

**UFOP - Universidade Federal de Ouro Preto**

Decom - Departamento de Ciência da Computação

# **Modelos Básicos para IPMSMs**

Autor: Bianca Barreto Leme

Matrícula: 24.1.4008

Professor: Rodrigo César Pedrosa Silva

Ouro Preto - MG  
4 de setembro de 2025

# Sumário

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introdução</b>   | <b>2</b> |
| <b>2</b> | <b>Fundamentos</b>  | <b>2</b> |
| 2.1      | Bases de dados . . . . .  | 2        |
| 2.2      | Motores analisados . . . . .  | 3        |
| 2.3      | Métricas de Avaliação de Modelos de Regressão . . . . .   | 4        |
| 2.3.1    | Mean Absolute Percentage Error (MAPE) . . . . .   | 4        |
| 2.3.2    | Mean Squared Error (MSE) . . . . .  | 4        |
| 2.3.3    | Coeficiente de Determinação ( $R^2$ ) . . . . .   | 4        |
| 2.4      | Modelos Utilizados . . . . .  | 5        |
| 2.4.1    | Regressão Linear . . . . .  | 5        |
| 2.4.2    | Regression Trees . . . . .  | 5        |
| 2.4.3    | Random Forests . . . . .  | 5        |
| 2.4.4    | XGBoost . . . . .   | 5        |
| 2.4.5    | CatBoost . . . . .  | 6        |
| <b>3</b> | <b>Metodologia</b>  | <b>7</b> |
| 3.1      | Experimento 1: Identificar os Modelos mais Promissores . . . . .  | 7        |
| 3.2      | Experimento 2: Identificar o melhor conjunto de hiper parâmetros para os modelos mais promissores . . . . . | 7        |
| <b>4</b> | <b>Resultados</b>   | <b>7</b> |
| 4.1      | Resultados em MAPE . . . . .  | 7        |
| 4.2      | Resultados em MAPE . . . . .  | 7        |
| 4.3      | Resultados em $R^2$ Score . . . . .   | 7        |
| <b>5</b> | <b>Conclusão</b>  | <b>7</b> |

# 1 Introdução

Os Motores Síncronos de Ímã Interno Permanente (IPMSMs) têm sido amplamente utilizados por serem uma alternativa menos agressiva ao meio ambiente, quando comparados aos motores de carros comuns.

Em fase de testes (Finite Element Analysis), os motores devem ser submetidos a várias condições de velocidade e torque. Fazer estes testes fisicamente é custoso e pode levar dias. Por essa razão, utilizar “motores virtuais” e prever suas perdas através de modelos de IA pode ser muito mais viável, pois este método não tem grandes custos e levam por volta de algumas horas.

Estudando IPMSMs podemos criar modelos de inteligência artificial que nos auxiliem a prever as principais causas de perdas em ferro de determinado motor: perda por histerese e perda por eddy current.

Nesse contexto, este estudo tem como objetivo principal encontrar o melhor modelo de IA possível para prever as perdas em histerese e eddy current de 3 motores IPMSMs analisados: 2D, Nabla e V.

## 2 Fundamentos

Nesta seção do trabalho, serão descritos os métodos, algoritmos, procedimentos e métricas utilizadas para atingir o objetivo.

### 2.1 Bases de dados

As bases de dados consistem em estados de cada parâmetro dos motores em diferentes condições, e se dividem em duas partes: a primeira destinada para o treinamento dos modelos (train) e a segunda para a testagem de suas acurácias (test).

Destes datasets, os parâmetros se categorizam em outros dois grupos: Features (X) – variáveis preditoras ou independentes – e Targets (y) – variáveis alvo ou dependentes.

- Features:
  - Variáveis geométricas (Xgeom);
  - Velocidade do motor (N);
  - Corrente no eixo direto (Id);
  - Corrente no eixo em quadratura (Iq);
- Targets:
  - Perda por histerese (hysteresis);
  - Perda por eddy current (joule).

Em resumo, a base de dados se separa em 4:

- X train: features para treino;
- y train: targets para treino;
- X test: features para teste;
- y test: targets para teste.

