UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS SOBRAL

Curso: Engenharia da Computação
Disciplina: Inteligência Computacional
Prof. Jarbas Joaci de Mesquita Sá Junior
1º Trabalho

Aluno: Francisco Jeferson da Silveira Pontes Matrícula: 397888

As questões foram elaboradas na linguagem de programação python, versão 3.6, e com o auxílio da IDE PyCharm 2019.2.2, e está disponível em : https://github.com/Jefersonpontes13/Trabalho-01-IC.

Se houver problemas ao executar, a pasta do projeto está no repositório, e contém o ambiente virtual do python3.6, que é adequado para execução.

1. Encontre o máximo da função f(x,y) = xsen(yπ/4) + ysen(xπ/4) por meio do algoritmo hill-climbing. As variáveis x e y pertencem ao intervalo entre 0 e 20. Os vizinhos de determinado estado (x, y) são (x, y ± 0,01), (x ± 0,01, y) e (x ± 0,01, y ± 0,01). Por exemplo, os vizinhos do estado (1,00; 1,00) são (1,00; 1,01), (1,01; 1,01), (0,99; 0,99), (0,99; 1,00) etc.

O algoritmo gera uma posição dentro do intervalo X = [0 , 20], e Y = [0, 20], e partindo da premissa do algoritmo Hillclibing Original (maior passo), move-se para o estado vizinho que proporciona a maior melhoria, aumento no valor da função objetivo, no caso de f(x, y), para no máximo local mais próximo.

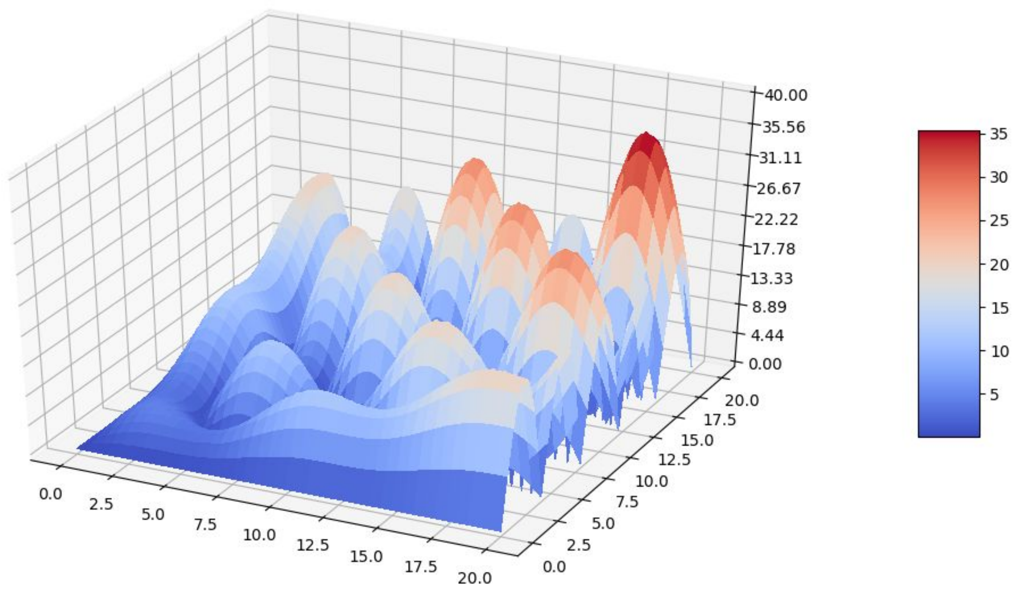

Figura - 01 : f(x, y) = xsen(yπ/4) + ysen(xπ/4)

Plotando o gráfico, é perceptível que a melhor solução do problema está entre 15 e 20 nos eixos X e Y, e f(x, y) por volta de 35. A resolução está no arquivo Hillclibing.py, e além dele, o arquivo Log-Hilclibing.txt auxilia na compreensão da resolução do problema.

Executando o script Hillclibing.py: O algoritmo implementa o Random-Restart: Múltiplas buscas hill-climbing a partir de diferentes estados iniciais gerados aleatoriamente. Foi definido 10 tentativas, e os resultados são gravados no arquivo Log-Hillclibing.txt, logo, é possível identificar se houve êxito na busca do máximo global, que no problema é 36,088, nas coordenadas X = 18,09, e Y = 18,09.

