CENTRO UNIVERSITÁRIO FEI

GABRIELA MOLINA CIOCCI

**USO DE REDES NEURAIS PARA ANÁLISE DE DADOS NAS DISCIPLINAS DE LABORATÓRIO DE FÍSICA**

São Bernardo do Campo

2024

GABRIELA MOLINA CIOCCI

**USO DE REDES NEURAIS PARA ANÁLISE DE DADOS NAS DISCIPLINAS DE LABORATÓRIO DE FÍSICA**

Relatório Parcial de Iniciação Didática apresentado ao Centro Universitário FEI, como parte dos requisitos do Programa PIBIC-FEI. Orientado pela Prof. Eliane de Fátima Chinaglia

São Bernardo do Campo

2024

**RESUMO**

Há muito tempo a formação em engenharia é desenvolvida apenas em sua parte técnica. Porém, o alto avanço da tecnologia implica a necessidade do aluno se tornar mais ativo em seus estudos. Analisar simulações e comparar com dados experimentais são ótimas opções para tornar o estudante protagonista de seus estudos. Com o auxílio de ferramentas e algoritmos. Esses algoritmos podem contribuir ao Laboratório de Física. Dessa forma os alunos poderão trabalhar de forma mais prática e eficiente alguns tópicos que demandam mais tempo para compreensão. Utilizando Python como linguagem de programação, Scikit-Learn como biblioteca de Rede Neural e Matplotlib como biblioteca responsável por esboçar gráficos e dendrogramas, o objetivo é criar um software intuitivo que auxilie o aluno de engenharia a entender melhor os conceitos e dados. Foi construído um trilho de ar, responsável por minimizar as forças de atrito do movimento dois corpos de massas diferentes acoplados por um fio em uma polia, juntamente com sensores responsáveis por captarem a variação da velocidade e tempo, utilizando o Data Studio. Dessa forma, o aluno conseguirá extrair os dados com melhor facilidade afim de intensificar a acurácia da sua Inteligência Artificial. Com a aquisição concluída, os estudos serão direcionados para um aprofundamento teórico e conceitual da Rede Neural, com o objetivo de obter modelos que gerem resultados cada vez mais próximos dos dados reais obtidos em experimentos.

Palavras-chave: Engenharia. Aprendizado. Rede Neural. Python. Machine Learning. Scikit-Learn. Física Básica.

**LISTA DE ILUSTRAÇÕES**

[Figura 1- Exemplo de uma rede poligonal de Thiessen 11](#_Toc175657515)

[Figura 2- Diagrama de forças do sistema 12](#_Toc175657516)

[Figura 3- Arranjo experimental do trilho de ar 14](#_Toc175657517)

[Figura 4- Programa de automação em Python 16](#_Toc175657518)

[Figura 5- Método de Elbow dos dados do projeto 19](#_Toc175657519)

[Figura 6- Análise de clusters obtidos 20](#_Toc175657520)

[Figura 7- Aproximação da rede poligonal de Thiessen do experimento 21](#_Toc175657521)

[Figura 8- Box Plot da aceleração em função de m1 22](#_Toc175657522)

[Figura 9- Box Plot da aceleração agrupada por m2 23](#_Toc175657523)

[Figura 10- Cronograma anual do projeto 25](#_Toc175657524)

**LISTA DE TABELAS**

[Tabela 1 - Variação das massas utilizadas no experimento 14](#_Toc175657359)

[Tabela 2 - Resultado do agrupamento pelo k-Means 19](#_Toc175657360)

**SUMÁRIO**

[**1 INTRODUÇÃO** 6](#_Toc175657907)

[**2 REVISÃO BIBLIOGRÁFICA** 8](#_Toc175657908)

[2.1 ANÁLISE DE AGRUPAMENTO NÃO-HIERÁRQUICO 8](#_Toc175657909)

[**2.1.1 Método k-Means** 10](#_Toc175657910)

[**2.1.2 Polígonos de Thiessen** 10](#_Toc175657911)

[2.2 REDE NEURAL 11](#_Toc175657912)

[2.3 CINEMÁTICA E DINÂMICA DE DOIS CORPOS ACOPLADOS 12](#_Toc175657913)

[**3 METODOLOGIA** 13](#_Toc175657914)

[3.1 AQUISIÇÃO DOS DADOS 14](#_Toc175657915)

[**3.1.1 Definição dos parâmetros** 15](#_Toc175657916)

[3.2 DESENVOLVIMENTO DE PROGRAMAS para a aquisição 15](#_Toc175657917)

[3.3 UTILIZAÇÃO DE BIBLIOTECAS 16](#_Toc175657918)

[**4 RESULTADOS OBTIDOS** 17](#_Toc175657919)

[4.1 DADOS ADQUIRIDOS 17](#_Toc175657920)

[4.2 MÉTODO DE ELBOW 18](#_Toc175657921)

[4.4 FORMAÇÃO DOS CLUSTERS 19](#_Toc175657922)

[**5 CONCLUSÃO PARCIAL E PRÓXIMAS ETAPAS 24**](#_Toc175657923)

[**6 CRONOGRAMA 24**](#_Toc175657924)

[**REFERÊNCIAS 26**](#_Toc175657925)

1 INTRODUÇÃO

Um dos aspectos fascinantes do campo da Inteligência Artificial (IA) é que a natureza precisa de seu assunto acaba sendo surpreendentemente difícil de definir. Para garantir uma compreensão adequada da natureza do artificial é preciso primeiramente obter uma compreensão adequada da ideia de inteligência. O que se supõe ser "artificial" sobre a IA, tem a ver com suas origens e modo de criação em surgir como um produto da invenção e engenhosidade humanas, porém de forma “evolutiva”. Coisas que são artificialmente inteligentes, em outras palavras, diferem daquelas que são naturalmente inteligentes como artefatos que possuem propriedades especiais normalmente possuídas por não-artefatos (FETZER, 1990).

