

Introduzindo aprendizado de máquina em cursos de física: o caso do rolamento no plano inclinado

Inserting machine learning in physics courses: the case of rolling on an inclined plane

H. Ferreira^{*1,2}, E.F. Almeida Junior³, W. Espinosa-García⁴, E. Novais¹,
J.N.B. Rodrigues¹, G.M. Dalpian¹

¹Universidade Federal do ABC, Centro de Ciências Naturais e Humanas, Santo André, SP, Brasil.

²Faculdade de Informática e Administração Paulista, São Paulo, SP, Brasil.

³Universidade Federal do Oeste da Bahia, Centro das Ciências Exatas e das Tecnologias, Barreiras, BA, Brasil.

⁴Universidad de San Buenaventura-Medellín, Facultad de Ingenierías, Medellín, ANT, Colombia.

Recebido em 27 de julho de 2022. Revisado em 13 de setembro de 2022. Aceito em 12 de outubro de 2022.

Em uma ciência cada vez mais orientada a dados, o uso de métodos computacionais inteligentes é progressivamente indispensável. Neste contexto, torna-se importante expor os estudantes de graduação em física às metodologias de inteligência artificial e aprendizado de máquina. Neste trabalho propomos uma forma de utilizar tais métodos na física, resolvendo o problema didático do rolamento em um plano inclinado. Para tanto, introduzimos os principais conceitos das técnicas de aprendizado de máquina e realizamos medidas do tempo que diferentes objetos (aro, disco e esfera) levam para percorrer uma certa distância, de acordo com a altura inicial e o ângulo de inclinação do plano. Com estes dados, aplicamos métodos de classificação capazes de prever o objeto que rolava sobre o plano com acurácia de 83%, e métodos de regressão capazes de prever a velocidade média do objeto com erro absoluto médio de 1.4 cm s^{-1} . Mostramos também que este modelo didático é instrutivo pois possibilita uma comparação direta com modelos físicos tradicionais e serve como exemplo introdutório para a discussão do que significa ensinar física para o computador.

Palavras-chave: Inteligência Artificial, Aprendizado de Máquina, Plano Inclinado.

In an increasingly data-oriented science, the use of automatic computational methods is progressively indispensable. In this context, it becomes important to expose undergraduate physics students to artificial intelligence and machine learning methodologies. In this work we propose a way to use such methods in physics, solving the didactic problem of rolling on an inclined plane. We introduce the main concepts of machine learning techniques and measure the travel time of different objects (rim, disk and sphere) for an initial height and tilt angle. Based on these data, we used classification models capable of predicting the object that was dropped with an accuracy of 83%, and regression models which were able to predict the average speed of the object that was rolled with mean absolute error of 1.4 cm s^{-1} . We also show that this didactic model is instructive because it allows a direct comparison with physical models and serves as a discussion of the meaning of teaching physics to the computer.

Keywords: Artificial Intelligence, Machine Learning, Inclined Plane.

1. Introdução

Algoritmos computacionais relacionadas à inteligência artificial já se tornaram corriqueiros no nosso dia a dia. Eles são responsáveis por nos sugerir o filme que vamos assistir, a música que iremos ouvir ou o livro que iremos comprar [1]. Da mesma forma, nas ciências naturais, o uso de técnicas associadas ao *big data* também tem se tornado habitual [2, 3]. Do ponto de vista do mercado de trabalho, os profissionais formados na área de física e engenharia também começam a encontrar lugar mais frequente em posições relacionadas à ciência de dados. Esse contexto contrasta com o fato de que a maioria dos cursos de física ainda não possuem

esses temas incorporados no seu currículo. Disciplinas de Física Computacional, muito comuns em cursos de graduação em física, normalmente focam em temas mais tradicionais, como a solução de equações diferenciais, métodos de integração e métodos de Monte Carlo (ver Apêndice A para uma discussão mais detalhada sobre as ementas das disciplinas de física computacional em universidades federais brasileiras). Um bom exemplo de inserção de inteligência artificial em cursos de graduação pode ser encontrado na proposta pedagógica da Ilum (CNPEM), que é um bacharelado de ciência e tecnologia, e contempla em todos os semestres (desde o primeiro) disciplinas sobre programação e ciência de dados, com ênfase em aprendizado de máquina. Como o foco do curso é na interdisciplinaridade, os conceitos aprendidos de programação e ciência de dados são

* Endereço de correspondência: hfsantos@ufabc.edu.br

aplicados nas ciências biológicas e da matéria [4]. Neste artigo oferecemos um exemplo concreto de como inserir técnicas de inteligência artificial em cursos de graduação em física, focando no problema do rolamento em um plano inclinado. A utilização dessas metodologias em um exemplo conhecido e bem estabelecido potencializará sua aplicação em outros exemplos mais complexos e exploratórios.

O problema do plano inclinado foi inicialmente analisado vários séculos atrás. Em 1638, no seu livro *Discursos sobre as Duas Novas Ciências*, Galileo Galilei apresentou um experimento no qual rolou uma esfera de bronze sobre uma peça de madeira inclinada em relação ao chão [5]. Dessas observações experimentais, Galileo concluiu que a distância percorrida pela esfera é proporcional ao quadrado do tempo transcorrido durante o movimento. Hoje esse conceito é amplamente difundido em capítulos de cinemática de livros didáticos de ensino médio e graduação [6–8], normalmente sob o título de movimento retilíneo uniformemente variado.

