**AI-ML Internship**  **SmartBridge**

**NAME : Jaimin Deepsinh Solanki**

**Collage : Government Engineering Collage, Sector 28**

**Title:**

Design of an Artificial Intelligence-Based Model for Electric Motor Temperature Prediction with Machine Learning

**Introduction:**

Permanent Magnet Synchronous Machines (PMSMs) are a part of a broad range of motion control applications. Robotics, machine tools, electric vehicles, actuators, and home appliances are some examples. Low torque ripple, high power density, good dynamic performance, and energy efficiency are the benefits offered by PMSMs. Due to these benefits, PMSMs are gaining popularity not only in industry but also in consumer markets.

Yet proper monitoring of the rotor temperature is imperative. Overheating can impair performance, shorten motor life, or lead to total system failure. Conventional sensors are often not real-time, cost-effective solutions. Machine learning can provide a strong, non-invasive option by extrapolating rotor temperature through measurable operating parameters.

This project is a predictive model of rotor temperature with various machine learning algorithms to implement the top-performing model as a web application via Flask.

**Problem Statement:**

Electric motor component real-time monitoring, particularly rotor temperature, is essential for performance and safety. However:

Physical sensors on rotors are expensive and invasive to install.

Indirect parameters (such as ambient temperature, stator winding temperature, torque, current) are available and can be utilized to predict rotor temperature.

There is not a globally deployed intelligent system that accurately estimates rotor temperature based on data.

Therefore, our intent is to design an AI model that could effectively estimate the rotor temperature from other available features with less reliance on physical sensors.

**Objective:**

The major objective of this project is to create and deploy a machine learning-based model that precisely predicts rotor temperature in PMSMs. Some of the specific objectives are:

Perform feature engineering to extract and transform relevant variables for training models.

Train and compare several machine learning algorithms:

Linear Regression

Decision Tree

Random Forest

Support Vector Machine (SVM)

Measure model performance with appropriate metrics such as (MAE, MSE, R² Score).

Use the best model and save it in .pkl format for further deployment.

Create a simple Flask-based web application that computes the rotor temperature given the user's input parameters.

**Why This Problem?**

With more PMSMs being integrated in applications of high demand, temperature monitoring becomes vital. High rotor temperature results in:

Reduced efficiency in performance

Potential insulation breakdown

Short motor life

Higher maintenance expenses

Machine learning can provide predictive models that avoid the use of direct rotor temperature sensors, allowing for low-cost, scalable, and quick solutions across industries. These smart systems facilitate preventive maintenance, enhancing dependability and operating safety.

**Solution:**

**Overview**:

The answer is a machine learning regression model that was trained on PMSM operating data. The model accepts several indirect input features (e.g., stator temperature, ambient temperature, torque) and outputs the rotor temperature.

The best model will be stored and incorporated into an easy-to-use Flask web interface for real-world application.

**Modeling Pipeline:**

**Data Preprocessing:**

Missing values treatment

Feature scaling (StandardScaler/MinMaxScaler)

Correlation analysis and feature selection

**Feature Engineering:**

Polynomial features (optional)

Temperature differential metrics

Interaction terms (e.g., torque × ambient temp)

**Model Training:**

Algorithms employed:

Linear Regression: to enable baseline comparison

Decision Tree: to identify non-linear patterns

Random Forest: for better generalization

Support Vector Regression (SVR): for accuracy in high-dimensional spaces

**Model Evaluation:**

R² Score (Coefficient of determination)

Mean Absolute Error (MAE)

Mean Squared Error (MSE)

Plot of predicted vs. actual temperature

**Model Saving:**

Save best model using joblib or pickle in .pkl format

**Flask Integration:**

Create a web app using Flask

Get user inputs for features

Load .pkl model and show predicted rotor temperature

**Technical Stack:**

|  |  |
| --- | --- |
| **Component** | **Tools/Frameworks** |
| Data Preprocessing | Python, Pandas, NumPy |
| Modeling | Scikit-learn |
| Visualization | Matplotlib, Seaborn |
| Model Deployment | Flask, HTML/CSS (for frontend) |
| Model Serialization | Pickle |

**Evaluation Metrics:**

R² Score: Indicates how accurately future samples are expected to be predicted.