O Machine Learning (ML), uma área vital e central da IA, é a técnica que melhora o desempenho do sistema aprendendo com a experiência por meio de métodos computacionais. Em sistemas de computador, a experiência existe na forma de dados, e a principal tarefa do aprendizado de máquina é desenvolver algoritmos de aprendizado que construam modelos a partir de dados. Ao alimentar o algoritmo de aprendizado com dados de experiência, obtemos um modelo que pode fazer previsões em novas observações (ZHOW, 2021).

Analisando a formação do curso de Engenharia, torna-se essencial que os futuros engenheiros adquiram, pelo menos, uma compreensão básica de como os algoritmos de IA e ML funcionam, visando o crescimento exponencial dessas tecnologias não apenas no cotidiano, mas também no mercado de trabalho. Tendo ciência dessa importância, o Laboratório de Física irá contribuir a fim de oferecer aos futuros engenheiros uma base sólida nos conceitos de máquinas inteligentes (MEI/CNI, 2018).

O objetivo do projeto é apresentar a um aluno de engenharia os conceitos básicos de uma IA interligados aos conceitos de física, como a Segunda Lei de Newton. Além disso, este projeto também visa aproveitar a oportunidade de coletar e analisar dados durante as aulas de laboratório para proporcionar aos alunos ingressantes a interação com algoritmos básicos de ML. Com isso, o aluno poderá aprender conhecimentos de física juntamente com os conceitos de IA e programação, destacando a importância dessa habilidade em sua formação (MITCHELL, 1997).

Neste projeto serão explorados algoritmos de ML tanto não supervisionados (agrupamento) quanto supervisionados (Rede Neural) para analisar dados obtidos em experimentos realizados pelos alunos no laboratório de física. A análise de agrupamento não hierárquico, também conhecido como partição, é uma técnica de aprendizado não supervisionado que busca dividir um conjunto de dados em um número pré-definido de grupos (clusters) de forma que os objetos dentro de cada grupo sejam mais semelhantes entre si do que com os objetos de outros grupos (FÁVERO, 2022). Redes neurais artificiais são modelos computacionais inspirados no sistema nervoso humano, capazes de aprender e tomar decisões complexas a partir de dados (GIORDANI, 2020).

2 REVISÃO BIBLIOGRÁFICA

Visando que o objetivo do projeto consiste em enriquecer o ensino introdutório de Física através da IA, é essencial refinar os conhecimentos sobre os temas trabalhados neste projeto. O tópico selecionado para a criação de uma simulação foi a cinemática e a dinâmica das partículas.

2.1 ANÁLISE DE Agrupamento NÃO-HIERÁRQUICO

A análise de agrupamento não hierárquico, também conhecido como partição, é uma técnica de aprendizado não supervisionado que busca dividir um conjunto de dados em um número pré-definido de grupos (clusters) de forma que os objetos dentro de cada grupo sejam mais semelhantes entre si do que com os objetos de outros grupos (FAVERO).

Os dois algoritmos de agrupamento não hierárquicos mais famosos são o k-Means e o k-Medoids. Eles diferem na definição dos protótipos de cluster. Em particular, os protótipos k-Means, chamados centroides, são definidos como os valores médios das unidades atribuídas aos clusters, enquanto os protótipos k-Medoids, chamados medoids, identificam as unidades observadas mais representativas para cada cluster (GIORDANI).

Para criar um agrupamento, é necessário definir a quantidade inicial de grupos e suas respectivas centroides. O objetivo desse passo é dividir os itens do banco de dados em K grupos, de modo que todos os itens constados em cada grupo estejam mais próximos entre si se comparados a qualquer outro item pertencente a um diferente. Para isso, os itens precisam ser alocados arbitrariamente nos grupos para que assim possa ser calculada os centroides através da média aritmética do grupo em relação a cada item (FAVERO).

Deve-se selecionar determinado item que se encontra mais próximo de um centroide e realocá-lo nesse grupo, fazendo assim com que outro grupo perca esse item e, portanto, recalcular os centroides do grupo que recebeu e do grupo que perdeu o item. Esse passo deve ser feito até que não seja mais possível realocar nenhum item por proximidade (FAVERO).

Definido um conjunto d-dimensional de n pontos de dados como os dados a serem agrupados. Após, defina um conjunto d-dimensional de k centros como a quantidade de cluster que o algoritmo deva refinar. (HAMERLY, 2002).

Uma função de adesão define a proporção de pontos de dados que pertence ao centro com restrições e . Alguns desses algoritmos usam uma função de associação chamada rígida, que significa que , enquanto outros utilizam uma função de associação chamada suave, que significa que . O k-Means utiliza da função de associação rígida para realizar seus cálculos. (HAMERLY, 2002).