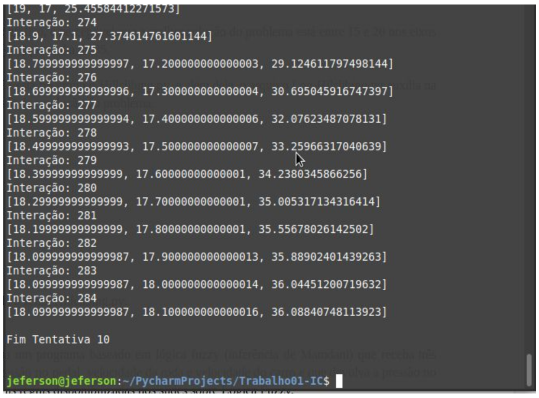

Fig - 2 : Terminal

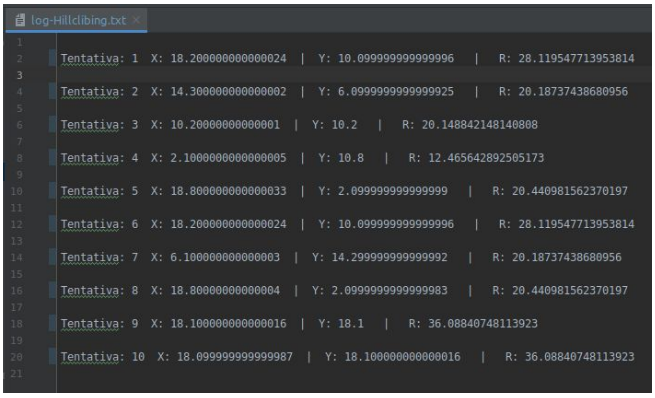

Fig - 3 : Log-Hillclibing.txt

Fig - 2: Mostra as interações e os valores obtidos.
Fig - 3: Mostra os resultados obtidos em cada tentativa.

2. Construa um programa baseado em lógica fuzzy (inferência de Mamdani) que receba três valores: pressão no pedal, velocidade da roda e velocidade do carro e que devolva a pressão no freio. Siga as regras disponibilizadas nos slides sobre Lógica Fuzzy.

Com base nas informações dos slides sobre Lógica fuzzy, a solução foi modelada no arquivo Fuzzy.py. Ao executar, é preciso definir os valores de entrada no terminal do python, e logo após, imprime os gráficos das funções de pertinência, e o resultado final da desnebulização e o gráfico de corte das funções de pertinência das variáveis (Apertar e Liberar freio) nos valores obtidos no processo de nebulização.

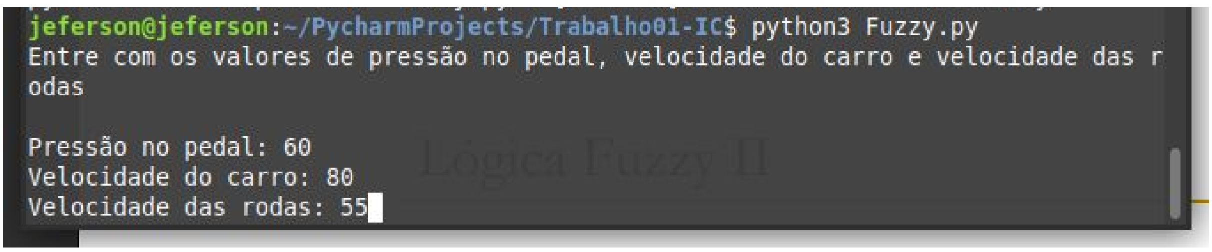

Fig - 4 : Inserção dos valores de entrada.

