Trabalho Computacional 1: Modelos de Regressão e Classificacão

Aluisio Gaspar - 2013836, Clysman Alves - 2220304

***Abstract*—The article addresses two distinct studies. The first study focuses on the application of regression models to data from a wind turbine, while the second study deals with classification using electromyography data from facial muscles. In the first study, regression models such as ordinary least squares (OLS), ridge regression, and the mean of observed values were employed. These models were applied to the wind turbine data to analyze and predict patterns related to its operation. In the second study, the adopted classification models were OLS, ridge regression, k-nearest neighbors (K-NN), and minimum distance from the centroid. These models were applied to electromyography data from facial muscles, specifically the Corrugator Supercilii (Sensor 1) and the Zygomaticus Major (Sensor 2), with the aim of classifying the obtained signals. The results of both studies provide valuable insights into the effectiveness of different models in handling the specific data of each domain, highlighting potential practical applications in electrical engineering and biomedical fields.**

***Index Terms*—Regression Models, Classification Models, Artificial Intelligence, Electromyography, Pattern Analysis, Predictive Modeling.**

I. Introdução

MODELOS de inteligência artificial são essenciais para aplicações que buscam emular o pensamento e a capacidade de decisão humana em sistemas computacionais. Eles representam abstrações complexas de como a inteligência pode ser aplicada em problemas específicos, abrangendo desde algoritmos simples até redes neurais profundas. Esses modelos são projetados para aprender com dados, identificar padrões e tomar decisões autônomas, características cruciais exploradas no trabalho proposto. Composto por duas etapas, este trabalho utiliza conceitos de IA baseados em modelos preditivos que aprendem por meio da minimização de uma função custo (loss function). Ambas as etapas empregam o paradigma supervisionado, onde os modelos aprendem a partir de pares de amostra e valor observado. Na primeira etapa, o foco está no desenvolvimento de um sistema para previsões quantitativas (regressão), enquanto a segunda etapa se concentra no desenvolvimento de um sistema para previsões qualitativas (classificação). Essas abordagens representam o alicerce para a construção de soluções inteligentes capazes de lidar com uma variedade de desafios e cenários do mundo real.

II. Tarefa de regressão

A. Entendendo o problema

Para entender o problema em questão, estamos lidando com um conjunto de dados do aerogerador, onde a variável de entrada é a velocidade do vento em metros por segundo (m/s) e a variável de saída é a potência gerada em quilowatts (kWatts). Nosso objetivo é implementar modelos de regressão sobre esse conjunto dados.

B. Interpretando os dados

Realizamos a visualização inicial dos dados por meio de um gráfico de dispersão (Figura 1). Esse gráfico nos permite observar a relação entre as variáveis regressoras (como a velocidade do vento) e as variáveis observadas (como a potência gerada). No contexto da análise dos dados do aerogerador, foi decidido adotar um modelo linear para investigar a relação entre a velocidade do vento e a potência gerada. A escolha por um modelo linear baseia-se na suposição de que existe uma relação direta e proporcional entre essas duas variáveis, o que pode ser representado por uma linha reta em um gráfico de dispersão.

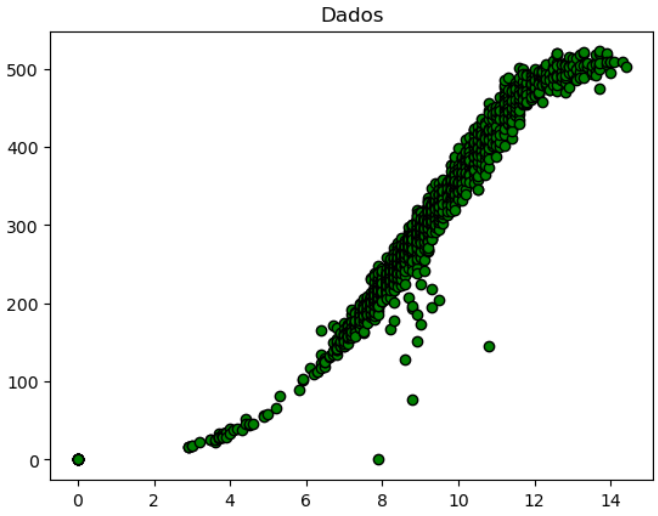

Fig. 1: Visualizando os dados do aerogerador

C. Organizando os dados

Em seguida, foi preciso organizar os dados de forma adequada para a análise. As variáveis regressoras foram armazenadas em uma matriz de dimensão $RN \times p$, onde RN representa o número de amostras e p representa o número de variáveis independentes. Da mesma forma, o vetor de variáveis observadas foi organizado em um vetor de dimensão $RN \times 1$. Essa organização nos permitirá realizar a implementação dos modelos posteriormente com os dados devidamente estruturados.