A função rígida torna possível muitas otimizações computacionais que não afetam a precisão do algoritmo.

Uma função de peso define o quanto de influência o banco de dados de pontos tem para computar novamente os parâmetros centrais na próxima iteração, com restrição . Com isso, podemos obter um modelo geral de clusterização iterativa baseada em centroides (HAMERLY, 2002):

1. Inicialize o algoritmo com a quantidade de centroides desejada;
2. Para cada ponto no banco de dados X, calcular sua associação em cada centroide e seu peso ;
3. Para cada centroide , recalcule todos os pontos do banco de dados X de acordo com suas associações e pesos, descrita na equação 1:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

1. Repita as etapas 2 e 3 até que o algoritmo tenha convergência.

2.1.1 Método k-Means

O algoritmo k-Means separa os dados em k grupos, tendo como solução um conjunto de k grupos, cada um localizado no centroide dos dados nos quais é o centro mais próximo. Isso resulta em um particionamento do espaço de dados em [células Voronoi](https://en.wikipedia.org/wiki/Voronoi_cell), que será explicado logo após esse tópico. O agrupamento de k-Means minimiza as variâncias dentro do agrupamento ([distâncias euclidianas ao quadrado),](https://en.wikipedia.org/wiki/Squared_Euclidean_distance) mas não as distâncias euclidianas regulares. A equação 2 informa qual equação o método utiliza em seu algoritmo (HAMERLY, 2002).

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

2.1.2 Polígonos de Thiessen

Os polígonos de Thiessen (também chamados de Dirichlet ou Voronoi) é uma partição de um plano em regiões próximas a cada um de um determinado conjunto de objetos. Se os dados estiverem em uma grade quadrada regular, então os polígonos de Thiessen são todos iguais (BURROUGH, 2015). Se os dados forem espaçados irregularmente, então o resultado obtido será uma rede irregular de polígonos, demonstrada na figura 1.

Figura 1- Exemplo de uma rede poligonal de Thiessen

Gráfico, Gráfico de radar

Descrição gerada automaticamente

Fonte: “Principles of Geographical Information Systems.” 3rd ed. p-16.

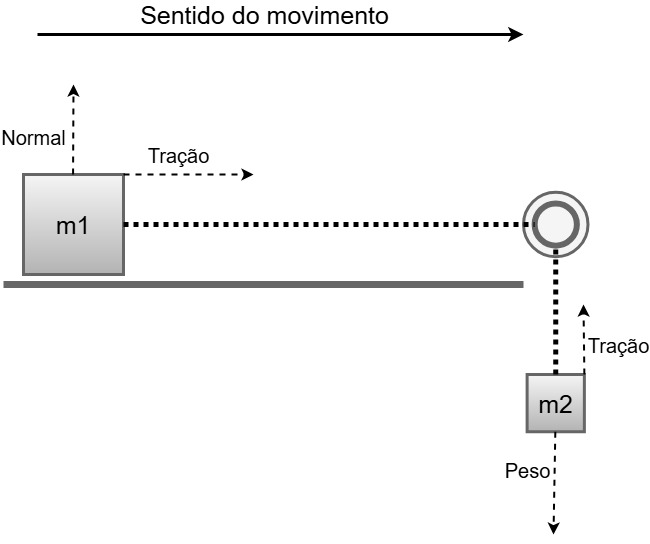
2.2 REDE NEURAL

Redes neurais artificiais são modelos computacionais inspirados no sistema nervoso humano, capazes de aprender e tomar decisões complexas a partir de dados (ZHANG). Funcionam como uma rede de neurônios artificiais interconectados, onde cada neurônio processa informações e as transmite para outros. Tecnicamente, os dados de entrada são alimentados na primeira camada, e a informação flui através das camadas ocultas até a camada de saída, onde é gerada a resposta final. Durante o treinamento, a rede é exposta a um grande conjunto de dados de treinamento, e os pesos são ajustados iterativamente utilizando algoritmos de otimização.

2.3 CINEMÁTICA E DINÂMICA DE DOIS CORPOS ACOPLADOS

Os tópicos de física básica abordados nesse projeto são “cinemática” e “dinâmica” e está totalmente relacionado com os corpos do sistema conforme mostra a figura 2 (HALLIDAY, 2016).

Figura 2- Diagrama de forças do sistema



Fonte: Gabriela Molina Ciocci

Nesse projeto, foi utilizado um trilho de ar responsável por minimizar a força de atrito, tornando-a praticamente nula. Por esse motivo, a força de atrito não está presente no diagrama.

Os dados extraídos são numéricos e foram classificados de tal forma:

: massa do corpo posicionado em um plano horizontal sem atrito;

**:** massa do corpo suspenso;

Para visar os resultados experimentais, é de extrema importância analisar a parte teoria do problema, por isso, calcular a aceleração teórica auxilia a analisar se os resultados obtidos estão sendo próximos aos resultados esperados.

Para determinar a aceleração teórica de um corpo, recorremos à Segunda Lei de Newton, um dos pilares da dinâmica. Essa lei estabelece uma relação fundamental entre três grandezas físicas: a força resultante (F) aplicada a um corpo, sua massa (m) e a aceleração (a) que ele adquire. Matematicamente, essa relação é expressa pela equação:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

resultante que atua sobre o corpo (N);

***m:*** massa do corpo (kg);

aceleração do corpo (m/s²).