MAE: Average of absolute prediction errors.

MSE: Average of squared prediction errors (punishes large errors).

Training vs Test Score Comparison: Prevents overfitting.

**Sample Output & Web Integration (Screenshots Placeholder):**

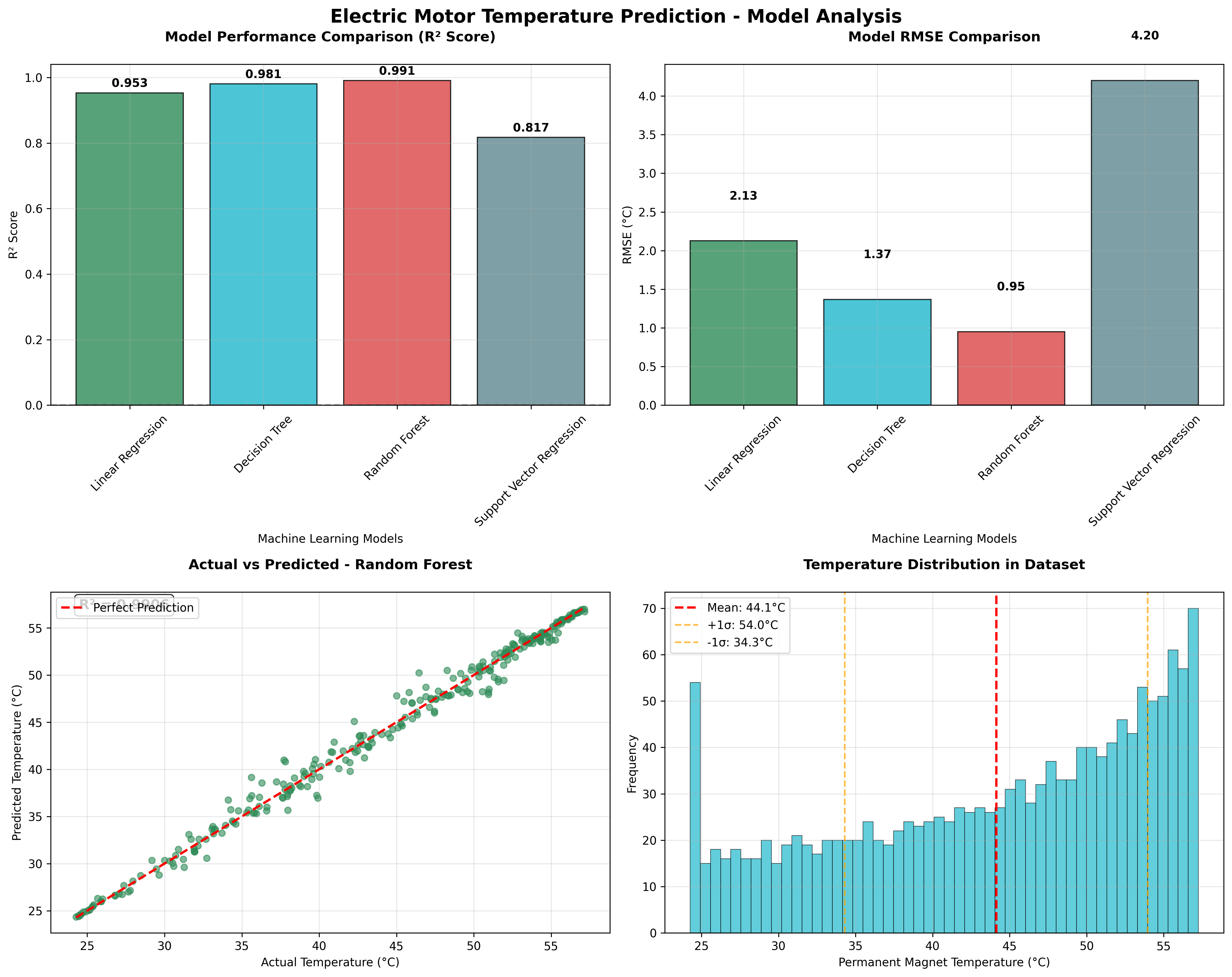


Fig 1.1: Plot comparing predicted and actual rotor temperature.