D. Definição da Quantidade de Rodadas

Foi estabelecido que seria necessário validar os modelos de regressão com uma quantidade específica de rodadas de treinamento e teste. Para isso, foi definido o valor de 1000 rodadas. Essa quantidade permite uma avaliação robusta da performance dos modelos ao longo de múltiplas iterações, reduzindo o viés e a variabilidade nos resultados.

E. Implementação dos modelos

Os modelos selecionados para esta etapa incluem o Método dos Quadrados Mínimos Ordinários (MQO) tradicional, o MQO regularizado (utilizando o método de Tikhonov) e a média de valores observáveis. Cada um desses modelos oferece abordagens distintas para lidar com a tarefa de regressão e será avaliado em relação à sua capacidade de prever a potência gerada com base na velocidade do vento.

Em cada modelo, a métrica que será analisada é o Erro Quadrático Médio (EQM). O foco principal será na minimização desse erro, buscando encontrar o modelo que oferece as previsões mais precisas e próximas dos valores reais da potência gerada pelo aerogerador.

O cálculo do erro quadrático médio (EQM) é uma medida comum utilizada para avaliar a qualidade de um modelo de regressão. Ele é calculado como a média dos quadrados das diferenças entre as previsões do modelo e os valores reais observados da variável dependente.

A fórmula geral para calcular o EQM é a seguinte:

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

Onde:

- n é o número de observações nos dados,
- y_i é o valor real da variável dependente para a i-ésima observação,
- $\hat{y}_i$ é o valor previsto pelo modelo para a i-ésima observação.

O EQM fornece uma medida da dispersão dos erros do modelo, ou seja, da diferença entre as previsões do modelo e os valores reais. Quanto menor o valor do EQM, mais próximas as previsões do modelo estão dos valores reais, e melhor é a performance do modelo.

F. Implementação do modelo utilizando MQO Tradicional

Para implementar o Método dos Quadrados Mínimos Ordinários (MQO), **Algoritmo 1**, utilizamos uma abordagem matricial, que nos permite trabalhar eficientemente com matrizes e operações de álgebra linear.

Para utilizar o MQO, foi preciso adicionar uma coluna de 1s à matriz de variáveis independentes, a fim de levar em conta o termo de interceptação no modelo de regressão linear, permitindo que o resultado do nosso modelo não sejam limitados a origem.

Para calcular os parâmetros do modelo MQO utilizamos a seguinte equação normal:

$$\beta = (X^T X)^{-1} X^T y$$

Onde:

- X é a matriz de variáveis independentes,
- y é o vetor de variável dependente,
- β são os parâmetros estimados.

Uma vez obtidos os parâmetros do modelo, podemos calcular as previsões multiplicando a matriz de variáveis independentes pelos parâmetros estimados. Em seguida, calculamos o erro quadrático médio (EQM).

Algorithm 1 Algoritmo para cálculo dos parâmetros do MQO e erro quadrático médio

Require: $X_{\text{train}}, y_{\text{train}}, X_{\text{test}}, y_{\text{test}}$
Ensure: $b_{\text{hat\_ols}}, y_{\text{pred\_ols}}, \text{mse\_ols}$
1: $b_{\text{hat\_ols}} \leftarrow \text{inv}(X_{\text{train}}^T \cdot X_{\text{train}}) \cdot X_{\text{train}}^T \cdot y_{\text{train}}$
2: $y_{\text{pred\_ols}} \leftarrow X_{\text{test}} \cdot b_{\text{hat\_ols}}$
3: $\text{mse\_ols} \leftarrow \text{mean\_squared\_error}(y_{\text{test}}, y_{\text{pred\_ols}})$

G. Implementação do modelo utilizando MQO Regularizado(Tikhonov)

A implementação do modelo utilizando o Método dos Quadrados Mínimos Ordinários (MQO) regularizado Tikhonov, **Algoritmo 2**, também conhecido como regularização de Ridge, envolve a introdução de um termo de penalidade na função de custo, com o objetivo de evitar overfitting e estabilizar as estimativas dos parâmetros do modelo. A regularização Tikhonov é especialmente útil quando há multicolinearidade entre as variáveis independentes.