Apesar de sua simplicidade, o experimento do plano inclinado é um problema de física fundamental de grande riqueza conceitual. Não só é possível estabelecer leis do movimento de translação dos corpos, como também propriedades da rotação, como o momento de inércia. É possível verificar a conservação de energia quando as superfícies são polidas, como no caso de Galileo, ou estabelecer a perda de energia quando usado um plano rugoso com atrito. Neste caso, o experimento do plano inclinado também pode ser usado para estabelecer os coeficientes de atrito estático e dinâmico entre a superfície e o corpo de prova [9, 10]. Sendo um experimento tão rico, ele é comumente explorado em exercícios e nos laboratórios didáticos dos cursos de física e engenharia.

A Física Computacional, por sua vez, demorou mais de trezentos anos depois de Galileo para surgir como área do conhecimento. Tradicionalmente, os livros-texto dessa disciplina focam em métodos de cálculo numérico para solução de problemas matematicamente bem estabelecidos, como soluções de equações diferenciais e análise de Fourier [11–13]. Por outro lado, os avanços na ciência da computação suprimiram novos métodos computacionais para solução de problemas com abordagens muito diferentes daquelas comumente exploradas na física. Em especial, a inteligência artificial e a mineração de dados tem se mostrado ferramentas capazes de acrescentar novos contornos aos paradigmas de fazer ciência [14]. Da mesma maneira que Galileo pode ser considerado como um representante arquetípico da transição entre a Ciência Empírica e a Ciência Teórica, a inteligência artificial é a ferramenta arquetípica da Ciência Orientada à Dados, o paradigma onde grandes volumes de dados podem ser medidos e/ou simulados, armazenados, acessados e processados para gerar conhecimento. Neste contexto, muitos livros de Física Computacional ainda se concentram somente no paradigma de Ciência Computacional Clássica (focada em simulações), deixando de lado os recentes

avanços interdisciplinares do uso de inteligência artificial e outras técnicas da ciência da computação para a física.

Neste trabalho exploramos essa questão demonstrando que os conceitos de inteligência artificial, particularmente do subcampo do Aprendizado de Máquina, já podem ser incluídos em disciplinas de Física Computacional e que este ferramental estará cada vez mais presente no dia a dia de cientistas das mais variadas áreas. Para isso, introduzimos o uso de métodos de Aprendizado de Máquina no problema do plano inclinado de Galileo, demonstrando como podemos executar as tarefas básicas de regressão e classificação para um problema fundamental de física.

Para apresentar esses conceitos, inicialmente discutimos, na Seção 1.1, o modelo físico do plano inclinado. Já na Seção 1.2 é feita uma introdução aos principais conceitos do Aprendizado de Máquina. Como este é um tema complexo, não detalharemos ou deduziremos as equações relacionadas a essas metodologias, mas ofereceremos referências nas quais essas informações podem ser encontradas. Nas Seções 2.1 e 2.2 são apresentadas as metodologias experimentais e computacionais utilizadas neste trabalho, respectivamente. O artigo se encerra com a apresentação dos principais resultados junto com a discussão sobre os principais problemas que podem ser observados na utilização desses métodos. As medidas realizadas envolvem rolar três objetos diferentes (aro, cilindro e esfera) em um plano inclinado, realizando medidas de tempo, ângulo e altura da qual o objeto foi largado. Munidos desses dados, escrevemos códigos de classificação que permitem prever qual objeto foi rolado e códigos de regressão que permitem prever a velocidade média dos objetos. Todos os dados medidos e utilizados neste trabalho, bem como os códigos que foram desenvolvidos para executar as simulações e obter os gráficos estão disponíveis como material suplementar.

1.1. Rolamento em plano inclinado

O problema do rolamento de um objeto sobre um plano inclinado é representado na Figura 1. Os parâmetros relevantes para o experimento são a distância percorrida pelo objeto, s , o ângulo de inclinação do plano, θ , e o tempo de descida do objeto, t , devido a aceleração da gravidade, $g \approx 9.8 \text{ m s}^{-2}$.

Do ponto de vista físico o objeto é caracterizado por sua massa, m , e seu momento de inércia I , que no caso de objetos com simetria radial é

$$I = \int_C r^2 dm = \beta m R^2, \quad (1)$$

onde R é o raio dos três objetos que estamos considerando (esfera, disco e aro). Para a esfera maciça, $\beta = \frac{2}{5}$; para a casca esférica, $\beta = \frac{2}{3}$; para o disco ou cilindro, $\beta = \frac{1}{2}$; e para o aro, $\beta = 1$.

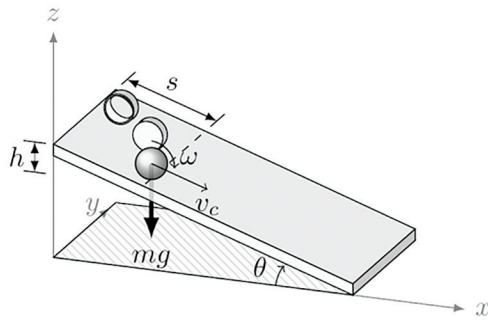


Figura 1: Ilustração do rolamento de um aro, um disco e uma esfera sobre um plano inclinado de um ângulo θ com relação ao plano horizontal. Na esfera são destacadas variáveis físicas de interesse (análogas para os outros objetos), como a força peso mg , a velocidade linear v_c do centro de massa, a velocidade angular ω de giro em relação ao centro de massa e a distância s percorrida sobre a superfície a partir da origem do movimento a uma altura $h = s \sin \theta$.