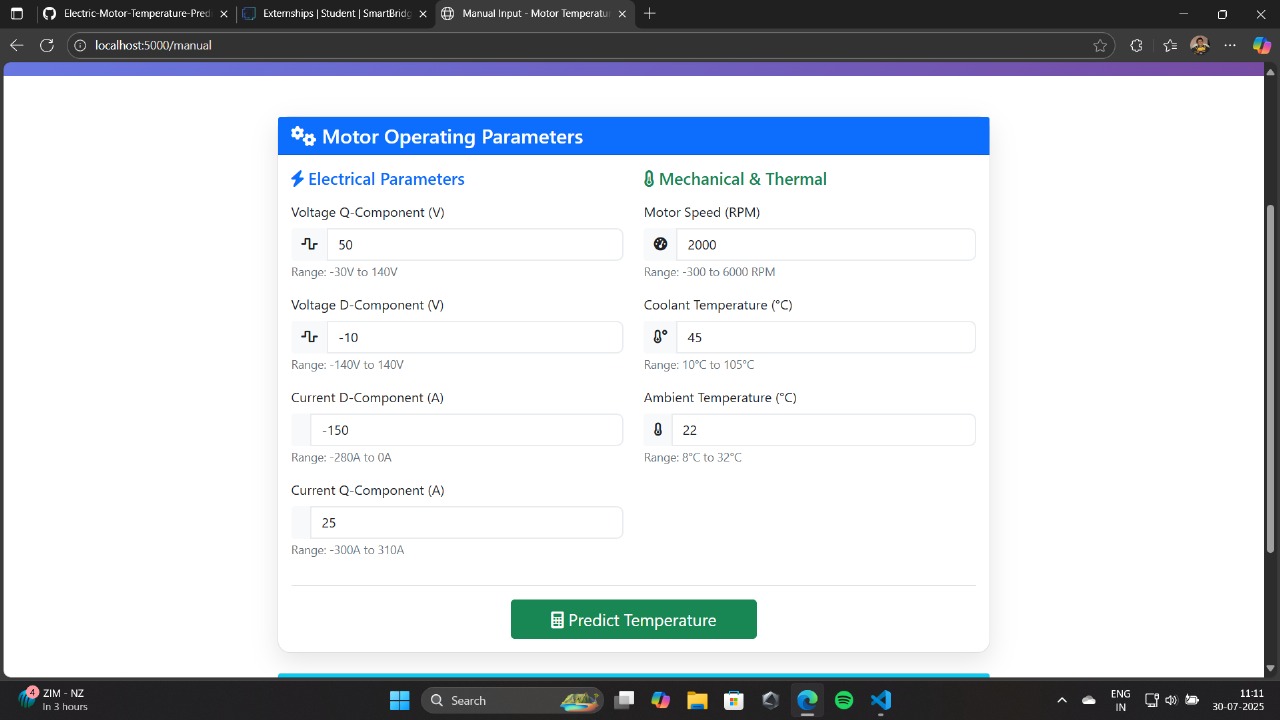


Fig 1.2: Input form of Flask web interface and output prediction result.

**Conclusion:**

The project demonstrates a real-world, data-driven approach to machine learning-based prediction of electric motor rotor temperature. It eliminates dependency on costly or difficult-to-maintain sensors with the aid of accessible operational features.

The system not only evaluates various ML algorithms based on performance but also provides an implementable tool through Flask for real-world deployment. Future developments can be:

Real-time IoT data integration

Deep learning techniques (e.g., LSTM for forecasting temperature in time-series)

Integration into motor control systems

Applying intelligent systems to motor diagnosis, this project enables smart maintenance, energy efficiency, and system reliability in industrial and commercial environments.