Electric Motor Temperature Prediction using Machine Learning

Valentin Ruillon et Luca Sormani

December 23, 2025

Abstract

This report presents a complete Machine Learning workflow designed to estimate the internal Permanent Magnet (PM) temperature of a Permanent Magnet Synchronous Motor (PMSM). Using a large experimental dataset (2 Hz sampling, $\sim$ 1.3 million samples), we perform exploratory analysis, preprocessing, dimensionality reduction, and model comparison. Linear, regularized, and ensemble learning models are evaluated under strict time-series constraints to avoid data leakage. The study highlights the domain-specific behavior of thermal variables and the unexpected superiority of regularized linear models over tree-based ensembles when generalizing to unseen driving profiles.

Contents

1 Business Case

1.1 Objective

The goal of this project is to predict the temperature of the Permanent Magnet (PM) inside an electric motor using non-invasive sensor measurements such as stator temperatures, electrical variables (i_d, i_q, u_d, u_q), and rotational speed. Since direct measurement of rotor temperature is technically challenging and costly, a virtual sensor based on data-driven modelling offers a cost-effective and reliable alternative.

1.2 Industrial Relevance

Monitoring rotor temperature is critical in electric vehicles and industrial drives. Excessive PM temperature may cause partial demagnetization, leading to irreversible torque loss. Conventional temperature sensors cannot be mounted on the rotor without mechanical complexity and reliability issues. A Machine Learning-based virtual sensor can:

- reduce hardware costs,
- improve reliability by removing vulnerable sensors,
- enable real-time estimation in embedded systems.

2 Dataset Description

2.1 Source

The dataset used in this work is the *Electric Motor Temperature* dataset made available by Paderborn University and published on Kaggle:
`https://www.kaggle.com/datasets/wkirgsn/electric-motor-temperature`

2.2 Structure of the Dataset

The dataset consists of approximately $\sim$ 1.3 million samples collected from multiple driving cycles with a sampling frequency of 2 Hz. Each driving cycle is identified by a unique identifier `profile_id`. The main variables include:

- **Ambient and coolant temperatures**,
- **Electrical inputs**: u_d, u_q, i_d, i_q,
- **Mechanical inputs**: motor speed,
- **Stator temperatures**: yoke, tooth, and winding,
- **Target**: `pm` (Permanent Magnet temperature).

Table 1 summarizes basic dataset statistics.

Statistic	Value
Number of samples	$\sim$ 1,300,000
Number of features	12
Number of driving profiles (`profile_id`)	290
Sampling frequency	2 Hz
Missing values	0 (Dataset is complete)

Table 1: Summary of dataset characteristics.

3 Data Exploration

3.1 Correlation Analysis

A correlation matrix was computed to identify linear dependencies between variables. The stator temperatures (winding, tooth, and yoke) show strong positive correlation with the PM temperature, which is consistent with known thermal diffusion mechanisms in PMSMs.

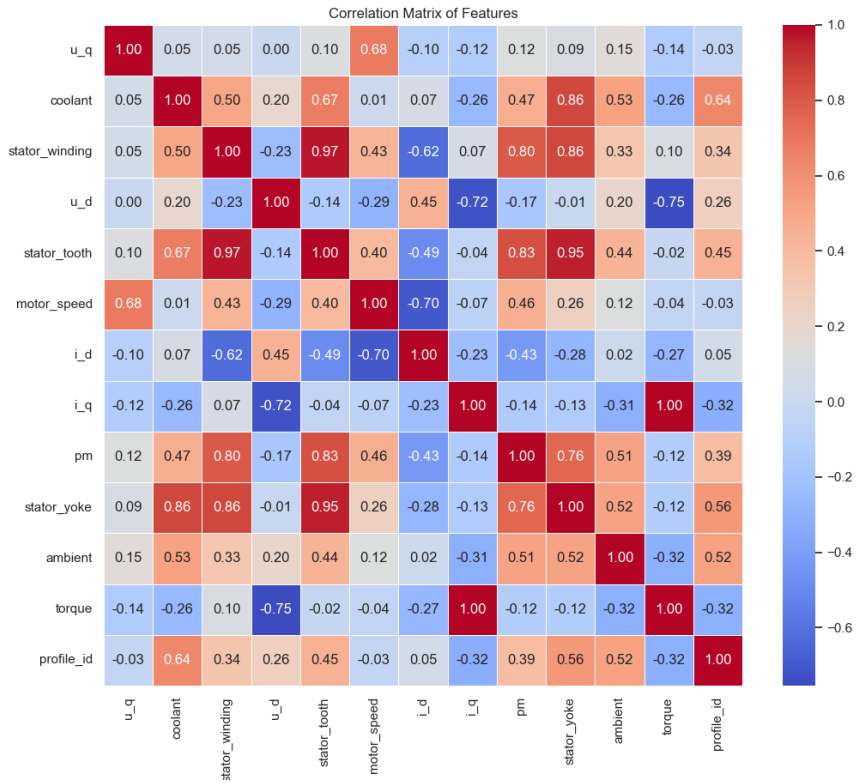

Figure 1: Correlation heatmap of all features.