Em excelente aproximação podemos considerar a conservação de energia e escrever que:

$$mgh = \frac{1}{2}mv^2 + \frac{1}{2}I\omega^2, \tag{2}$$

onde a parcela à esquerda representa a energia potencial gravitacional inicial do objeto e a parcela à direita, a energia cinética translacional e rotacional final do objeto.

Assumindo que o objeto rola sem deslizar, a velocidade linear do centro de massa, v_c , e a velocidade tangencial, $v_t = R\omega$, são exatamente iguais, $v_c = v_t = R\omega$ (onde ω é a velocidade angular de rotação do objeto). Assim podemos reescrever a Eq. 2 como:

$$mgh = \frac{1}{2}mv^2(1 + \beta), \tag{3}$$

e a velocidade máxima final v_f alcançada ao final do percurso s para uma altura inicial h será:

$$v_f = \sqrt{\frac{2gh}{1 + \beta}}. \tag{4}$$

Para uma aceleração constante, a velocidade final v_f é exatamente o dobro da velocidade média v_{med} no percurso (uma demonstração disso é fornecida no Apêndice B do material suplementar), de forma que podemos concluir que:

$$v_{med} = \frac{1}{2}\sqrt{\frac{2gh}{1 + \beta}}. \tag{5}$$

A Eq. 5 é, portanto, um modelo analítico preditivo do comportamento cinemático de objetos de massa uniformemente distribuída e simetria radial, que rolam sem deslizar sobre um plano inclinado. A princípio, com essa equação, dado (h, θ, β) podemos prever v_{med} , ou, dado (h, θ, v_{med}) prever β . Usaremos exatamente

essas duas possibilidades para explorar a aplicação de aprendizado de máquina supervisionado para prever v_{med} (tarefa de regressão) e prever o objeto, o que é análogo a prever β (tarefa de classificação).

Contudo, tal formulação não considera um parâmetro fenomenológico importante para o modelo físico, o coeficiente de atrito, μ , entre o objeto e a superfície do plano (que como depende da área de contato também é função da forma do objeto). É o atrito entre o objeto e a superfície que produz o torque que faz o objeto entrar em rotação. Se o atrito fosse zero, o objeto simplesmente deslizaria. No limite oposto, o objeto rola sem deslizar, quando a velocidade do centro de massa é igual a velocidade tangencial da superfície do objeto. A situação experimental em geral está entre esses dois limites físicos. Podemos modelar isso através de um coeficiente f , tal que $v_t = fv_c$, de maneira que $f = 1$ corresponde a situação na qual o objeto rolou sem deslizar, e $0 < f \leq 1$, a situação de rolar e deslizar concomitantemente. Dessa forma, a Eq. 3 poderia ser reescrita como:

$$mgh = \frac{1}{2}mv_c^2(1 + \beta f^2). \tag{6}$$

Lembrando que $h = s \sin(\theta)$ e que $v_c = 2v_{med} = 2\frac{s}{t}$:

$$\beta f^2 = \frac{gt^2 \sin(\theta)}{2s} - 1. \tag{7}$$

Por outro lado, usando a segunda lei de Newton, $F = ma_c$ e $\tau = I\frac{a_t}{R}$, as acelerações do centro de massa, a_c , e tangencial, a_t , do corpo girante que está descendo o plano inclinado são calculadas por:

$$a_c = g \sin(\theta) - \mu g \cos(\theta), \tag{8}$$

$$a_t = \frac{\mu g}{\beta} \cos(\theta). \tag{9}$$

Logo, as velocidades podem ser expressas por:

$$v_c = a_c t = gt \sin(\theta) - \mu g t \cos(\theta), \tag{10}$$

$$v_t = a_t t = \frac{\mu g t}{\beta} \cos(\theta). \tag{11}$$

Então o coeficiente f é:

$$f = \frac{1}{\beta} \frac{\mu}{\tan(\theta) - \mu}. \tag{12}$$

Substituindo a Eq. 12 na Eq.7 e usando novamente a equação para a_c , encontramos:

$$\gamma = \frac{\mu^2}{\beta} = \frac{2sgt^2 \sin(\theta) - 4s^2}{g^2 t^4 \cos^2(\theta)}. \tag{13}$$

A Eq. 13 é um modelo que relaciona a forma do objeto e o atrito característico dele com a superfície do plano inclinado com os parâmetros experimentais (s, θ, t) para uma condição arbitrária que pode misturar os movimentos de rolamento e deslizamento do objeto sobre o plano.

Para uma abordagem diferente através de dinâmica Newtoniana, na qual o atrito é discutido explicitamente, recomendamos os trabalhos de Chaplin e Miller [9] e Cross [10]. Já uma discussão por mecânica Lagrangiana assumindo o vínculo de movimento é feita no Apêndice B deste trabalho.

1.2. Aprendizado de máquina

A área de Inteligência Artificial tem desenvolvido diferentes abordagens para a solução de problemas complexos através do uso de computadores. Dentre essas abordagens, o Aprendizado de Máquina visa desenvolver algoritmos que aprendam a generalizar soluções através de uma grande quantidade de exemplos sobre o problema. O termo é normalmente atribuído a Arthur Samuel, pioneiro nessa abordagem, que o define como a ciência de programar computadores para que eles aprendam através da experiência, eliminando assim a necessidade de uma programação detalhada [15]. Tais algoritmos, ditos de aprendizado, são, em geral, agnósticos ao tipo de problema resolvido, dependendo apenas dos dados fornecidos para encontrar uma solução.