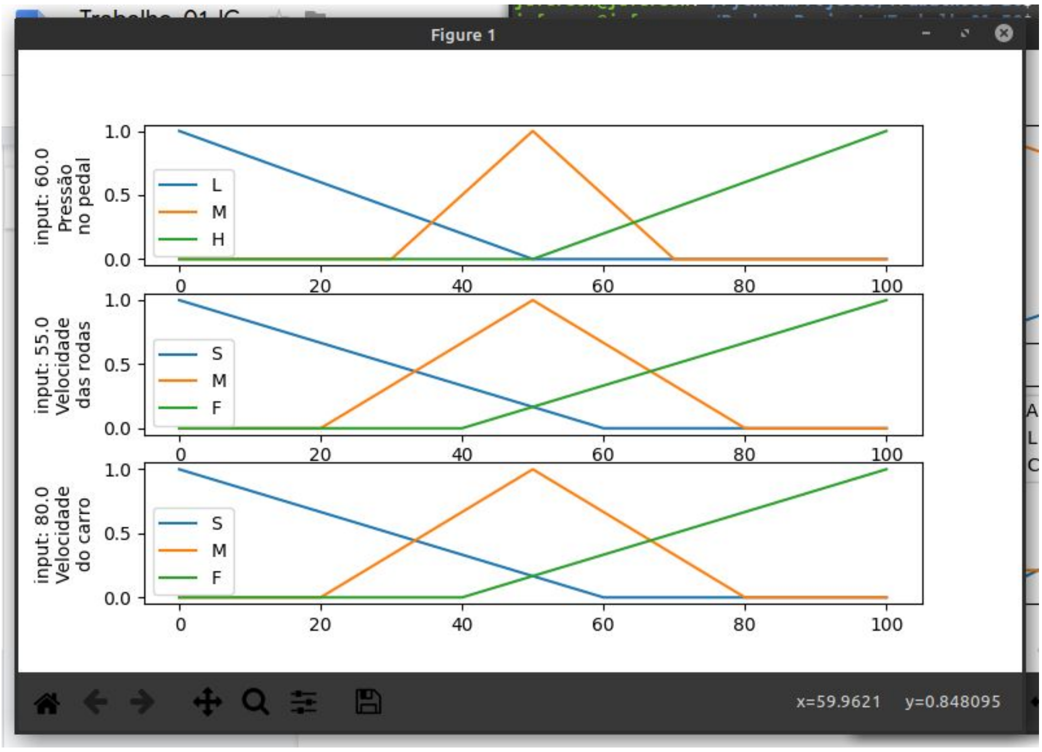

Fig - 5 : Gráfico das funções de pertinência.