Para calcular os parâmetros do modelo MQO Regularizado(Tikhonov) utilizamos a seguinte equação normal:

$$\beta = (X^T X + \lambda I)^{-1} X^T y$$

Onde:

- X é a matriz de variáveis independentes,
- y é o vetor de variáveis dependentes,
- λ é o parâmetro de regularização,
- I é a matriz identidade.

O parâmetro de regularização λ é um hiperparâmetro do modelo que controla o grau de penalidade aplicado aos parâmetros do modelo. Seu valor é ajustado para ser aquele que tem o valor médio mínimo de EQM.

Uma vez calculados os parâmetros do modelo, as previsões são feitas utilizando a matriz de variáveis independentes. A qualidade do modelo é então avaliada usando métricas apropriadas, como o erro quadrático médio (EQM), em um conjunto de dados de teste independente.

Algorithm 2 Algoritmo para cálculo dos parâmetros do MQO Regularizado(Tikhonov) e erro quadrático médio

Require: $X_{\text{train}}, y_{\text{train}}, X_{\text{test}}, y_{\text{test}}$
Ensure: $b_{\text{hat\_ols\_rigde}}, y_{\text{pred\_ols}}, \text{mse\_ols}$
1: $b_{\text{hat\_ols\_rigde}} \leftarrow \text{inv}(X_{\text{train}}^T \cdot X_{\text{train}} + \lambda I) \cdot X_{\text{train}}^T \cdot y_{\text{train}}$
2: $y_{\text{pred\_ols}} \leftarrow X_{\text{test}} \cdot b_{\text{hat\_ols\_rigde}}$
3: $\text{mse\_ols} \leftarrow \text{mean\_squared\_error}(y_{\text{test}}, y_{\text{pred\_ols}})$

H. Implementação do modelo utilizando Média de valores observáveis

A implementação desse modelo, **Algoritmo 3**, é uma abordagem simples para previsão. Nesse método, a previsão para cada amostra é simplesmente a média dos valores observados no conjunto de treinamento.

Para cada variável dependente no conjunto de treinamento, calculamos a média dos valores observados. Isso nos dá um único valor para cada variável dependente. Para cada amostra no conjunto de teste, atribuímos a previsão como sendo a média calculada para a respectiva variável dependente.

Uma vez que as previsões tenham sido feitas para o conjunto de teste, podemos avaliar o desempenho do modelo utilizando métricas adequadas, como o erro quadrático médio (EQM).

Algorithm 3 Algoritmo para o modelo de regressão Média dos Valores Observados

Require: $X_{\text{train}}, y_{\text{train}}, X_{\text{test}}, y_{\text{test}}$
Ensure: $mean\_y\_train, y\_pred\_mean, mean\_mse$
1: $mean\_y\_train \leftarrow \text{mean}(y\_train)$
2: $y\_pred\_mean \leftarrow mean\_y\_train \times y\_test$
3: $mean\_mse \leftarrow \text{mean\_squared\_error}(y\_test, y\_pred\_mean)$

I. Validação dos modelos

Antes de dividir os dados em conjuntos de treinamento e teste, as amostras do conjunto de dados completo são embaralhadas. Isso é feito para garantir que não haja viés na distribuição dos dados e que cada partição contenha uma representação aleatória das amostras.

Após o embaralhamento, os dados são divididos em dois conjuntos: um conjunto de treinamento e um conjunto de teste. Neste caso, 80% dos dados são atribuídos ao conjunto de treinamento e 20% ao conjunto de teste. Essa divisão permite que o modelo seja treinado em uma porção dos dados e avaliado em outra porção independente, o que ajuda a estimar sua capacidade de generalização para novos dados não vistos.