Ao mencionar a Segunda Lei de Newton, fica claro que o uso desta lei no projeto é essencial, porém é necessário fazer algumas alterações na fórmula pois se trata de dois corpos acoplados sem a presença do atrito.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Após feito as devidas alterações na fórmula e manter a aceleração devidamente isolada para fins de calculá-la, é possível comparar o resultado obtido através do experimento com a aceleração medida.

3 METODOLOGIA

A primeira etapa para o desenvolvimento desse projeto consistiu na aquisição de dados para obter um banco de dados eficiente e confiável, com o intuito de treinar a IA. Foi utilizado um trilho de ar para a fim de minimizar as forças de atrito no movimento de dois corpos de massas diferentes e acoplados por um fio. Sensores ópticos posicionados estrategicamente permitiram a medida da velocidade de um dos corpos (. Com esse arranjo experimental, foi possível determinar a aceleração do sistema a partir das medidas de velocidade em função do tempo. O trilho pode ser horizontal ou inclinado, formando um ângulo. Nesta etapa, o trilho de ar foi mantido a 0°, ou seja, horizontalmente. A figura 3 apresenta o arranjo experimental do projeto.

Figura 3 - Arranjo experimental do trilho de ar

Escritório com mesa de madeira

Descrição gerada automaticamente com confiança média

Fonte: Gabriela Molina Ciocci, Bruno Arthur Basso Silva

Na tabela 1, está apresentada a variação de valores de e utilizada para a aquisição de dados do projeto.

Tabela 1 - Variação das massas utilizadas no experimento

|  |  |
| --- | --- |
|  |  |
| 0.209 | 0.004 a 0.013 |
| 0.309 | 0.014 a 0.019 |
| 0.409 | 0.02 a 0.03 |

Fonte: Gabriela Molina Ciocci

3.1 AQUISIÇÃO DOS DADOS

Para a aquisição de dados foi utilizado o programa Data Studio, uma ferramenta essencial para visualização dos dados quando se trata da obtenção de informações de velocidade e tempo de forma extremamente precisa. Ao realizar o experimento, com o movimento de um dos corpos no trilho de ar, o software registra meticulosamente em intervalos regulares de tempo e com o auxílio de quatro sensores posicionados simetricamente com a finalidade de melhorar a precisão dos resultados. Em seguida, os dados coletados foram armazenados uma planilha no Excel.

3.1.1 Definição dos parâmetros

Os dados de entrada para treinar uma IA podem ser numéricos ou categóricos, desde estejam formatados em um padrão para que a Rede Neural possa classificar os dados de forma eficaz.

Para realização do projeto, foi necessário identificar quais parâmetros são indispensáveis para que resultado obtido seja conforme o esperado. Após realizar essa análise, as variáveis escolhidas para melhor execução do algoritmo de classificação foram:

* : massa do corpo posicionado no trilho de ar
* : aceleração do corpo (m/s²)

Foram escolhidas essas variáveis pois o principal objetivo do projeto consiste em treinar a Rede Neural para poder classificar qual grupo de massas o experimento se encontra*.* Ainda não foi adicionado nenhum dado que tenha variação de ângulo, por esse motivo a variável não foi mencionada, mas futuramente ela será acrescentada tendo em vista que sua variação ocasionará mudanças drásticas na eficácia da Rede Neural.

3.2 DESENVOLVIMENTO DE PROGRAMAS para a aquisição

Para otimizar e facilitar o trabalho experimental, foi desenvolvido um programa que automatiza a adição dos dados obtidos pelo Data Studio no banco de dados no Excel utilizado a linguagem de programação Python e o auxílio da biblioteca Pandas. A figura 4 mostra o terminal integrado do software VSCode.

Figura 4 - Programa de automação em Python

Texto

Descrição gerada automaticamente

Fonte: Gabriela Molina Ciocci, Bruno Arthur Basso Silva

3.3 UTILIZAÇÃO DE BIBLIOTECAS

Para realização do projeto, foi preciso utilizar algumas bibliotecas em Python para o desenvolvimento da IA e para análise de gráficos e dendrogramas (PYTHON, 2001). São elas:

* Scikit-Learn: também conhecido como sklearn, é uma das bibliotecas mais populares em Python para aprendizado de máquina. Ela oferece uma ampla variedade de algoritmos de aprendizado de máquina, tanto para tarefas supervisionadas quanto não supervisionadas (SCIKIT-LEARN, 2007).
* Pandas: é amplamente utilizada para manipulação e análise de dados. Ela oferece estruturas de dados poderosas, como o Data Frame, que permite armazenar e manipular conjuntos de dados de forma eficiente. Além disso, é possível realizar operações de limpeza, transformação e análise estatística em dados tabulares de maneira intuitiva e eficaz (PANDAS, 2008).
* Matplotlib.pyplot: é uma sub-biblioteca do Matplotlib, que fornece uma interface para criar gráficos de alta qualidade em Python. Com ele, é possível criar uma ampla variedade de gráficos, como gráficos de linha, histogramas, gráficos de dispersão e muito mais (MATPLOTLIB, 1991).
* Seaborn: é uma biblioteca de visualização de dados que fornece uma interface de alto nível para criar gráficos estatísticos atraentes e informativos. E simplifica a criação de gráficos complexos, como diagramas de dispersão com regressão linear, box plots e mapas de calor (SEABORN, 2012).
* SciPy: é uma biblioteca usada para computação científica e técnica. Ela é construída sobre o NumPy e fornece funcionalidades adicionais para operações matemáticas, otimização, processamento de sinais, álgebra linear, integração numérica, interpolação, estatísticas, entre outros. (SCIPY, 2008).

