial process data and provide practical insights for deploying predictive models in steel manufacturing quality control systems.

2 Introduction

2.1 Project Overview

Steel production is a highly complex industrial process in which product quality is influenced by numerous interacting process parameters, such as temperature control, chemical composition, and operational conditions. These relationships are often nonlinear and difficult to model using traditional analytical or rule-based approaches. As production systems generate increasing volumes of data, machine learning methods have become powerful tools for extracting patterns and building predictive models for industrial quality control.

In the context of steel manufacturing, accurate prediction of quality-related indicators enables early detection of potential defects, supports process optimization, and contributes to cost reduction and production efficiency. Data-driven regression models, in particular, offer the ability to learn complex mappings between process variables and quality outcomes without requiring explicit physical modeling. This project situates itself within this context by applying modern machine learning techniques to historical steel production data.

2.2 Project Objectives

The primary objective of this project is to develop a complete and reproducible machine learning pipeline for predicting a steel quality indicator based on production process data. The project emphasizes not only predictive performance but also methodological rigor and engineering applicability.

The specific objectives are as follows:

(1) To perform systematic data preprocessing, including data cleaning, outlier handling, feature encoding, and normalization.

(2) To implement and train multiple regression models representing different modeling paradigms.

(3) To optimize model hyperparameters using cross-validation.

(4) To evaluate and compare model performance using standardized statistical metrics.

(5) To analyze model strengths, limitations, and suitability for industrial deployment.

(6) Through these objectives, the project aims to identify effective machine learning approaches for steel production quality prediction and to provide insights that are transferable to similar industrial data analysis problems.

3 Data Description

3.1 Dataset Characteristics

The dataset used in this project consists of historical steel production records, where each row represents an individual production instance and each column corresponds to a measured process parameter or operational variable. The target variable, referred to as output, represents a quantitative indicator of steel quality and serves as the prediction objective for all implemented models.

The dataset includes a mixture of continuous and categorical features that capture different aspects of the steel manufacturing process. As is typical for industrial datasets, the data exhibit several challenges, including potential missing values, varying feature scales, and the presence of extreme observations. These characteristics necessitate careful preprocessing to ensure that the data are suitable for machine learning model training and evaluation.

3.2 Data Preprocessing Steps

To improve data quality and ensure reliable model performance, a structured preprocessing pipeline was applied prior to model training. The key steps are summarized as follows:

(1) Duplicate Removal: Duplicate samples were identified and removed to prevent redundant information from biasing the learning process.

(2) Missing Value Handling: Missing values were addressed using statistical imputation methods, such as mean or median substitution, depending on feature distribution characteristics.

(3) Outlier Detection and Treatment: The Interquartile Range (IQR) method was employed to detect outliers. Extreme values were treated to reduce their influence on model fitting, particularly for models sensitive to large deviations.

(4) Categorical Feature Encoding: Categorical variables were converted into numerical representations to ensure compatibility with regression-based machine learning models.

(5) Dataset Splitting: The cleaned dataset was split into training, validation, and test subsets. This separation enables unbiased hyperparameter tuning and fair evaluation of model generalization performance.

(6) Feature Scaling: Feature normalization using standard scaling was applied to align feature magnitudes, which is especially important for distance-based and gradient-based models such as SVR, MLP, and GPR.

4 Methodology

4.1 Implemented Models

To comprehensively evaluate different regression paradigms, four machine learning models were implemented:

Random Forest Regressor (RF): An ensemble learning method that constructs multiple decision trees and aggregates their predictions. Random Forest is well known for its robustness to noise, ability to capture nonlinear feature interactions, and reduced risk of overfitting through bootstrap aggregation.

Support Vector Machine (SVR): A kernel-based regression model that seeks to find a function within a predefined error margin while maximizing model generalization. The use of kernel functions allows SVR to model nonlinear relationships in high-dimensional feature spaces.

Multi-Layer Perceptron (MLP): A feedforward neural network consisting of multiple fully connected layers. By applying nonlinear activation functions, MLP models can approximate complex nonlinear mappings between input features and the target variable.

Gaussian Process Regressor (GPR): A probabilistic regression model that defines a prior over functions and updates it using observed data. GPR provides flexible function approximation and can capture uncertainty in predictions, making it suitable for modeling complex industrial processes.