O processo de embaralhamento e particionamento é repetido várias vezes, em cada rodada de validação. Isso é feito para garantir que cada amostra tenha a oportunidade de estar tanto no conjunto de treinamento quanto no conjunto de teste em diferentes iterações. Geralmente, é especificada uma quantidade fixa de rodadas de validação, como mencionado no contexto (1000 rodadas).

Durante cada rodada de validação, é calculado e armazenado o erro quadrático médio (EQM) com base nas previsões do modelo e nos valores reais nos dados de teste.

J. Resultados

Para cada modelo utilizado, os valores de EQM obtidos em todas as 1000 rodadas foram coletados. Em seguida, foram calculadas as seguintes estatísticas: média, desvio padrão, valor máximo e valor mínimo do EQM.

Os valores calculados das estatísticas foram organizados em uma tabela e em um gráfico para cada modelo. Na tabela, cada linha representa um modelo, e as colunas correspondem às estatísticas calculadas (média, desvio padrão, máximo e mínimo do EQM).

Exemplo de resultados obtidos:

Modelo	Mean MSE	Desvio Padrão	Mínimo/Máximo MSE
MQO	789.69	± 163.09	$423.71/1414.61$
Ridge	782.77	± 156.00	$413.59/1276.11$
MVO	11155.64	± 623.29	$9275.98/13353.02$

Fonte: Dados do aerogerador (*)Ridge: $\lambda \approx 0.63$

TABLE I: Tabela de Comparação dos Resultados dos Modelos

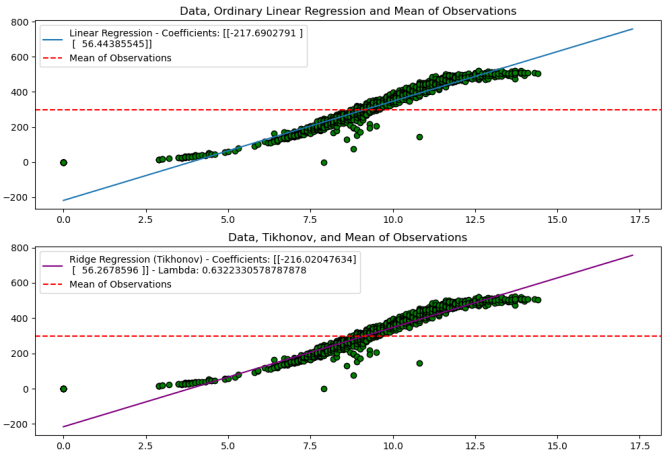

Fig. 2: Gŕafico de Comparação dos Resultados dos Modelos

Uma semelhança notável entre os modelos de Mínimos Quadrados Ordinários (MQO) e Mínimos Quadrados Ordinários Regularizados (MQO regularizado) reside nos resultados do modelo, especialmente quando se trata da previsão da variável dependente

Embora essas semelhanças existam nos resultados dos modelos, é importante notar que o MQO regularizado oferece uma vantagem adicional em termos de controle sobre o ajuste excessivo (overfitting). Ao adicionar uma penalidade à função de perda, o MQO regularizado pode ser mais eficaz na criação de modelos que generalizam bem para novos dados, especialmente em cenários com muitas variáveis independentes ou com multicolinearidade entre as variáveis independentes. No entanto, em situações onde não há problemas de multicolinearidade ou overfitting, os resultados do MQO regularizado podem ser semelhantes aos do MQO tradicional.

III. Trabalho de Classificação

A. Entendendo o problema

Para o trabalho de classificação, o conjunto de informações será relacionado aos sinais de eletromiografia dos músculos faciais. Especificamente, temos dados referentes ao Corrugador do Supercílio (Sensor 1) e ao Zigomático Maior (Sensor 2). Este conjunto de dados é composto por 50.000 observações para ambos os sensores, distribuídas em classes perfeitamente balanceadas, totalizando 10.000 observações para cada classe.