4 RESULTADOS OBTIDOS

Após uma imersão completa no projeto, compreendendo suas nuances e fundamentos, é necessário sintetizar todos os resultados obtidos até o momento. Essa etapa é essencial para identificar tanto os aspectos positivos que impulsionam o projeto quanto os desafios que ainda precisam ser superados. Com base nessa análise criteriosa, é possível traçar um caminho mais eficiente para a continuidade do projeto, maximizando ainda mais suas chances de sucesso.

4.1 DADOS ADQUIRIDOS

Os resultados experimentais estão alinhados com as previsões teóricas, indicando que a coleta de dados está sendo realizada de forma eficiente conforme o planejado. Essa precisão na aquisição de dados é fundamental para o sucesso do treinamento da Inteligência Artificial, pois garante que o modelo seja exposto a dados de alta qualidade e representativos do fenômeno em estudo contribuindo significativamente para a construção de um banco de dados robusto. Para treinar a Rede Neural, até o momento, foram adquiridos 256 dados, tendo uma variedade grande de valores para .

4.2 MÉTODO DE ELBOW

Para determinar a quantidade ideal de clusters em um conjunto de dados, foi utilizado o método de Elbow.

O método de Elbow é um método utilizado para descobrir a quantidade ideal de clusters em um conjunto de dados. O método é chamado dessa forma pois ele sempre está em busca do ponto de inflexão na curva, assemelhando-se ao formato de um cotovelo. Esse método inicia com um único cluster e vai adicionando mais clusters gradualmente. A cada novo cluster, a qualidade da classificação é avaliada. Quando a melhora na classificação se torna muito pequena, ou seja, quando os ganhos em precisão se tornam insignificantes, o processo para. Nesse ponto, a divisão em clusters adicionais não traz mais benefícios significativos (YELLOWBRICK, 2016). A figura 5 apresenta o gráfico do método utilizado com o banco de dados adquirido através desse projeto.

Figura 5- Método de Elbow dos dados do projeto

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Fonte: Gabriela Molina Ciocci

4.4 Formação dos clusters

Após a análise do gráfico, foi observado que a quantidade mais eficaz para o projeto até o momento seria de 5 clusters. Então, foi efetuado o treinamento da IA utilizando 30% dos dados obtidos até o momento. Após seu treinamento, obtivemos os resultados do agrupamento pelo k-Means apresentados na tabela 2.

Tabela 2- Resultado do agrupamento pelo k-Means

|  |  |  |  |
| --- | --- | --- | --- |
| Cluster | Quantidade de itens por Cluster | Valor mínimo de | Valor máximo de |
| 1 | 54 | 0.017 | 0.020 |
| 2 | 59 | 0.015 | 0.021 |
| 3 | 61 | 0.009 | 0.021 |
| 4 | 20 | 0.031 | 0.031 |
| 5 | 52 | 0.004 | 0.004 |

Fonte: Gabriela Molina Ciocci

Na figura 6 é possível observar o gráfico que contém a forma como os dados foram agrupados através das fórmulas apresentadas nas seções 2, 2.1 e na figura 7 é possível analisar aproximadamente a rede poligonal de Thiessen e onde estão localizados seus centroides.