Esses termos serão utilizados ao longo do artigo para melhor comunicação.

## 2.2 Motores analisados

Neste trabalho, foram analisados 3 IPMSMs distintos:

- 2D

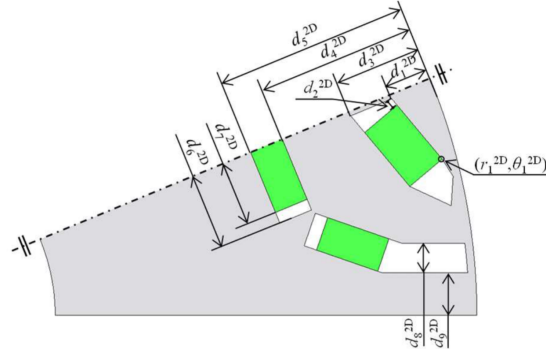


Figura 1: Diagrama do motor 2D.

- Nabla

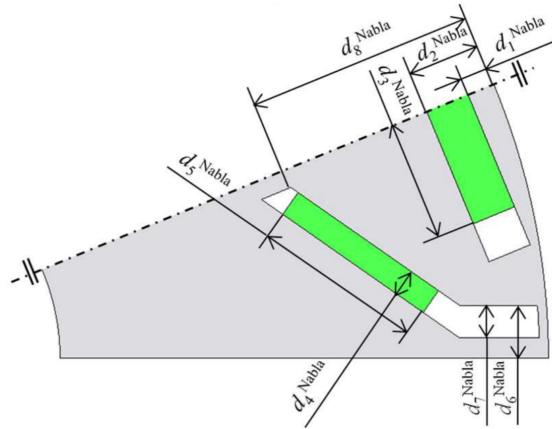


Figura 2: Diagrama do motor Nabla.

- V

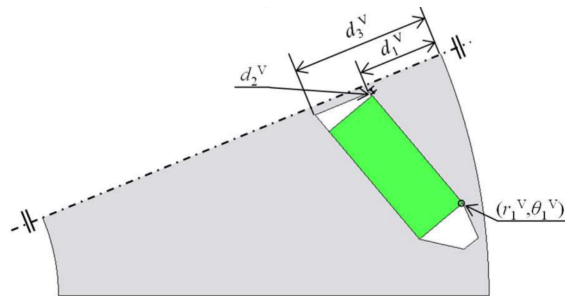


Figura 3: Diagrama do motor V.

## 2.3 Métricas de Avaliação de Modelos de Regressão

Para a análise da eficácia de cada modelo na predição dos atributos, foram definidas três métricas...

### 2.3.1 Mean Absolute Percentage Error (MAPE)

O *Mean Absolute Percentage Error* (MAPE) mede o erro percentual médio entre os valores reais  $y_i$  e os valores previstos  $\hat{y}_i$ :

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

### 2.3.2 Mean Squared Error (MSE)

O *Mean Squared Error* (MSE) mede o erro quadrático médio, penalizando mais fortemente desvios grandes:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

### 2.3.3 Coeficiente de Determinação ( $R^2$ )

O coeficiente de determinação, ou  $R^2$ , mede a proporção da variância dos dados reais que é explicada pelo modelo:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

onde  $\bar{y}$  é a média dos valores reais.

## 2.4 Modelos Utilizados

Nesta seção, serão apresentados os diferentes modelos de aprendizado de máquina utilizados para a predição.

### 2.4.1 Regressão Linear

A regressão linear busca modelar a relação entre variáveis de entrada  $x_1, x_2, \dots, x_p$  e uma variável de saída  $y$ , assumindo que essa relação é aproximadamente linear. O modelo é definido por:

$$\hat{y} = \beta_0 + \sum_{i=1}^p \beta_i x_i,$$

onde  $\beta_0$  é o intercepto e os coeficientes  $\beta_i$  representam o peso de cada variável preditora.