Os dados fornecidos consistem em 10 rodadas de aquisições dos sinais de EMG. Em cada rodada, as aquisições seguem uma ordem específica: 1000 dados para o gesto neutro, 1000

dados para o gesto sorridente, 1000 dados para o gesto aberto, 1000 dados para o gesto surpreso e 1000 dados para o gesto rabugento.

Para uma amostra, o vetor de características pode ser representado por:

$$x = [x1, x2] = [\text{Dado lido pelo sensor } 1, \text{Dado lido pelo sensor } 2]$$

Onde:

- $xj \in [0, 1, 2, ...4095]$,
- $j = 1, 2$.

B. Interpretando os dados

Realizamos a visualização inicial dos dados por meio de um gráfico de dispersão (Figura 3). Esse gráfico nos permite observar a distribuição das cinco classes de gestos: neutro, sorridente, aberto, surpreso e rabugento. Cada classe é representada por um conjunto de pontos no gráfico, onde cada ponto corresponde a uma observação.

Ao observar a complexidade das relações entre os preditores e as classes de gestos faciais nos dados de sinais de eletromiografia, optar por um modelo que tenha carateristicas não lineares é essencial para capturar a variedade de padrões e nuances presentes nas expressões faciais. Modelos não lineares oferecem maior flexibilidade e capacidade de generalização, permitindo assim a captura de interações complexas entre os preditores. Dessa forma, estamos buscando uma abordagem mais precisa para a classificação das expressões faciais, a fim de proporcionar resultados mais robustos e próximos do real.

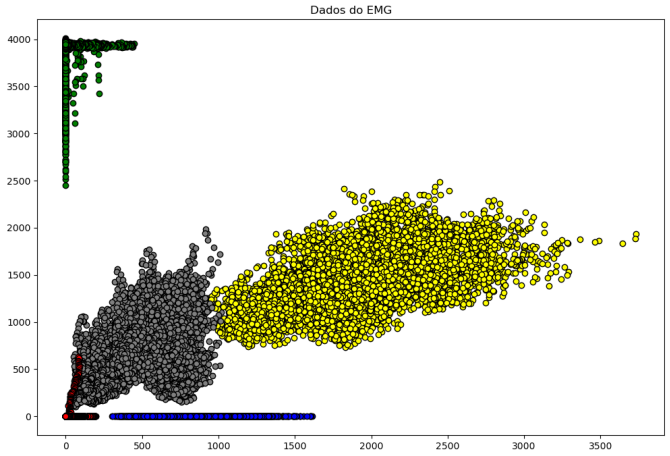

Fig. 3: Dados do EMG

C. Organizando os dados

Para organizar os dados conforme especificado, primeiro identificamos os parâmetros fornecidos: número de preditores (p = 2), quantidade de amostras (N = 50.000) e quantidade de classes (c = 5). Em seguida, preparamos os dados da seguinte forma:

1) Os dados de entrada (X) consistem em uma matriz de dimensões $50,000 \times 2$, representando a atividade do corrugador do supercílio e do zigomático maior para cada amostra.
2) Os dados de saída (Y) consistem em uma matriz de dimensões $50,000 \times 2$, onde cada elemento representa a classe associada a uma amostra. Para cada amostra do conjunto de dados, sua matriz de saída (Y) acompanhará a seguinte organização:
   a) Neutro: $Y = [1, -1, -1, -1, -1]$
   b) Sorrindo: $Y = [-1, 1, -1, -1, -1]$
   c) Aberto: $Y = [-1, -1, 1, -1, -1]$
   d) Surpreso: $Y = [-1, -1, -1, 1, -1]$
   e) Rabugento: $Y = [-1, -1, -1, -1, 1]$

D. Definição da Quantidade de Rodadas

Foi estabelecido que seria necessário validar os modelos de classificação com uma quantidade específica de rodadas de treinamento e teste. Para isso, foi definido o valor de 100 rodadas. Essa quantidade permite uma avaliação robusta da performance dos modelos ao longo de múltiplas iterações, reduzindo o viés e a variabilidade nos resultados.