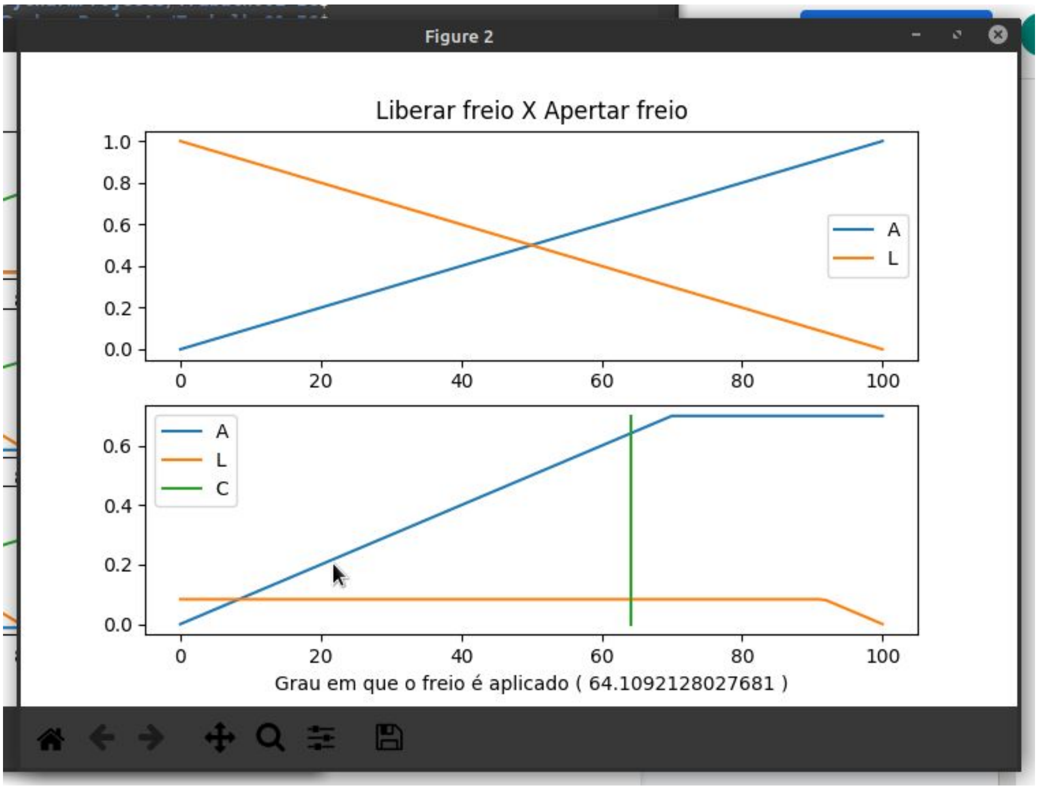

Fig - 6 : Resultado final da desnebulização.

3. Usando o conjunto de dados do aerogerador (variável de entrada: velocidade do vento – m/s, variável de saída: potência gerada – kWatts), determine os modelos de regressão polinomial (graus 2, 3, 4 e 5) com parâmetros estimados pelo método dos mínimos quadrados. Avalie a qualidade de cada modelo pela métrica R2 e R2aj (equações 48 e 49, slides sobre Regressão Múltipla.

A solução está no arquivo Regressao.py, ao executar o algoritmo, deve-se inserir o grau do polinômio, e automaticamente será gerado o gráfico da distribuição dos dados e a função de grau pré-definido.

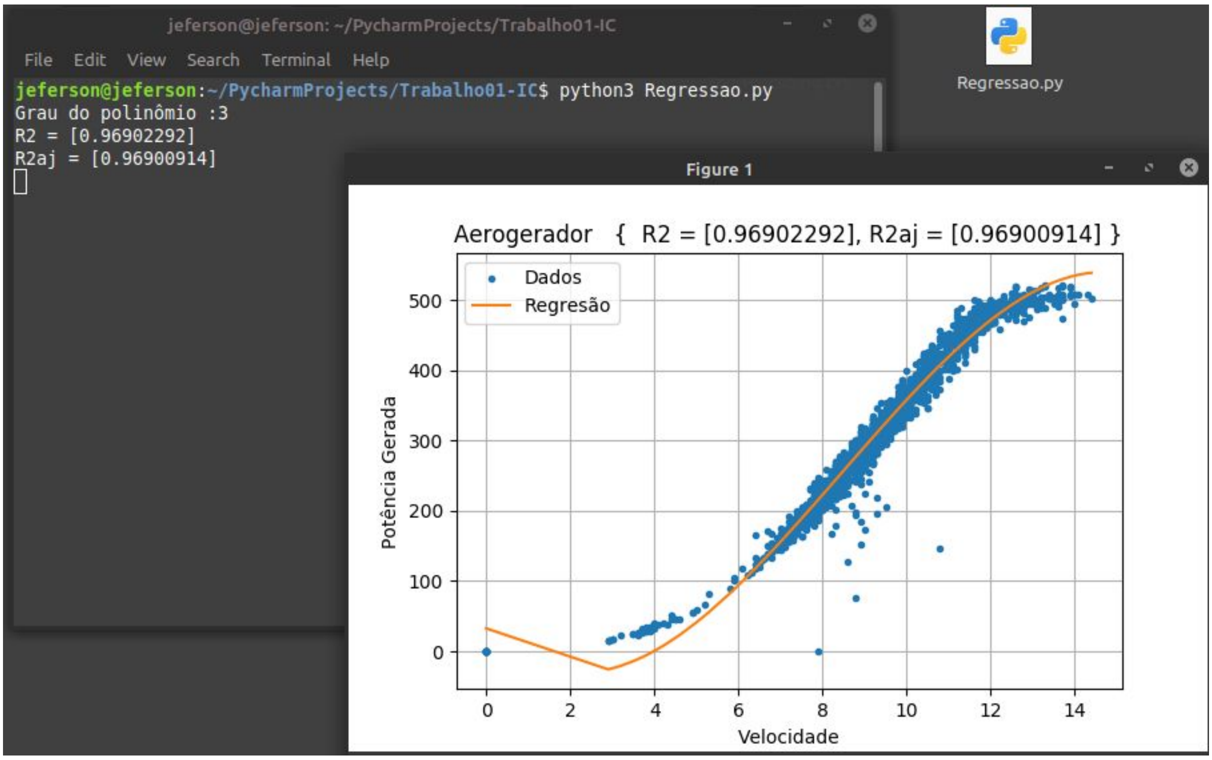

Fig - 7 : Execução do algoritmo.