Os parâmetros  $\beta$  são estimados pelo método dos mínimos quadrados ordinários (MQO), que minimiza o erro quadrático médio:

$$\hat{\beta} = \arg \min_{\beta} \sum_{j=1}^n \left( y_j - \beta_0 - \sum_{i=1}^p \beta_i x_{ij} \right)^2.$$

### 2.4.2 Regression Trees

Uma árvore de regressão divide o espaço das variáveis preditoras em regiões disjuntas  $R_1, R_2, \dots, R_M$ . O modelo pode ser escrito como:

$$\hat{y}(x) = \sum_{m=1}^M c_m \cdot \mathbb{I}(x \in R_m),$$

onde  $\mathbb{I}(\cdot)$  é a função indicadora, que vale 1 se  $x$  pertence à região  $R_m$  e 0 caso contrário. O valor  $c_m$  é geralmente a média dos valores de saída  $y$  dos pontos de treino naquela região.

### 2.4.3 Random Forests

O modelo de Random Forests combina  $B$  árvores de regressão, cada uma construída a partir de amostras bootstrap dos dados de treino e subconjuntos aleatórios das variáveis. A previsão é a média das árvores:

$$\hat{y}(x) = \frac{1}{B} \sum_{b=1}^B T_b(x),$$

onde  $T_b(x)$  representa a previsão da  $b$ -ésima árvore.

### 2.4.4 XGBoost

O XGBoost é baseado em *gradient boosting*, que combina árvores de forma aditiva para corrigir erros sucessivos. O modelo após  $K$  iterações é:

$$\hat{y}^{(K)}(x) = \sum_{k=1}^K f_k(x), \quad f_k \in \mathcal{F},$$

onde cada  $f_k$  é uma árvore de regressão e  $\mathcal{F}$  é o espaço de funções possíveis.

A cada passo, adiciona-se uma nova árvore  $f_k(x)$  para minimizar uma função de perda regularizada:

$$\mathcal{L}^{(K)} = \sum_{j=1}^n l(y_j, \hat{y}^{(K-1)}(x_j) + f_K(x_j)) + \Omega(f_K),$$

onde  $l(\cdot)$  é a função de perda (ex.: erro quadrático) e  $\Omega(f)$  controla a complexidade do modelo.

### 2.4.5 CatBoost

O CatBoost também segue o princípio do *gradient boosting*, mas introduz técnicas específicas para lidar com variáveis categóricas e reduzir o viés preditivo. Sua formulação geral é semelhante ao XGBoost:

$$\hat{y}^{(K)}(x) = \sum_{k=1}^K f_k(x),$$

mas a diferença está no processo de construção das árvores  $f_k(x)$ , que utiliza estratégias de ordenação para codificação de variáveis categóricas e um esquema de regularização que evita o sobreajuste.

## 3 Metodologia

Para atingir os objetivos deste trabalho, foram definidos 2 experimentos...

### 3.1 Experimento 1: Identificar os Modelos mais Promissores

Para esta primeira fase, os hiperparâmetros utilizados foram os já estabelecidos pela linguagem Python, sem nenhuma alteração feita pela pesquisa.

Os modelos de aprendizado foram alimentados com as bases de dados destinadas para treino, X train e y train. Os modelos com target em hysteresis e os com target em joule foram treinados separadamente.

Após essa fase, os dados de X test são fornecidos aos modelos já treinados que fornecem, então, os resultados preditos – y pred. A acurácia dos modelos é avaliada pela semelhança entre os valores reais em y test e os valores preditos em y pred. Essa acurácia é refletida nas métricas MSE, MAPE e  $R^2$  score.

Esse processo foi executado com todos os modelos descritos na seção 2.4.

### 3.2 Experimento 2: Identificar o melhor conjunto de hiper parâmetros para os modelos mais promissores

Após o primeiro experimento, foi observado que os dois modelos mais promissores para predição de perdas são XGBoost e CatBoost.

## 4 Resultados

### 4.1 Resultados em MAPE

### 4.2 Resultados em MAPE

### 4.3 Resultados em $R^2$ Score

## 5 Conclusão

Retome os objetivos e destaque os principais achados. Comente limitações e sugestões para trabalhos futuros.