Apesar dessa virtual universalidade dos algoritmos, certas limitações e considerações prévias devem ser ponderadas para cada tipo de problema. Decidir os dados a serem coletados e como representá-los no computador é de fundamental importância, exigindo conhecimentos específicos sobre o domínio do problema e sobre o funcionamento dos algoritmos utilizados. Na Figura 2 (a) apresentamos uma metodologia de trabalho em ciência de dados e inteligência artificial.

A representação dos dados é a representação do conhecimento que temos sobre o problema na forma de uma estrutura computacional. Por exemplo, informações visuais coloridas podem ser representadas em arquivos de imagens *bitmap*, que, de maneira simplificada, são estruturas de dados compostas por três matrizes

de números inteiros, normalmente com cada elemento podendo ir de 0 a 255 de intensidade da cor, cada matriz representando uma cor do modelo aditivo RGB (vermelho, verde e azul), e cujas dimensões de largura e altura são exatamente a largura e a altura em *pixels* da imagem armazenada. Cada tupla ordenada (R, G, B) de números inteiros associada a um elemento i, j das matrizes é denominada *pixel* da imagem. Este intrincado exemplo serve para mostrar o grau de sofisticação que uma representação de dado pode ter. Um algoritmo de aprendizado de máquina que opera sobre imagens deve ser construído de tal maneira a entender a representação aqui discutida ou a representação da imagem deve ser transformada para um formato compreensível ao padrão de entrada do algoritmo. Existem muitas outras formas de representar informação e conhecimento em computadores, como dicionários (ontologias), grafos e tabelas. Neste trabalho focamos na utilização de dados representados na forma tabular.

Já a modelagem envolve a escolha e uso de algoritmos de aprendizado de máquina adequados para o tipo de problema que se quer resolver e com os dados disponíveis (quantidade, qualidade e representação). Os algoritmos podem se referir a modelos descritivos ou redutivos [16]. Nesta etapa, além da escolha do algoritmo adequado, é feito o ajuste fino de seus hiperparâmetros, isto é, são configurados parâmetros internos de funcionamento do algoritmo que são independentes do problema a ser tratado, mas cujos valores podem ser otimizados para a solução pretendida. Quando se tem dados suficientes, é comum separar parte deles somente para realizar a otimização dos hiperparâmetros – este conjunto separado de dados é chamado de conjunto de validação [17, 18]. Por outro lado, para a escolha do algoritmo é necessário compreender as estratégias usadas e as tarefas que podem ser realizadas. Na Figura 2 (b) são apresentadas as principais estratégias de aprendizado: para a *abordagem supervisionada*, os

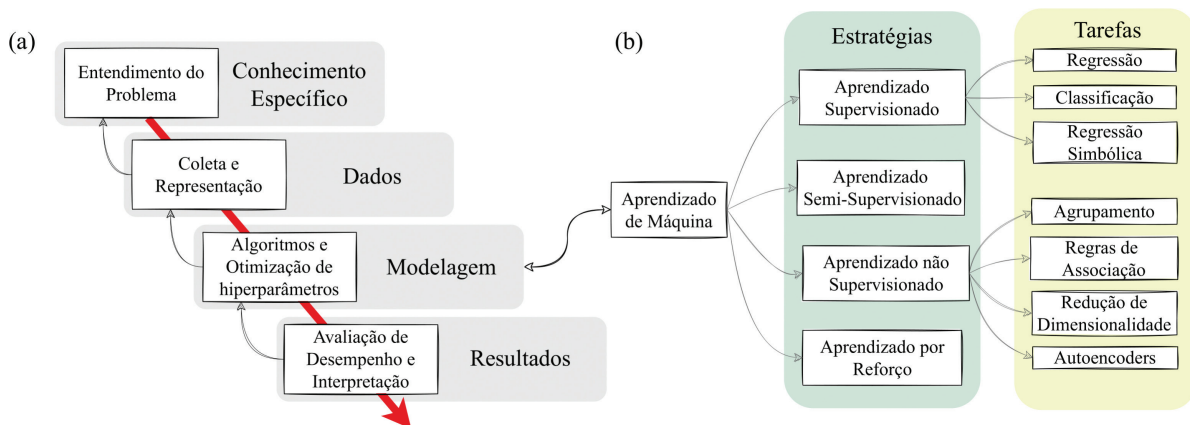


Figura 2: Aprendizado de Máquina: (a) Etapas para a utilização de técnicas de aprendizado de máquina (a seta vermelha simboliza o fluxo direto de trabalho e as setas em preto, o retorno às etapas anteriores visando o aperfeiçoamento); (b) Particionamento das técnicas de aprendizado de máquina que são usadas na etapa de modelagem (a), separadas de acordo com as estratégias de aprendizado (em verde) e o tipo de tarefa resolvida pelo algoritmo (em amarelo).

dados utilizados devem ser instâncias de um problema contendo uma resposta conhecida, denominada rótulo; no caso *não supervisionado*, não existem informações sobre a resposta esperada; métodos de *aprendizado por reforço* também não possuem rótulos para os exemplos, entretanto possuem um sinal de feedback que deve ser otimizado durante cada interação de coleta de dados; já no *aprendizado semi-supervisionado*, são utilizados métodos supervisionados e não supervisionados em série ou paralelo para realizar a tarefa desejada.