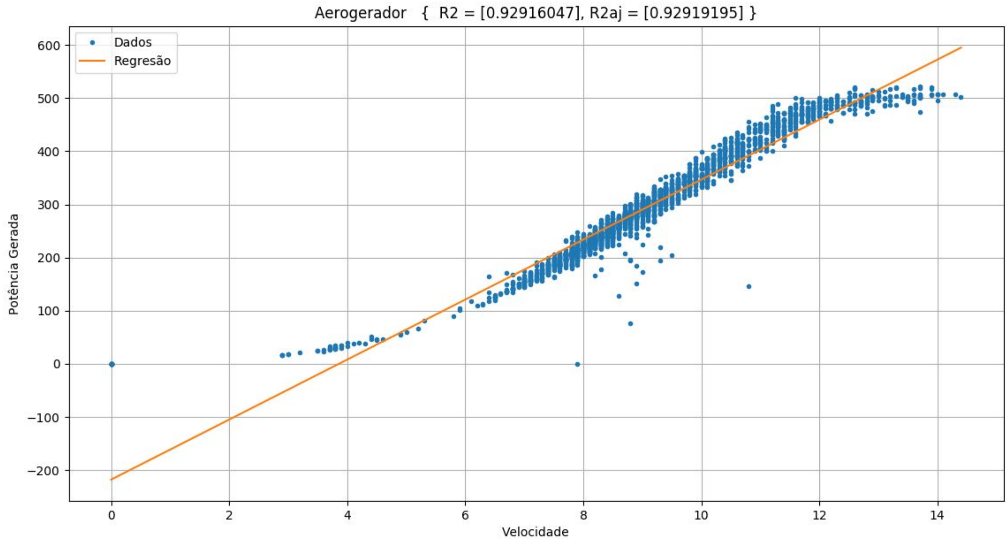

Fig - 8 : Regressão linear.

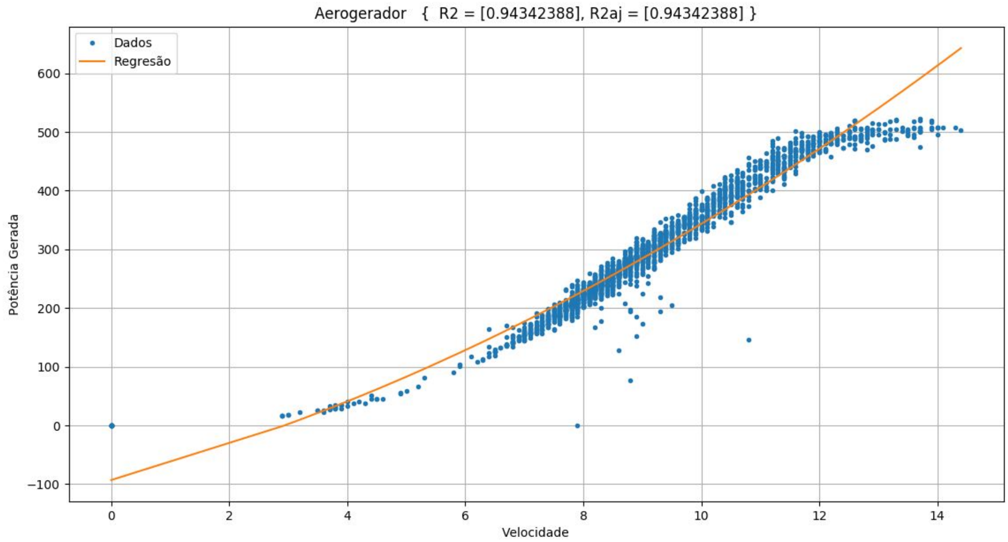

Fig - 9 : Regressão de grau 2.