E. Implementação dos modelos

Os modelos selecionados para esta etapa incluem o Método dos Quadrados Mínimos Ordinários (MQO) tradicional, o MQO regularizado (utilizando o método de Tikhonov), Classificador k-Vizinhos mais Próximos (k-NN) e Distância Mínima ao Centróide (DMC). Cada um desses modelos oferece abordagens distintas para lidar com a tarefa de classificação e será avaliado em relação à sua capacidade de prever a classe dado um sinal de eletromiografia.

Em cada modelo, a métrica que será analisada é a acurácia. O foco principal será na maximização da acurácia, buscando encontrar o modelo que oferece as previsões mais precisas e próximas dos valores reais das classes para cada sinal de EMG.

A acurácia é uma métrica comum utilizada para avaliar a qualidade de modelos de classificação. Ela representa a proporção de previsões corretas feitas pelo modelo em relação ao total de observações.

A fórmula para calcular a acurácia é a seguinte:

$$\text{Acurácia} = \frac{\text{Número de previsões corretas}}{\text{Número total de observações}}$$

A acurácia fornece uma medida da precisão do modelo em classificar corretamente as observações. Quanto maior a acurácia, melhor é a performance do modelo em fazer previsões precisas.

F. Implementação do modelo utilizando MQO Tradicional

A implementação do MQO Tradicional segue os mesmos princípios já abordados na seção II-F, com a diferença de que a métrica utilizada será a acurácia.

Neste algoritmo, *accuracy* é uma função que calcula a acurácia comparando as previsões feitas pelo modelo ($y_{\text{pred\_ols}}$) com as classes reais das observações (y_{test}). O algoritmo para calcular a acurácia é mostrado abaixo:

Algorithm 4 Algoritmo para cálculo dos parâmetros do MQO e acurácia

Require: $X_{\text{train}}, y_{\text{train}}, X_{\text{test}}, y_{\text{test}}$
Ensure: $b_{\text{hat\_ols}}, y_{\text{pred\_ols}}, \text{accuracy\_ols}$
1: $b_{\text{hat\_ols}} \leftarrow \text{inv}(X_{\text{train}}^T \cdot X_{\text{train}}) \cdot X_{\text{train}}^T \cdot y_{\text{train}}$
2: $y_{\text{pred\_ols}} \leftarrow X_{\text{test}} \cdot b_{\text{hat\_ols}}$
3: $\text{accuracy\_ols} \leftarrow \text{accuracy}(y_{\text{test}}, y_{\text{pred\_ols}})$

Algorithm 5 Algoritmo para cálculo da acurácia

Require: $y_{\text{true}}, y_{\text{pred}}$
Ensure: accuracy
1: $n_{\text{correct}} \leftarrow \text{sum}(y_{\text{true}} == y_{\text{pred}})$
2: $\text{accuracy} \leftarrow \frac{n_{\text{correct}}}{\text{len}(y_{\text{true}})}$

G. Implementação do modelo utilizando MQO Regularizado(Tikhonov)

A implementação do MQO Tradicional segue os mesmos princípios já abordados na seção II-G, com a diferença de que a métrica utilizada será a acurácia.

Algorithm 6 Algoritmo para cálculo dos parâmetros do MQO Regularizado(Tikhonov) e acurácia

Require: $X_{\text{train}}, y_{\text{train}}, X_{\text{test}}, y_{\text{test}}$
Ensure: $b_{\text{hat\_ols\_ridge}}, y_{\text{pred\_ols\_ridge}}, \text{accuracy\_ols\_ridge}$
1: $b_{\text{hat\_ols\_rigde}} \leftarrow \text{inv}(X_{\text{train}}^T \cdot X_{\text{train}} + \lambda I) \cdot X_{\text{train}}^T \cdot y_{\text{train}}$
2: $y_{\text{pred\_ols\_ridge}} \leftarrow X_{\text{test}} \cdot b_{\text{hat\_ols\_ridge}}$
3: $\text{accuracy\_ols\_ridge} \leftarrow \text{accuracy}(y_{\text{test}}, y_{\text{pred\_ols\_ridge}})$

H. Implementação do modelo utilizando Classificador K-Vizinhos mais Próximos (k-NN)

Na implementação do modelo utilizando o algoritmo k-NN (k-Vizinhos Mais Próximos), seguimos alguns passos essenciais.