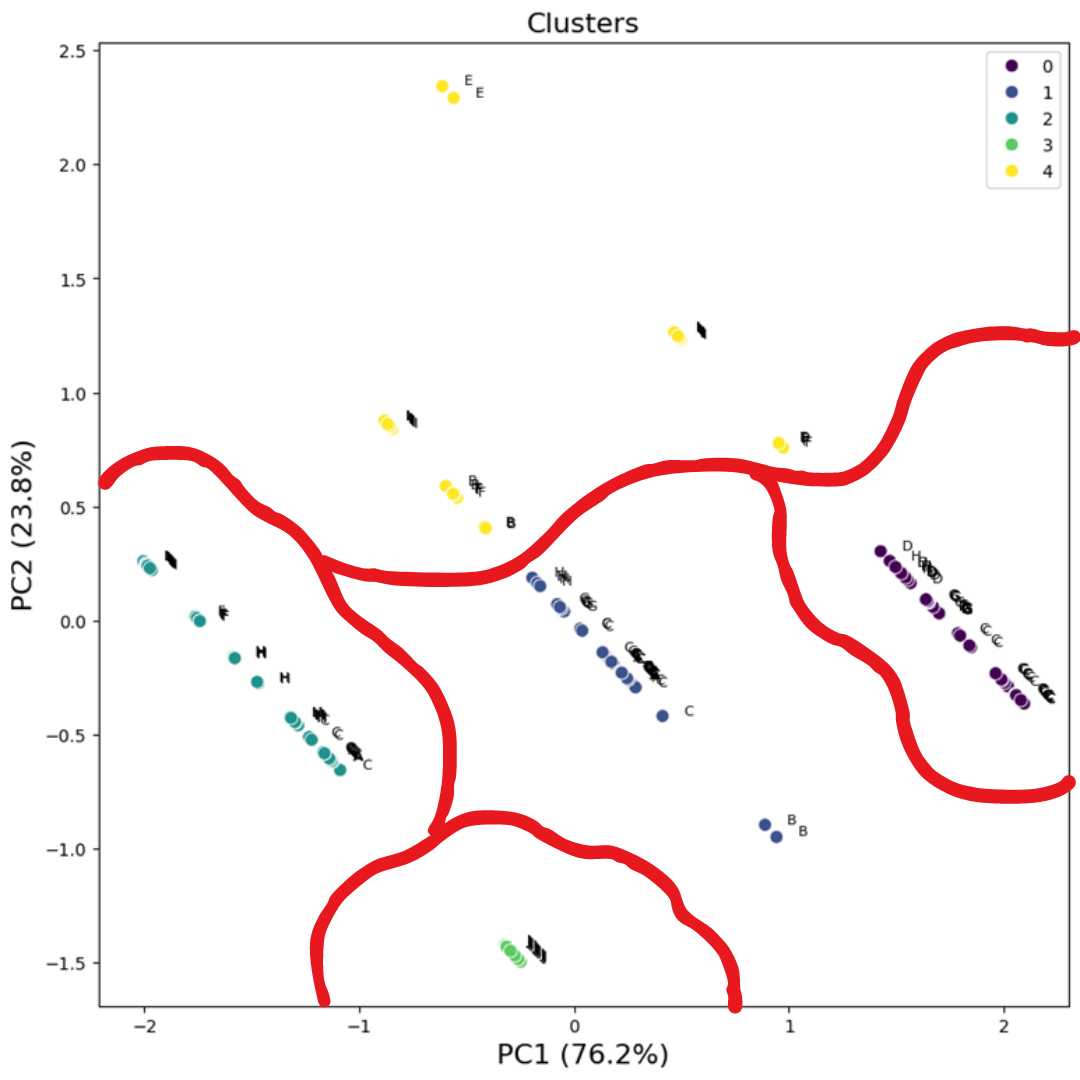
Figura 6- Análise de clusters obtidos

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Fonte: Gabriela Molina Ciocci

Figura 7- Aproximação da rede poligonal de Thiessen do experimento



Fonte: Gabriela Molina Ciocci

Analisando o gráfico, pode-se perceber que o cluster 3 apresentou um ótimo agrupamento enquanto os clusters 0, 1, 2, e 4 estão com uma grande dispersão dos valores da massa , o que indica uma dificuldade para a identificação correta da massa. Portanto, será necessário a diminuição de variabilidade de massas para melhorar posteriormente a acurácia inicial da Rede Neural.

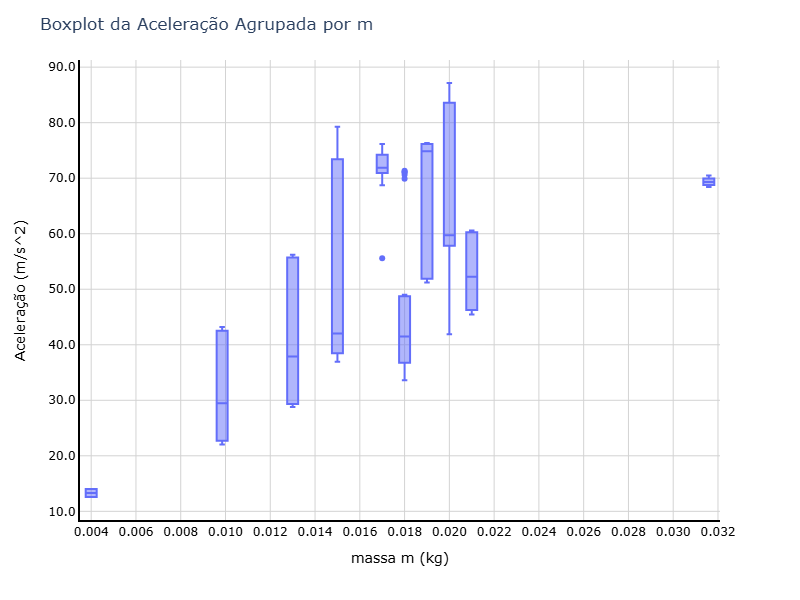
Com o intuito de aprofundar a análise da variabilidade dos dados coletados, optou-se por uma abordagem estatística utilizando o gráfico de Box Plot. Essa representação gráfica permite visualizar de forma clara a distribuição dos dados, identificando valores atípicos e comparando diferentes grupos. A seguir, são apresentados os resultados obtidos.

4.5 ANÁLISE DADOS UTILIZANDO BOX PLOT

O Box Plot, também conhecido como diagrama de caixa, é uma ferramenta visual poderosa para explorar a distribuição de um conjunto de dados. Essa representação gráfica é composta por uma caixa, que delimita a amplitude interquartil (AIQ), e por bigodes que se estendem a partir da caixa, indicando o intervalo total dos dados, excluindo os outliers. A linha horizontal dentro da caixa representa a mediana, que divide os dados em duas partes iguais. Os outliers, pontos que se encontram significativamente distantes dos demais dados, são representados por pontos ou asteriscos e podem influenciar a análise estatística. A posição da mediana em relação à caixa fornece informações sobre a simetria da distribuição: se a mediana estiver centralizada na caixa, a distribuição é considerada simétrica; caso contrário, ela é assimétrica. (SEABORN, 2012).