3.2 Target Distribution

The PM temperature distribution exhibits multiple modes, each corresponding to different operating profiles or thermal regimes.

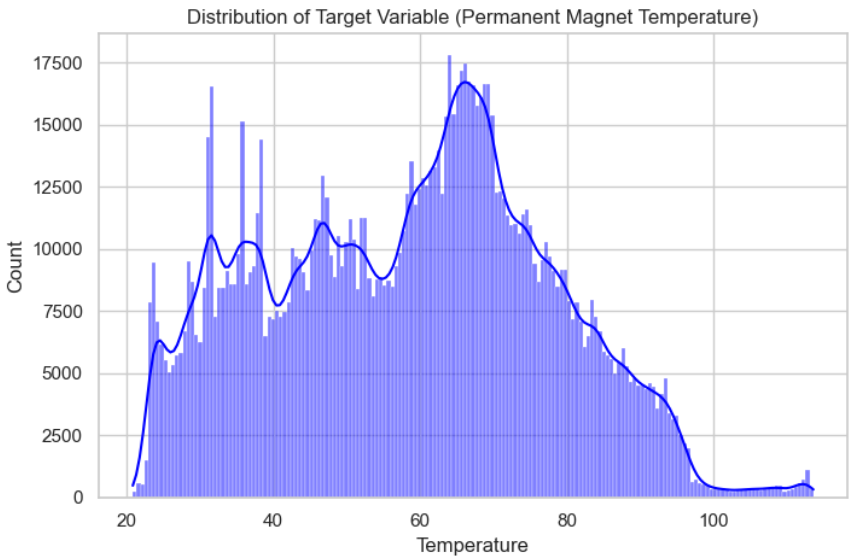

Figure 2: Distribution of the Permanent Magnet temperature.

4 Formalization of the Predictive Problem

The task is formalized as a **supervised regression** problem. The model learns a function:

$$\hat{y} = f(X), \tag{1}$$

that maps sensor readings X to the continuous PM temperature y.

The main performance metric is the Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{2}$$

which penalizes large errors. RMSE is emphasized here because large temperature estimation errors may create operational risk (e.g., overheating or demagnetization).

5 Obstacles and Mitigation Strategies

5.1 Avoiding Data Leakage in Time-Series Data

A standard random train-test split would incorrectly mix samples from the same driving cycle in both sets, creating a strong data leakage effect. To prevent this, the split was performed using `GroupShuffleSplit` on the `profile_id`, ensuring that each driving session appears exclusively in either training or testing.

5.1.1 Train/Test Split Details (Group-Aware)

The dataset was split at the profile level using a group-aware strategy with an 80/20 ratio (train/test). In our run, this corresponds to **55 training profiles** and **14 testing profiles** (total: 69 profiles). Due to unequal profile lengths, the effective test set represents approximately 22.9% of the total samples.

5.2 Hardware Constraints and Dataset Scale

The dataset contains ~ 1.3 million rows, which imposes memory constraints during training of tree-based ensemble models. To avoid memory overflow:

- all numerical columns were cast to `float32`,
- Random Forest depth was limited to a safe range (e.g., `max_depth=15`),
- XGBoost was configured with `tree_method=hist` and early stopping.

5.3 Overfitting and Underfitting

Linear models were regularized using Ridge Regression. Learning curves and residual analysis were used to diagnose remaining overfitting.

6 Dimensionality Reduction

Principal Component Analysis (PCA) was applied to explore latent structure and reduce redundancy. PCA was used solely as an exploratory tool (variance structure and redundancy); it was not used in the final predictive pipeline because the full feature set yielded better test performance.