Primeiramente, escolhemos um valor apropriado para o parâmetro k, que representa o número de vizinhos mais próximos a serem considerados ao fazer uma previsão. O valor escolhido foi o que gerou o valor médio maior de acurácia.

Em seguida, calculamos a distância euclidiana entre o ponto de teste e todos os pontos de treinamento. A distância também pode ser calculadas utilizando outras métricas, como a distância de Manhattan ou a distância de Minkowski.

Uma vez calculadas as distâncias, identificamos os k-vizinhos mais próximos do ponto de teste. Esses vizinhos são determinados com base nas menores distâncias calculadas anteriormente.

Com os k-vizinhos mais próximos identificados, determinamos a classe atribuída ao ponto de teste. Essa classe é determinada pela classe mais frequente entre os k vizinhos mais próximos.

Finalmente, avaliamos o desempenho do modelo utilizando a acurácia.

Algorithm 7 Algoritmo para o modelo de classificação K-NN

Require: Conjunto de treinamento $X_{\text{train}}, y_{\text{train}}$ e conjunto de teste X_{test}
Ensure: Vetor de previsões y_{pred}
1: Escolha um valor para k
2: $n_{\text{correct}} \leftarrow 0$
3: **for** cada ponto de teste $x_i \in X_{\text{test}}$ **do**
4: Calcule a distância entre x_i e todos os pontos de treinamento em X_{train}
5: Identifique os k vizinhos mais próximos de x_i
6: Atribua a classe mais frequente entre os k vizinhos mais próximos como a classe de x_i
7: **if** a classe atribuída é igual à classe real de x_i **then**
8: $n_{\text{correct}} \leftarrow n_{\text{correct}} + 1$
9: **end if**
10: **end for**
11: $\text{accuracy} \leftarrow \frac{n_{\text{correct}}}{\text{len}(X_{\text{test}})}$

I. Implementação do modelo utilizando Mínima Distância ao Centróide

Para realizar a implementação do modelo utilizando a Mínima Distância ao Centróide, seguimos uma série de passos que nos permitiram alcançar o resultado desejado.

Inicialmente, calculamos os centróides para cada classe no conjunto de treinamento. O centróide de uma classe é o ponto médio das amostras pertencentes a essa classe no espaço de características. Quando uma nova amostra do conjunto de teste é apresentada ao modelo, calculamos a distância entre essa amostra e cada centróide. A nova amostra é então atribuída à classe cujo centróide está mais próximo dela, ou seja, à classe cuja a distância euclidiana entre a amostra e o centróide é a menor.

Durante a fase de teste, este processo de cálculo da distância e atribuição de classes é repetido para cada nova amostra apresentada ao modelo. Ao final, avaliamos o desempenho do modelo utilizando a acurácia.

Algorithm 8 Algoritmo para um modelo de classificação utilizando Mínima Distância ao Centróide

Require: Conjunto de treinamento $X_{\text{train}}, y_{\text{train}}$ e conjunto de teste $X_{\text{test}}, y_{\text{test}}$
Ensure: Vetor de previsões y_{pred}
1: $n_{\text{correct}} \leftarrow 0$
2: Calcular os centróides para cada classe em X_{train}
3: **for** cada amostra $x_i \in X_{\text{test}}$ **do**
4: Calcular a distância entre x_i e cada centróide
5: Atribuir a classe cujo centróide está mais próximo de x_i
6: Armazenar a previsão em y_{pred}
7: **if** y_{pred} é igual à classe real de x_i **then**
8: $n_{\text{correct}} \leftarrow n_{\text{correct}} + 1$
9: **end if**
10: **end for**
11: $\text{accuracy} \leftarrow \frac{n_{\text{correct}}}{\text{len}(X_{\text{test}})}$

J. Validação e Resultados

Para realizar a análise do desempenho dos modelos após as 100 rodadas de treinamento e teste, seguimos um procedimento sistemático. Primeiramente, para cada modelo utilizado, calculamos a acurácia em cada uma das 100 rodadas.

Em seguida, para cada modelo, computamos as estatísticas resumidas das acurácias obtidas. Isso incluiu o cálculo da média, do desvio padrão, do valor máximo, do valor mínimo e da moda das acurácias.