A figura 8 apresenta o Box Plot dos dados obtidos e nos mostra a distribuição dos valores, a amplitude e a simetria conforme a variação de . Já a figura 9 apresenta o Box Plot da aceleração em função dos valores de .

Figura 8- Box Plot da aceleração em função de m1



Fonte: Gabriela Molina Ciocci

Figura 9- Box Plot da aceleração agrupada por m2

Gráfico, Gráfico de caixa estreita

Descrição gerada automaticamente

Fonte: Gabriela Molina Ciocci

Analisando a figura 8, pode-se perceber que entre os valores de entre a variação de 0,010kg até 0,022kg houve grande dispersão de possíveis valores para a massa . Essa variabilidade trouxe dificuldade para a identificação correta dos clusters e consequentemente, a diminuição da acurácia da Rede Neural.

Uma possível solução para o problema no momento é ajustar o número de valores da variável . Ao reduzir essa quantidade, os intervalos entre os valores se tornam maiores, o que pode facilitar a interpretação dos dados.

5 CONCLUSÃO PARCIAL e próximas etapas

A coleta de dados experimentais, embora desafiadora, é crucial para o treinamento eficaz da Rede Neural. Para aprimorar o modelo, o próximo passo será otimizar os valores de massa e adquirir novos dados com o trilho de ar inclinado, com diferentes angulações.

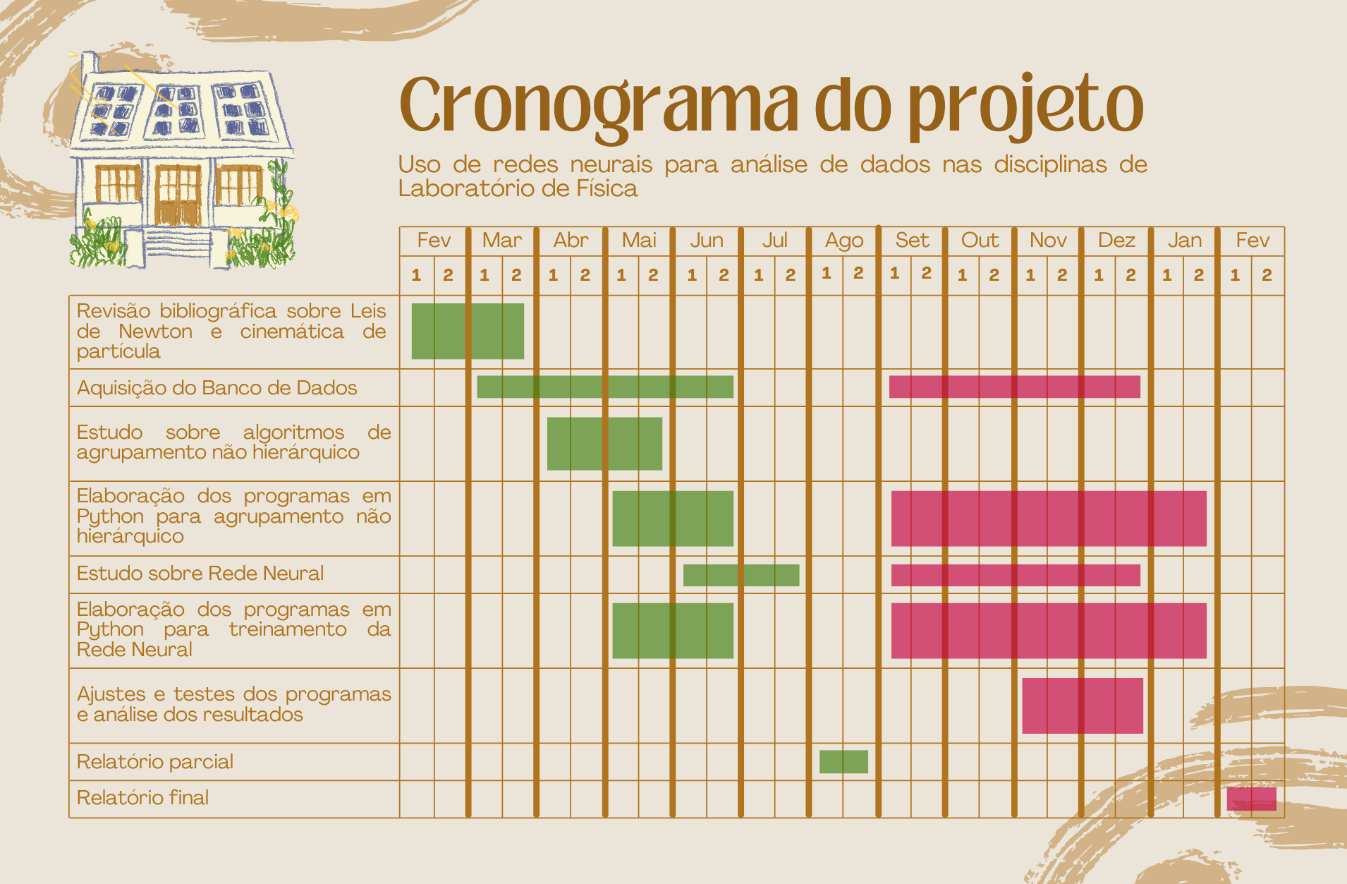
Assim, a expectativa é aumentar a quantidade dos dados, melhorar sua qualidade e obter uma melhor precisão da classificação, permitindo que a IA estime a massa com excelência a partir dos dados , ângulo e aceleração fornecidos, auxiliando nas aulas de laboratório de Física 1.