4.2 Hyperparameter Optimization Strategy

Each model contains hyperparameters that significantly influence its performance and generalization ability. To ensure a fair comparison among models, hyperparameter optimization was performed using Grid Search combined with k-fold cross-validation.

Specifically, a 5-fold cross-validation strategy was adopted on the training set. In this approach, the training data were divided into five subsets, and each subset was used once as a validation fold while the remaining subsets were used for training. The average performance across folds was used to evaluate each hyperparameter configuration. This strategy reduces the variance associated with a single train – validation split and leads to more robust parameter selection.

The optimization objective was the negative mean squared error, which penalizes large prediction errors and aligns well with regression accuracy requirements in industrial quality prediction tasks.

4.3 Training Procedure

Model training followed a structured, multi-stage procedure designed to avoid data leakage and ensure unbiased performance evaluation:

Initial Training Phase

Each model was trained on the training subset using all candidate hyperparameter combinations defined in the search space.

Model Selection Phase

The best-performing hyperparameter configuration for each model was selected based on cross-validation results. The selected model was then evaluated on a separate validation set to assess its generalization performance.

Final Training Phase

After model selection, the training and validation datasets were merged to form a larger training set. The selected model was retrained on this combined dataset to maximize the amount of information available before final testing.

Final Evaluation Phase

The final trained models were evaluated on an independent test set that was not used during any training or tuning step. This ensures that reported test results reflect true out-of-sample performance.

4.4 Computational Performance Measurement

In addition to predictive accuracy, computational efficiency was explicitly considered. For each model, both training time and inference time were measured:

Training Time reflects the computational cost required to fit the model and is particularly relevant for large-scale or frequently retrained systems.

Inference Time represents the time required to generate predictions and is critical for real-time or near-real-time industrial deployment.

By reporting both accuracy metrics and computational metrics, the evaluation provides a more comprehensive basis for selecting models suitable for practical steel production environments.

5 Results

5.1 Learning Curve Analysis

Learning curves were employed to analyze the relationship between model performance and training data size for four regression models: Random Forest, Support Vector Regression (SVR), Multi-Layer Perceptron (MLP), and Gaussian Process Regression (GPR).

The Random Forest model achieved the lowest training RMSE across all training sizes, indicating a strong ability to fit the training data. However, a persistent gap between training and validation errors suggests mild overfitting, with limited improvement in validation performance as the dataset size increases.

The MLP model exhibited a very low training error but significantly higher validation error, which is characteristic of high-variance behavior. Although increasing the training set size reduced the validation error, a noticeable gap remained, indicating that the model tends to overfit under the current configuration.

The SVR model showed moderate training and validation errors with a relatively stable gap between them. While the validation RMSE decreased gradually as more data became available, the model did not fully converge, suggesting limited model capacity or suboptimal kernel parameterization.

In contrast, the GPR model demonstrated the most balanced learning behavior. The training and validation RMSE values were consistently close, and the validation error steadily decreased with increasing training samples. This indicates a favorable bias–variance trade-off and strong generalization capability.