Das tarefas que os algoritmos podem resolver (Figura 2 (b)): a *regressão* envolve descobrir uma forma de relacionar a entrada X com a saída Y , sendo ambas, em geral, variáveis numéricas; na *classificação*, apesar da entrada ser numérica (contínua ou discreta), a saída é discreta ou categórica (um número ou palavra que define a classe alvo)¹; na *regressão simbólica* são fornecidas as variáveis numéricas de entrada e saída do problema, de forma que a saída do algoritmo é a expressão matemática que melhor descreve a relação entre os dados; o *agrupamento* visa identificar grupos nos dados contínuos fornecidos, devolvendo um rótulo discreto; nas *regras de associação*, pretende-se encontrar correlações entre grupos de objetos de entrada, que são dados não numéricos (*string* ou objetos); a *redução de dimensionalidade* visa diminuir a representação de um dado de entrada, de um número N de variáveis de entrada para um número L de variáveis de saída, onde $L \leq N$; já os *autoencoders* convertem uma representação de entrada em uma nova representação de saída, com dimensionalidades iguais ou não.

Por fim, é realizada a etapa de análise de resultados representada na Figura 2 (a), na qual são usadas métricas de desempenho específicas para cada classe de algoritmo e tarefa realizada (também chamada de etapa de *teste* no caso supervisionado). Além de verificar se os algoritmos possuem desempenho satisfatório para a solução do problema, é nessa etapa que são feitas as interpretações de resultados, seja para a geração de conhecimento extraído com o auxílio dessas técnicas ou para a compreensão das decisões tomadas pelos algoritmos. É importante perceber que por serem algoritmos estatísticos, as soluções encontradas por métodos de aprendizado de máquina raramente possuem precisão de 100% (em geral, é uma boa prática duvidar de soluções com tais precisões por normalmente estarem associadas a erros de uso). Por isso tais técnicas são comumente usadas em cenários onde não se conhecem soluções exatas e as performances obtidas são comparadas as melhores soluções vigentes (chamadas de modelo base).

¹ Em todos os casos discutidos, técnicas de pré-processamento da entrada podem ser usadas para adequar entradas simbólicas para numéricas, como o *Label Encoding* e o *One Hot Encoding*, etapa essa denominada de transformação de dados [16]. Algoritmos de Árvore de Decisão podem ser diretamente implementados para usarem dados categóricos como entrada.

A característica estatística desses algoritmos requer que os dados sejam exemplos de um mesmo problema. Matematicamente, podemos representar um exemplo ou instância por uma tupla $(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$, onde as variáveis denotadas por x representam informações sobre o problema (em inglês, *features*) e as variáveis denotadas por y representam soluções observadas (saídas). Neste caso, a instância tem n entradas e m saídas. Podemos simplificar nossa abordagem para os problemas que apresentam uma única saída, isto é, $(x_1, x_2, \dots, x_n, y)$. Além disso, cada tupla é apenas uma instância de um problema sobre o qual observamos k exemplos, que podemos estruturar na forma de uma tabela com k linhas, ilustrada na Figura 3 (a).

Assumindo que estabelecemos nossa representação tabular por k tuplas, na Figura 3 (b) apresentamos um esquema de uso de algoritmos supervisionados. A caixa denotada por IA representa um algoritmo de aprendizado de máquina que possui parâmetros internos de funcionamento (hiperparâmetros). Esse algoritmo de IA recebe dados de entrada e gera dados de saída, a princípio, como uma caixa-preta. O uso do algoritmo se dá em duas etapas: na primeira etapa, denominada etapa de treinamento, são fornecidos exemplos de um problema que se quer modelar; na segunda etapa, denominada de teste, o algoritmo treinado é testado e seu desempenho é comparado com o modelo base e o desempenho de outros algoritmos para a mesma tarefa.

Para realizar o treinamento, selecionamos um conjunto de exemplos da nossa tabela de observações e fornecemos para o algoritmo os pares entrada/saída. Cada algoritmo de aprendizado supervisionado utilizará uma tática diferente, que em linhas gerais, envolve um aprendizado estatístico de minimização de uma função custo (*loss function*) que calcula a diferença entre o valor estimado y_{pred} (usando as entradas X_{train}) e as saídas fornecidas y_{train} . Uma vez que a otimização da função custo se encerra sobre os exemplos de treinamento fornecidos, o algoritmo está pronto para ser usado como um modelo preditivo.

Uma questão importante que surge é se o algoritmo tem generalidade suficiente para encontrar uma resposta plausível para novos dados de entrada, quando dados de treinamento e teste são amostras da mesma região do espaço do problema. Para testar isso, uma parte das observações que não foram usadas para treinamento são usadas agora na etapa de teste. Cada saída do algoritmo y_{pred} (usando as entradas X_{test}) é comparada com a resposta verdadeiramente observada y_{true} . Essas comparações são feitas usando métricas estabelecidas para cada tipo de tarefa e representam a qualidade final do algoritmo em solucionar o problema proposto. É importante notar que, caso sejam usados os mesmos dados de treinamento e teste, então o algoritmo apenas irá gerar a saída já observada (e otimizada na etapa de treino). Este problema é um tipo de sobre-ajuste (*overfitting*), quando observamos um bom desempenho

do algoritmo apenas sobre o mesmo conjunto de dados usado no treinamento, e um desempenho ruim quando são fornecidos exemplos novos. Diz-se que o modelo treinado não tem capacidade de generalização por ter aprendido a variância contida em um único conjunto de dados. O *overfitting* também pode ocorrer quando o modelo é demasiado complexo, capturando todos os nuances dos dados, como quando usamos uma função polinomial de grau elevado para realizar o ajuste (*fitting*) dos dados.

Outro problema que pode ocorrer é o sub-ajuste (*underfitting*). Neste caso o modelo é incapaz de capturar o padrão dos dados. Isto pode ocorrer pois o modelo é demasiado simples (por exemplo, uma função linear para descrever dados não lineares), ou porque não foram fornecidos dados suficientes para o modelo.