Até a conclusão do projeto, estima-se obter uma IA treinada para que em uma aula do Laboratório de Física 1 os alunos entrem com os valores obtidos e a IA retorne o valor da massa com precisão.

6 CRONOGRAMA

A figura 10 apresenta o cronograma das próximas etapas do projeto.

Figura 10- Cronograma anual do projeto



Fonte: Gabriela Molina Ciocci

Tópicos que serão realizados nos próximos meses de projeto:

* Aquisição do Banco de Dados;
* Elaboração dos programas em Python para o treinamento e teste da Rede Neural;
* Estudos mais aprofundados sobre Rede Neural;
* Ajustes e testes dos programas;
* Análise de resultados obtidos;
* Relatório final.

REFERÊNCIAS

Fetzer, J.H. (1990). What is Artificial Intelligence?. In: Artificial Intelligence: Its Scope and Limits. Studies in Cognitive Systems, vol 4. Springer, Dordrecht. Disponível em: <https://doi.org/10.1007/978-94-009-1900-6_1> . Acesso em: 22/08/2024.

Zhou, Zhi-Hua. Machine learning. Springer. 2021. Disponível em: <https://doi.org/10.1007/978-981-15-1967-3>. Acesso em: 22/08/2024.

MEI/CNI, Abenge. DIRETRIZES PARA O CURSO DE ENGENHARIA. 2018. Disponível em:

http://www.abenge.org.br/documentos/propostadcnabengemei\_cni.pdf. Acesso em: 07/03/2024.

MITCHELL, Tom M. MACHINE LEARNING. v.3 - Decision Tree Learning. 1997. Disponível em: <https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf>. Acesso em: 01/02/2024.

FÁVERO. MANUAL DE ANÁLISE DE DADOS ESTATÍSTICA E MODELAGEM MULTIVARIADA COM EXCEL, SPSS E STATA, v.9.1 – v.9.2, p. 300 – 327, 2022. Acesso em: 08/06/2024.

Giordani, P., Ferraro, M.B., Martella, F. (2020). Non-Hierarchical Clustering. In: An Introduction to Clustering with R. Behaviormetrics: Quantitative Approaches to Human Behavior, vol 1. Disponível em: <https://doi.org/10.1007/978-981-13-0553-5_3> . Acesso em: 11/08/2024.

Hamerly, Greg; Elkan, Charles (2002). ["Alternatives to the *k*-means algorithm that find better clusterings"](http://people.csail.mit.edu/tieu/notebook/kmeans/15_p600-hamerly.pdf) (PDF). Proceedings of the eleventh international conference on Information and knowledge management (CIKM). Acesso em: 23/08/2024.

Burrough PA, McDonnell R, Lloyd C. Principles of Geographical Information Systems. 3rd ed. Oxford University Press; 2015, p. 160 – 161. Acesso em: 23/08/2024

ZHANG, Z. (2018). Artificial Neural Network. In: Multivariate Time Series Analysis in Climate and Environmental Research. Disponível em: <https://doi.org/10.1007/978-3-319-67340-0_1> . Acesso em: 11/08/2024

HALLIDAY, David; RESNICK, Robert; WALKER, Jearl. FUNDAMENTOS DE FÍSICA, VOLUME 1: MECÂNICA. 10. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora LTDA, 2016. v. 1. Acesso em: 25/05/2024.

PYTHON. **PYTHON 3.12.4 DOCUMENTATION**. 2001. Disponível em:

<https://docs.python.org/3/>. Acesso em: 14/05/2024.

PANDAS. PANDAS DOCUMENTATION. 2008. Disponível em: <https://pandas.pydata.org/docs/>. Acesso em: 07/06/2024.

SCIKIT-LEARN. **SCIKIT-LEARN.ORG**. 2007. Disponível em: <https://scikit-learn.org/stable/about.html#history>. Acesso em: 07/06/2024.

MATPLOTLIB. **MATPLOTLIB.PYPLOT**. 1991. Disponível em: <https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html>. Acesso em: 07/06/2024.

SCIPY, api. **SCIPY DOCUMENTATION**. 2008. Disponível em: <https://docs.scipy.org/doc/scipy/>. Acesso em: 07/06/2024.

SEABORN. **SEABORN DOCUMENTATION.** 2012. Disponível em: <https://seaborn.pydata.org/tutorial.html/> . Acesso em: 05/08/2024.

YELLOWBRICK. **ELBOW METHOD.** 2016. Disponível em: <https://www.scikit-yb.org/en/latest/api/cluster/elbow.html>. Acesso em: 08/06/2024.