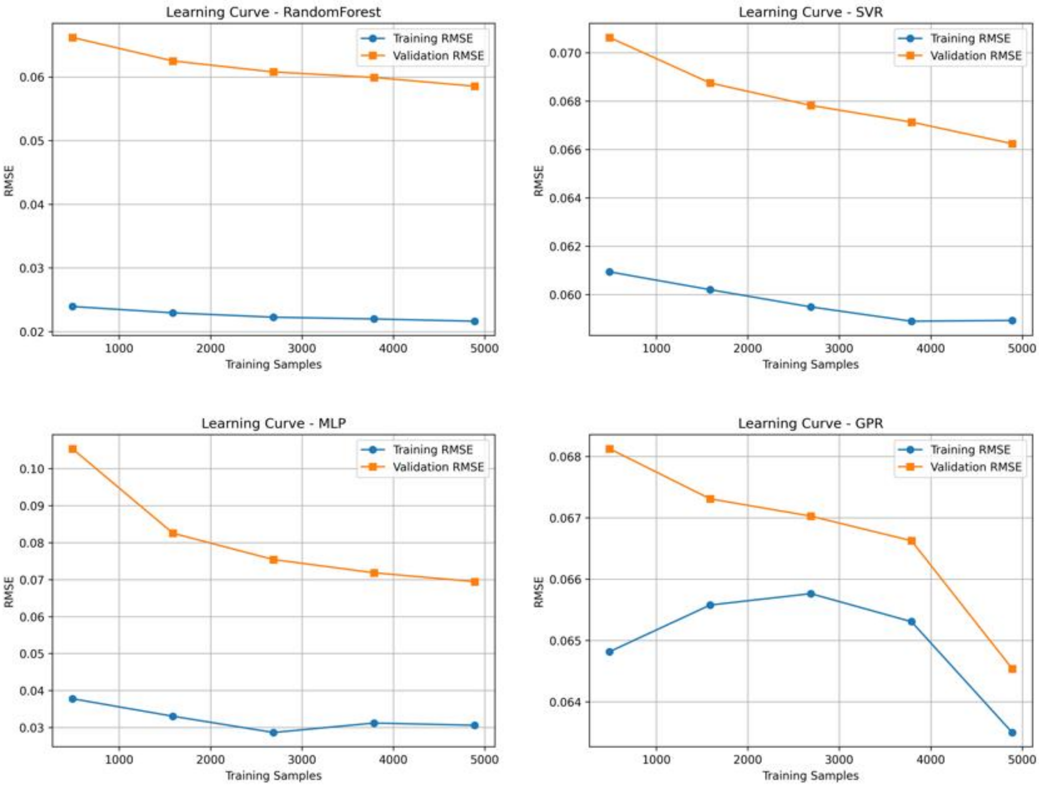

Figure 1 Learning Curve

5.2 Predicted vs Actual Comparison

Scatter plots of predicted versus actual values were used to evaluate the predictive accuracy and trend-capturing ability of each model.

The Random Forest and GPR models produced predictions that closely followed the ideal diagonal line, indicating strong agreement between predicted and true values. Random Forest predictions were particularly accurate within the central range of the target variable, although some discretization effects were observed due to the tree-based structure.

The SVR and MLP models displayed more dispersed point clouds. In the case of SVR, predictions tended to be compressed toward the mean, especially at extreme values, reflecting the smoothing effect of the kernel function. The MLP model captured nonlinear trends but showed increased variability and less stable predictions across the output range.

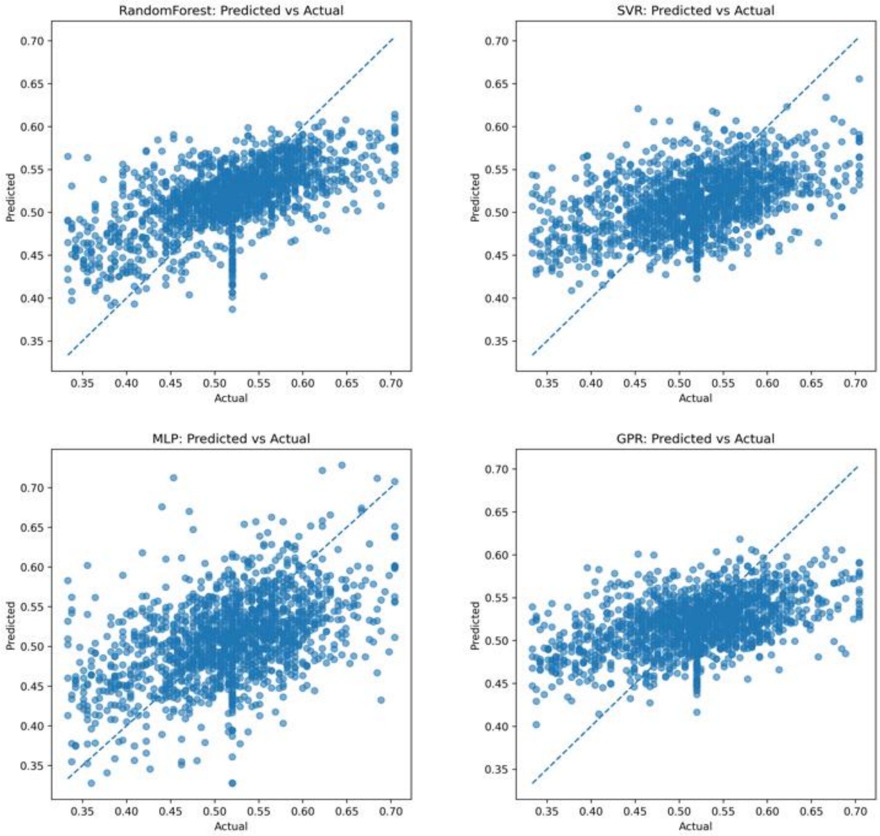

Figure 2 Predicted vs Actual

5.3 Residual Analysis

Residual plots were analyzed to detect systematic prediction errors and assess model stability.