Esses dois problemas (*overfitting* e *underfitting*) são ilustrados esquematicamente na Figura 3 (c). Em linhas gerais, ambos os problemas estão relacionados a complexidade do modelo (se elevada ou simplificada demais) e a estatística dos dados usados (espaço amostrado pelos exemplos fornecidos – caracterizado pela quantidade, dimensionalidade, qualidade e relação das amostras). Assim, a divisão treino/teste é uma metodologia padrão para evitar tais problemas no aprendizado de máquina supervisionado.

Somado a esses problemas, outra questão importante a ser discutida sobre os modelos de IA é seu tratamento

como caixa-preta. Neste sentido, o modelo não explica como chegou nas conclusões apresentadas, sejam elas descritivas ou preditivas. Pelo fato desses algoritmos de aprendizado serem agnósticos ao domínio do problema, mapeando um problema específico fornecido em um problema de otimização, o modelo não elabora passos de análise e síntese que físicos e outros profissionais estão habituados. Por isso, ao analisar os resultados obtidos, percebemos em geral, que o modelo funciona apenas como uma caixa que dada uma entrada, reproduz o padrão esperado da saída. Isso pode gerar uma baixa confiança no modelo [19]. Caso haja um modelo teórico já estabelecido e acurado, a utilidade de um modelo de aprendizado de máquina no formato caixa-preta faz sentido quando possui uma acurácia igual ou superior e um custo computacional (de tempo e/ou recursos) menor. Este tipo de modelo também é chamado de modelo substituto (*surrogate model*) e sua utilidade reside em reduzir custo de processamento com relação ao modelo base teórico. Por outro lado, novas técnicas tem sido pesquisadas para tornar as caixas-pretas mais transparentes, em um esforço de criar modelos de inteligência artificial explicáveis (conhecidos sob a sigla XAI, *Explainable AI* [20, 21]). Modelos de regressão simbólica que encontram as expressões matemáticas que relacionam a entrada com a saída são um exemplo de modelos explicáveis [22], que inclusive são facilmente interpretáveis por físicos.

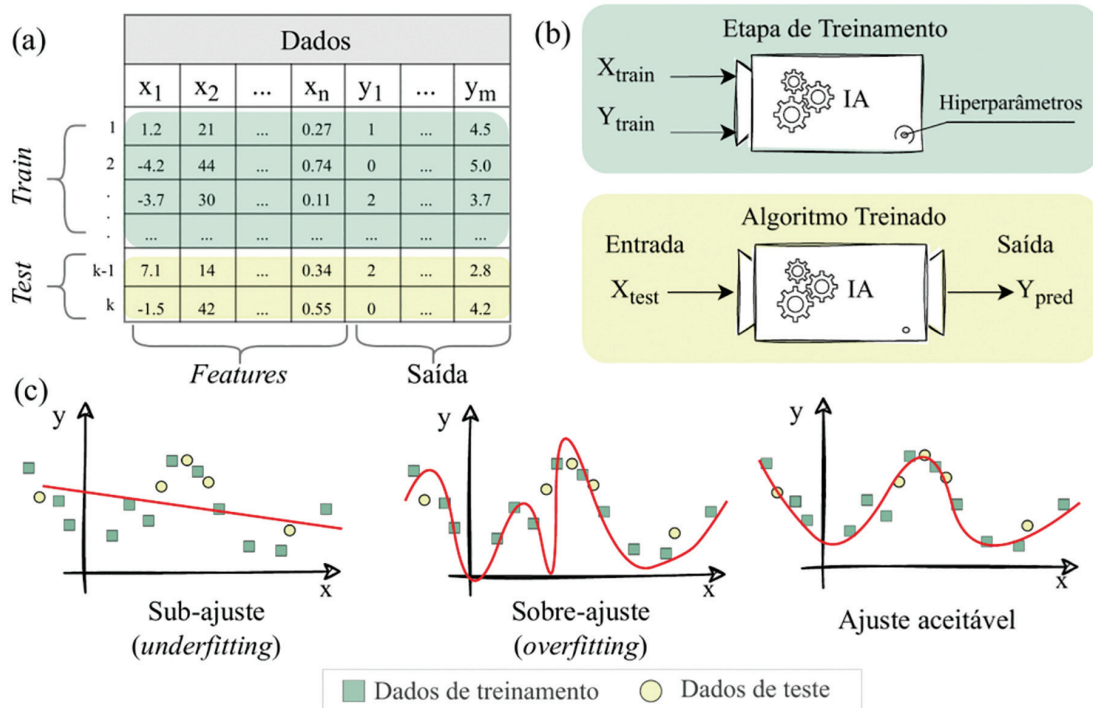


Figura 3: Aprendizado de Máquina Supervisionado: (a) Dados em representação tabular, com destaque para uma separação arbitrária entre conjunto de treinamento (em verde) e teste (em amarelo); (b) Uso de algoritmo supervisionado onde primeiro são usados dados para o treinamento (em verde), seguido do uso do algoritmo treinado (em amarelo); (c) Resultado esquemático de uma regressão em uma dimensão (apenas uma *feature* denotada por x), na qual é possível distinguir entre curvas treinadas com sub-ajuste, sobre-ajuste e ajuste aceitável.

Outras iniciativas visam diminuir a quantidade de dados (instâncias) necessária para se obter bons desempenhos com os algoritmos, na linha denominada *Zero-shot Learning*, [23], ou utilizar modelos treinados com dados de um domínio em outros domínios relacionados, linha denominada de *Transfer Learning* [24].