Após obter esses valores, organizamos os resultados em uma tabela e um gráfico para facilitar a visualização e comparação entre os diferentes modelos.

Exemplo de resultados obtido:

Modelo	Média	Desvio Padrão	Mínima/Máxima	Moda
MQO	72.20	± 0.0018	72.48/71.99	72.26
Ridge	72.19	± 0.0016	72.48/71.99	72.26
K-NN	98.12	± 0.003	97.48/99.94	98.76
DMC	96.05	± 0.0016	96.28/95.9	95.89

Fonte: Dados do EMG (*)Ridge: $\lambda \approx$ 1e-05, K = 7

TABLE II: Tabela de Comparação dos Resultados dos Modelos de Classificação

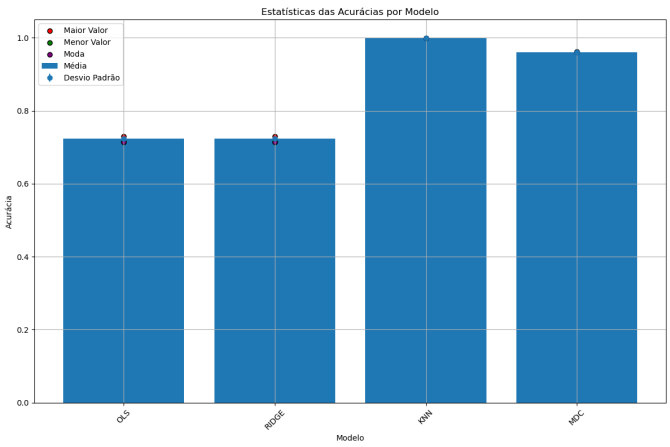

Fig. 4: Gráfico de Comparação dos Resultados dos Modelos de Classificação

Ao discutir os resultados obtidos, consideramos não apenas as métricas de desempenho médio, mas também a variabilidade das acurácias entre as rodadas. Isso nos permitiu identificar padrões de consistência ou variabilidade nos diferentes modelos.

Como observado na tabela e no gráfico, modelos com características não lineares apresentaram uma tendência a terem uma acurácia média maior em comparação com modelos lineares. Isso sugere que a capacidade de capturar relações não lineares nos dados pode ser crucial para obter um desempenho superior em determinados problemas.

IV. Conclusão

Portanto, este estudo destaca a eficácia dos modelos de regressão e classificação na análise de dados de grandes conjuntos de dados. Ao empregar técnicas como mínimos quadrados ordinários, MQO regularizado e métodos de classificação como k-vizinhos mais próximos e distância mínima do centroide, pudemos observar como esses modelos podem ser aplicados com sucesso na previsão e classificação dos dados.

Os resultados obtidos não apenas fornecem uma compreensão mais profunda de como realizar analises preditivas, mas também destacam o potencial desses modelos para aplicações práticas em diversos âmbitos. Essas descobertas têm implicações significativas para o desenvolvimento de sistemas de monitoramento e controle em diversas áreas, desde a saúde até a eficiência energética. No futuro, pesquisas adicionais podem explorar ainda mais essas aplicações e aprimorar os modelos existentes para melhor atender às necessidades desses domínios.

V. Implementações

Disponivel em: regression-and-classification-models

References

[1] T. C. Carneiro, P. A. Rocha, P. C. Carvalho, and L. M. Fernández-Ramírez, "Ridge regression ensemble of machine learning models applied to solar and wind forecasting in brazil and spain," *Applied Energy*, vol. 314, p. 118936, 2022.

[2] S. Haykin, *Redes neurais: princípios e prática*. Bookman Editora, 2001.

[3] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. Springer, 2003, pp. 986–996.

[4] E. Z. G. Max, "Seleção de instâncias baseado em aprendizado de métricas para k vizinhos mais próximos," 2016.

[5] D. Figueiredo Filho, F. Nunes, E. C. da Rocha, M. L. Santos, M. Batista, and J. A. S. Júnior, "O que fazer e o que não fazer com a regressão: pressupostos e aplicações do modelo linear de mínimos quadrados ordinários (mqo)," *Revista Política Hoje*, vol. 20, no. 1, 2011.