The GPR residuals were symmetrically distributed around zero with no evident structure or trend, indicating that the model errors were largely random and that the model assumptions were well aligned with the data distribution.

The Random Forest residuals were generally concentrated around zero but exhibited localized patterns, suggesting sensitivity to certain feature regions and mild systematic deviations.

The MLP and SVR residual plots showed wider dispersion and signs of heteroscedasticity, indicating inconsistent prediction accuracy across different output ranges. These observations further support the presence of overfitting in MLP and limited generalization in SVR.

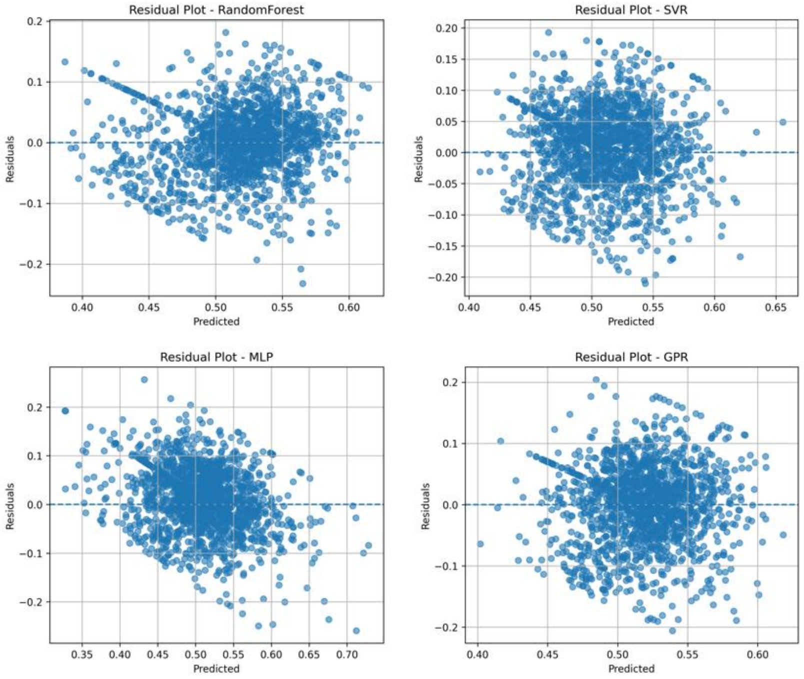

Figure 3 Residual Plot

5.4 Model Comparison with Error Bars

A bar chart with error bars was used to compare model performance in terms of cross-validated RMSE mean and standard deviation.

The Random Forest model achieved the lowest average RMSE, demonstrating strong predictive accuracy. However, its error bars were relatively larger, indicating moderate variability across cross-validation folds.

The GPR model achieved slightly higher RMSE than Random Forest but exhibited the smallest standard deviation, highlighting superior stability and robustness.

Both SVR and MLP models showed higher average RMSE values and larger variability, suggesting inferior overall performance under the current settings.

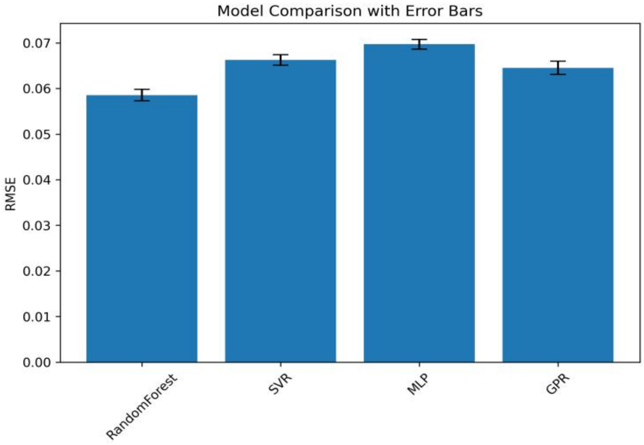

Figure 4 Model Comparison

5.5 Performance Table Analysis

Table 1 Model performance on the test set

Model	RMSE	MAE	R^2	Train_Time (s)	Inference_Time (s)
Random Forest	0.0576	0.0446	0.3784	11.6479	0.1040
SVR	0.0652	0.0520	0.2037	0.2478	0.07090
MLP	0.0678	0.0533	0.1399	4.4893	0.0040
Gaussian Process	0.0644	0.0508	0.2229	69.8868	0.1554