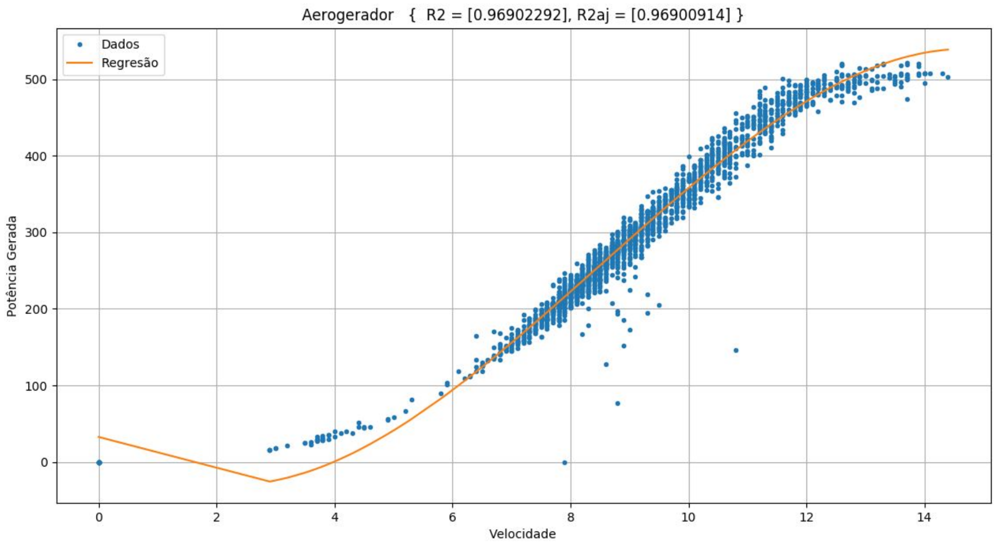

Fig - 10 : Regressão de grau 3.

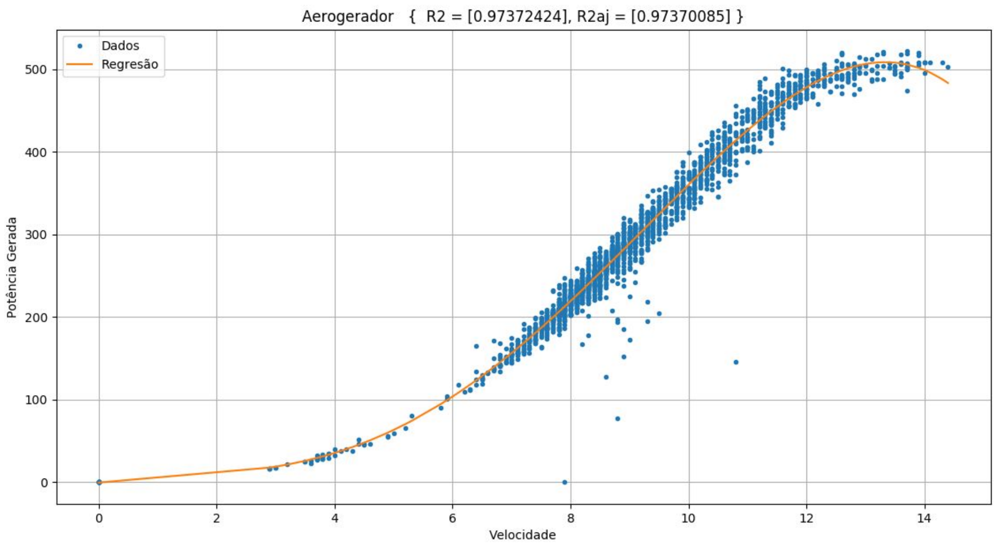

Fig - 11 : Regressão de grau 4.