A área de aprendizado de máquina é muito ampla e ativa, com novos desenvolvimentos aparecendo constantemente na literatura. Não é objetivo deste trabalho descrever em detalhes todas as suas nuances. Para uma discussão mais aprofundada, recomendamos a leitura da literatura base já bem estabelecida [16–18, 25]. Para uma introdução ao *Aprendizado Profundo* no qual são usados algoritmos de redes neurais artificiais, sugerimos o trabalho de Arruda et al. [26].

2. Metodologia

2.1. Experimental

A obtenção dos dados de rolagem de diferentes objetos foi feita usando dois planos inclinados apresentados na Figura 4. O plano inclinado em (a) é adaptado de um kit didático com sensores eletrônicos e controle preciso de angulação, e as medidas foram feitas dentro de um laboratório didático da Universidade Federal do Oeste da Bahia. Já o plano inclinado mostrado em (b) foi construído com materiais de fácil acesso a qualquer estudante, sendo que os dados foram coletados com auxílio de um cronômetro de um telefone celular disparado manualmente e foram realizadas na residência do primeiro autor deste trabalho (HF). Em cada aparato foram feitas marcações de largada e chegada. Os objetos foram soltos do repouso a partir de uma linha de largada, momento em que se inicia a contagem de tempo, que é medido até o objeto cruzar a linha de chegada. Foram realizadas, para cada altura inicial, cerca de dez repetições para cada objeto. Os objetos utilizados foram: aro, cilindro (disco) e esfera de materiais diferentes como se observa na Figura 4.

Uma característica dos métodos de aprendizado de máquina é que eles são adaptativos: quanto mais dados

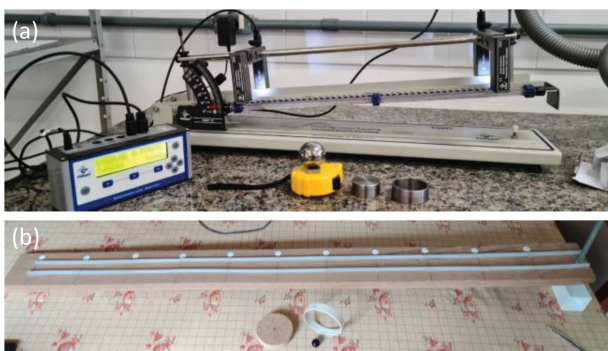


Figura 4: Aparatos experimentais utilizados: (a) para os ângulos de 5°, 10°, 15° e 20° com esfera, disco e aro de aço; (b) para os ângulos de 4° e 7.7° com aro de PVC e esfera de vidro.

disponíveis, mais precisos eles tendem a ser. Desta forma, neste experimento coletamos uma grande quantidade de dados, cerca de 800 medições, que são disponibilizados no material suplementar.

2.2. Computacional

Neste trabalho realizamos as tarefas de classificação e regressão para demonstrar o uso do Aprendizado de Máquina no problema do plano inclinado. Inicialmente foram selecionadas as variáveis descritivas (*features*) que representam cada exemplo dos dados para cada tarefa. Para a classificação, foi considerada a tupla (altura, ângulo, tempo) como entrada X e o objeto (aro, cilindro ou esfera) como saída y . Isso significa que desejamos descobrir o objeto nos baseando somente na medida da altura, do ângulo e do tempo que o objeto demora para percorrer o percurso. Para a regressão, foi considerada a tupla (altura, ângulo, classe 1, classe 2, classe 3) como entrada X e a velocidade média como saída y . A utilização de três novas colunas para representar cada classe é denominada *one-hot encoding*, uma vez que entradas de algoritmos de aprendizado de máquina devem ser números (palavras aro, cilindro e esfera não podem ser inseridas diretamente). O *one-hot encoding* transforma cada palavra da coluna de objeto inicial em uma nova coluna, colocando o número 1 quando a respectiva linha for do objeto denotado naquela coluna e 0 quando não for.

Todas as *features* foram escalonadas através da técnica *MinMaxScaler* que consiste em transformar cada coluna para que os valores fiquem no intervalo de 0 a 1. Isso é necessário para que todas as *features* sejam consideradas de maneira igual no funcionamento interno de alguns algoritmos de aprendizado de máquina. O escalonamento do k -ésimo elemento da i -ésima coluna é escrito como:

$$x'_{ki} = \frac{x_{ki} - \min(x_i)}{\max(x_i) - \min(x_i)}. \quad (14)$$

Foram utilizadas a biblioteca de Aprendizado de Máquina de código aberto Scikit-Learn [27] na versão 0.24.2 e a linguagem de programação Python [28] na versão 3.9.5. O código desenvolvido está disponível como um Jupyter Notebook no material suplementar deste artigo (<https://github.com/simcomat/IArpi>). Isso permite que o leitor reproduza os resultados e faça as análises que considerar conveniente.