The Random Forest model achieved the best overall predictive performance, as evidenced by the lowest RMSE and MAE and the highest R^2 score. Gaussian Process Regression demonstrated strong generalization stability but suffered from high computational cost. Support Vector Regression provided a good balance between accuracy and efficiency, while the MLP model offered fast inference but comparatively weaker predictive performance. The results highlight the importance of balancing accuracy and computational efficiency when selecting a model for practical applications.

6 Discussion

This study systematically compared four regression models—Random Forest, Support Vector Regression (SVR), Multi-Layer Perceptron (MLP), and Gaussian Process Regression (GPR)—for predicting a quality-related output in steel production processes. The discussion focuses on interpreting the observed performance differences, linking quantitative results to model characteristics, and evaluating their suitability for industrial applications.

From a predictive accuracy perspective, the Random Forest model achieved the lowest RMSE and MAE values and the highest R^2 score on the test set. This result highlights the effectiveness of ensemble tree-based methods in capturing nonlinear relationships and complex feature interactions commonly present in industrial process data. However, learning curve and residual analyses revealed a persistent gap between training and validation errors, indicating mild overfitting. While this does not severely impact test performance in the current dataset, it suggests that Random Forest models may require careful tuning or regularization when deployed in evolving production environments.

Gaussian Process Regression demonstrated strong generalization behavior, as evidenced by closely aligned training and validation errors and well-behaved residual distributions. The probabilistic nature of GPR allows it to model smooth nonlinear functions and implicitly quantify uncertainty, which is highly valuable in industrial quality monitoring. Nevertheless, the extremely high training and inference times observed for GPR significantly limit its scalability. As dataset size increases, the cubic computational complexity of GPR becomes a critical bottleneck, making it less suitable for large-scale or real-time industrial deployment without approximation techniques.

Support Vector Regression showed moderate predictive performance with relatively stable training and validation errors. The learning curves indicate that SVR continues to benefit from additional data but converges slowly, suggesting limited model capacity under the chosen kernel and hyperparameter configuration. While SVR offers advantages in terms of computational efficiency and robustness, its tendency to compress predictions toward the mean reduces accuracy for extreme quality values, which may be critical in defect detection scenarios.

The Multi-Layer Perceptron model exhibited the lowest inference time, making it attractive for real-time prediction tasks. However, its comparatively high RMSE and MAE values, along with wide residual dispersion, indicate overfitting and unstable generalization. This behavior suggests that the current network architecture and training configuration are insufficient to fully exploit the available data. With more extensive hyperparameter tuning, regularization strategies, or larger datasets, neural networks may achieve improved performance, but such enhancements were beyond the scope of this project.

Overall, the results illustrate that no single model is optimal across all criteria. Instead, model selection involves a trade-off between predictive accuracy, generalization stability, and computational efficiency. These trade-offs are particularly important in industrial contexts, where deployment constraints such as training cost, prediction latency, and system scalability must be considered alongside accuracy.

7 Conclusion

This project developed and evaluated a complete machine learning pipeline for predicting a steel production quality indicator using historical process data. Through systematic data preprocessing, model training, hyperparameter optimization, and comprehensive evaluation, the study compared four widely used regression models within a unified experimental framework.

The results demonstrate that Random Forest achieved the highest predictive accuracy, making it a strong candidate for offline quality prediction and process optimization tasks. Gaussian Process Regression exhibited excellent generalization stability and well-structured residual behavior but suffered from prohibitively high computational cost. Support Vector Regression provided a reasonable balance between accuracy and efficiency, while the Multi-Layer Perceptron offered fast inference but showed limited predictive performance under the current configuration.

These findings highlight the importance of aligning model choice with application requirements. For scenarios prioritizing prediction accuracy, ensemble methods such as Random Forest are highly effective. In contrast, applications requiring uncertainty awareness may benefit from Gaussian Process models, provided computational constraints are addressed. For real-time systems, lightweight models with fast inference, such as neural networks, may be preferable after further optimization.