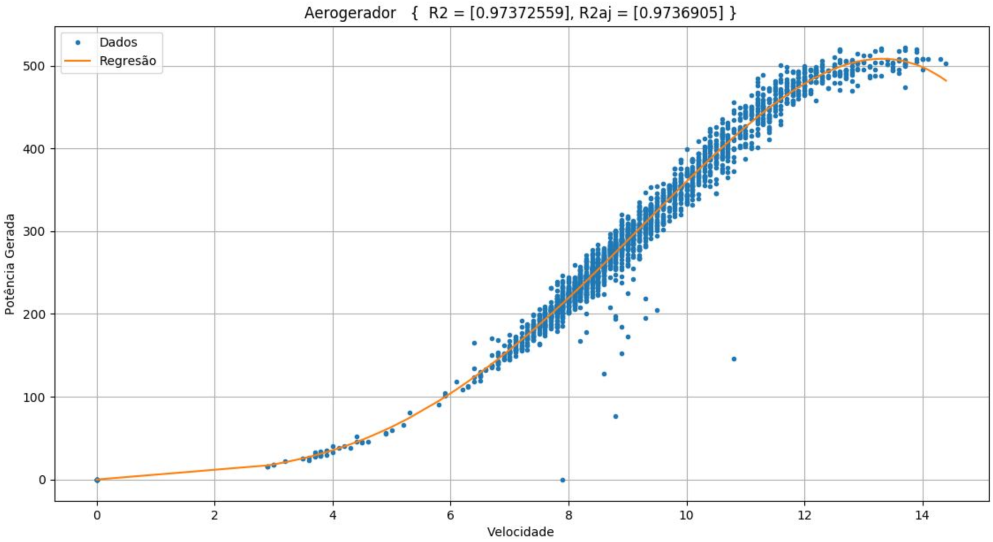

Fig - 12 : Regressão de grau 5.

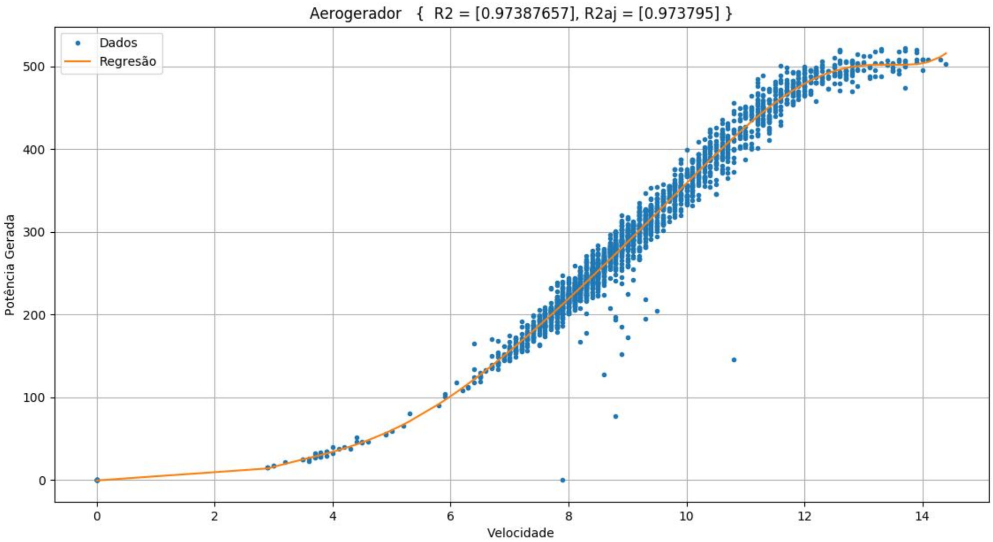

Fig - 13 : Regressão de grau 9.

Foram feitas regressões de grau 1 a 5, e uma especial de grau 9, a partir dos gráficos e de R2 e R2aj, pode-se concluir que houve uma maior correspondência do comportamento do gráfico da regressão com o comportamento dos dados, conforme aumenta o grau do polinômio, mas nitidamente o ganho é reduzido drasticamente a partir da quarta regressão, e o polinômio de grau 9, obteve um ganho quase que desprezível, comparado com a de grau 5.
Além do mais, Analisando os resultados de R2, e R2aj, percebe-se que até grau 4 os valores são similares, mas a partir do grau 5, R2aj aponta uma diminuição no desempenho da regressão múltipla, embora R2 insista em mostrar um ganho. Logo, é mais interessante usar R2aj como parâmetro de análise de desempenho de uma regressão polinomial, pois de maneira subjetiva aponta que o ganho não vale o custo computacional e de espaço, para um polinômio de grau muito elevado.