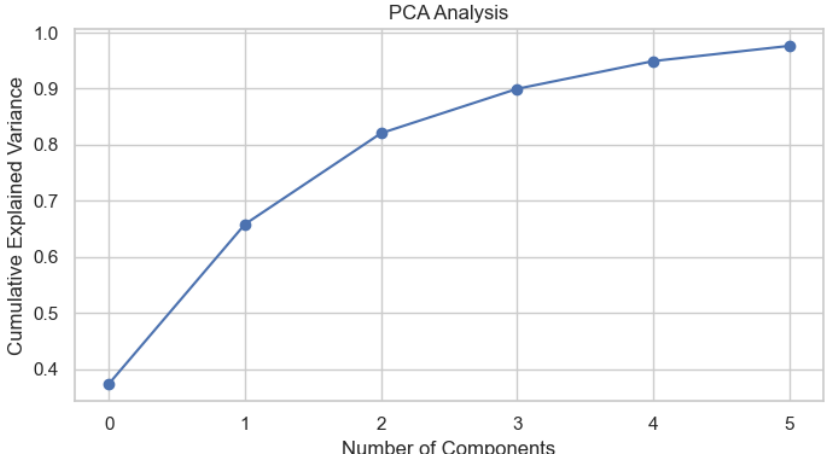

Figure 3: Explained variance per PCA component.

7 Modeling Approach

The following models were evaluated:

1. **Naïve Mean Predictor** (context baseline): predicts the mean PM temperature from the training set,
2. **Linear Regression**,
3. **Ridge Regression** with hyperparameter tuning over α,
4. **Random Forest Regressor**,
5. **XGBoost Regressor**.

All hyperparameter searches were performed using group-aware cross-validation to prevent leakage across driving profiles.

8 Model Comparison

Model	RMSE	MAE	R^2
Naïve Mean Baseline	19.42	16.43	-0.005
Ridge Regression	**6.43**	4.93	0.89
XGBoost	7.71	5.85	0.84
Random Forest	8.63	6.17	0.80

Table 2: Comparison of model performance on the test set.

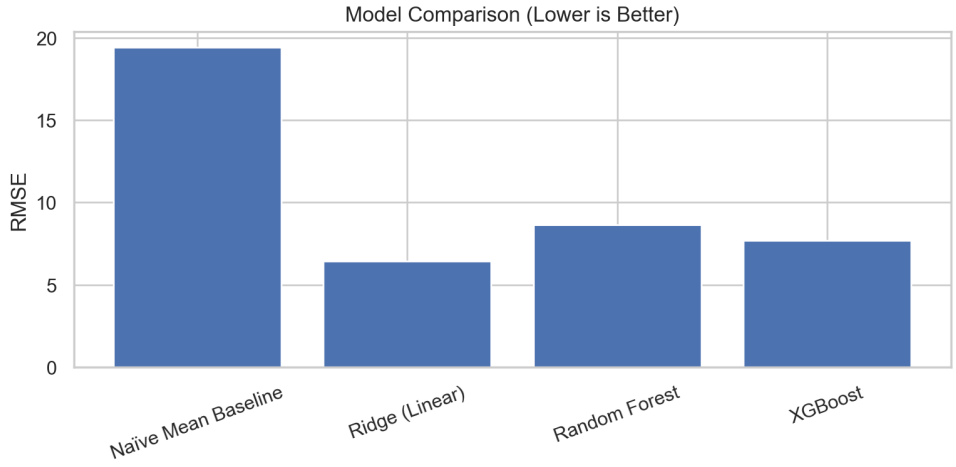

Figure 4: RMSE of all evaluated models (including the naïve baseline).

9 Conclusion

The project demonstrates that a relatively simple regularized linear model (Ridge Regression) provides the best generalization performance when predicting PM temperature from stator and electrical features.

Ridge generalizes better in this context because the underlying thermal physics are mostly linear in steady-state operation. Conversely, tree-based models (XGBoost, Random Forest) may struggle to extrapolate outside the distribution of the training profiles, leading to higher errors on unseen driving cycles.

The final model provides an efficient, interpretable, and computationally light virtual sensor suitable for deployment in embedded systems.

10 Reproducibility Notes

10.1 Software Versions

The following versions were used in this project:

- Python 3.x

- NumPy 1.26.4
- Pandas 2.2.2
- Scikit-Learn 1.5.1
- XGBoost 3.1.2

10.2 Environment Setup

To reproduce the environment without providing a `requirements.txt`, install the core dependencies directly:

```
pip install numpy==1.26.4 pandas==2.2.2 scikit-learn==1.5.1 xgboost==3.1.2 matplotlib
```

All experiments were executed with a fixed random seed to ensure stability:

```
np.random.seed(42)
```

11 References

- Kirchgässner, W., Wallscheid, O., & Böcker, J. (2021). *Deep Residual Convolutional and Recurrent Neural Networks for Temperature Estimation in Permanent Magnet Synchronous Motors.*
- Kaggle Dataset: `https://www.kaggle.com/datasets/wkirgsn/electric-motor-temperature`
- Scikit-Learn Documentation.