Para a classificação considerou-se dez algoritmos diferentes, sendo oito deles de aprendizado de máquina, um denotado de modelo base (BM) e outro construído através do modelo físico da Eq. 5. O modelo BM é um comparativo com a probabilidade de aleatoriamente rotular cada entrada (altura, ângulo, tempo) com uma classe (aro, cilindro ou esfera) equiprovável. Neste caso, o BM é usado como valor mínimo de comparação, uma vez que qualquer método de classificação diferente do puro acaso deve possuir índices de performance superiores a ele em dados balanceados (aproximadamente a mesma quantidade de exemplos para cada classe). Os classificadores

LDA (Linear Discriminant Analysis), QDA (Quadratic Discriminant Analysis) e GNB (Guassian Naive Bayes) são algoritmos de aprendizado de máquina baseados em modelos probabilísticos. Os classificadores RF (Random Forest) e GD (Gradient Boosting) são algoritmos de ensemble de Árvores de Decisão. O algoritmo MLP (Multilayer Perceptron) é baseado em redes neurais. Já o KNN (K-Vizinhos Mais Próximos) e o SVM (Máquina de Vetor Suporte) são métodos baseados na proximidade espacial dos dados em N dimensões. Uma discussão detalhada sobre o funcionamento interno de cada um desses algoritmos pode ser encontrada na literatura especializada [16–18, 25, 27]. Como exemplo, apresentamos as principais ideias do KNN no Apêndice C.

Como métrica de desempenho da classificação foram usadas a acurácia e o coeficiente κ de Cohen. A acurácia mede o total de exemplos corretamente classificados pelo total de exemplos testados. Ela pode ser calculada através da matriz de confusão, uma representação gráfica onde as linhas i representam as classes verdadeiras dos dados e as colunas j as classes previstas (cada valor e_{ij} da matriz é o total de exemplos da classe i previsto para a classe j , sendo o total de acertos quando $i = j$ e de erros quando $i \neq j$). Já o coeficiente κ de Cohen mede o quanto o classificador difere de um classificador que apenas sorteia a classe prevista, onde $-1 \leq \kappa \leq 1$, sendo valores abaixo de 0.2 considerados próximos ao classificador aleatório. Para valores próximos de 1 o classificador está predizendo exatamente os valores esperados (i.e., $y_{pred} = y_{true}$) [29].

Para a regressão foram escolhidos sete algoritmos diferentes, sendo seis deles de aprendizado de máquina e um o modelo físico (MF). Para o aprendizado de máquina foram escolhidas as contrapartes de regressão da Máquina de Vetor Suporte (SVMR), do K-Vizinhos Mais Próximos (KNNR), da Random Forest (RFR), do Gradient Boosting (GBR) e do Multilayer Perceptron (MLPR). Além destes, foi feita a regressão linear tradicional (LR) que realiza a minimização dos quadrados dos resíduos para obter os melhores coeficientes de uma combinação linear das *features* de entrada.

Como métrica de desempenho da regressão foram usados o coeficiente de determinação R^2 e o erro absoluto médio (MAE), conforme implementados no Scikit-Learn [27]:

$$R^2 = 1 - \frac{\sum_{j=1}^{k_{test}} (y_{true_j} - y_{pred_j})^2}{\sum_{j=1}^{k_{test}} (y_{pred_j} - \bar{y}_{true})^2}, \quad (15)$$

$$MAE = \frac{1}{k_{test}} \sum_{j=1}^{k_{test}} |y_{true_j} - y_{pred_j}|, \quad (16)$$

onde k_{test} é a quantidade de exemplos no conjunto de teste e \bar{y}_{true} a média aritmética de y_{true} .

Não foi feita a otimização dos hiperparâmetros de nenhum dos algoritmos utilizados, ou seja, todos os hiperparâmetros, tanto para a regressão quanto para a

classificação, foram mantidos como padrão (*default*) da implementação dos algoritmos no Scikit-Learn [27].

Para cada uma das tarefas, foram usados 80% dos dados como treinamento do modelo e 20% como teste. Cada exemplo foi escolhido aleatoriamente para pertencer a cada um desses dois conjuntos, sendo usado o *random state* de 42 (semente da função pseudoaleatória usada para inicializar a sequência).

3. Resultados

Na Figura 5 (a), (b) e (c) são apresentados os $k = 803$ dados coletados para a esfera (321), o cilindro (160) e o aro (322), respectivamente. Observa-se que os dados coletados pelo aparato experimental da Figura 4 (b) (ângulos 4.0 e 7.7) possuem maior dispersão (menor precisão) do que aqueles medidos com o aparato (a), devido a imprecisões de construção e ao tempo de reação humano na medição. Apesar disso, ambos os resultados concordam com as tendências de quanto maior o ângulo de inclinação do plano, menor o tempo transcorrido; e que, para os mesmos ângulos e alturas, o tempo de descida do aro é maior do que do cilindro, que por sua vez, é maior do que da esfera.

A velocidade média apresentada na Fig. 5 (d), (e) e (f) foi calculada considerando-se a distância percorrida sobre o plano inclinado para o respectivo intervalo de tempo. Dessa forma, observa-se a propensão dos dados para uma linha de tendência geral para cada objeto, cujo tempo transcorrido é proporcional a raiz quadrada da altura inicial. Apesar da dispersão dos dados, é possível identificar a tendência de diminuição da velocidade média indo da esfera para o aro.

Essa descrição analítica dos resultados pode ser usada para gerar modelos fenomenológicos da relação entre as variáveis através de uma abordagem física tradicional. A abordagem que focamos aqui é a construção de um modelo fenomenológico através dos algoritmos de aprendizado de máquina supervisionados, como descritos na Seção 2.2.

Uma vez estabelecidas as tarefas e as *features* necessárias, cada algoritmo escolhido foi treinado e testado. Os resultados são apresentados nas Figuras 6 e 7. O modelo que apresentou melhor desempenho da tarefa de classificação foi o KNN, como pode ser visto nas Figuras 6 (a) e (b), conseguindo acertar qual objeto rolava sobre o plano em 83.2% dos casos, com maior coeficiente de Cohen, $\kappa = 0.739$. Na Figura 6 (c) é apresentada a matriz de confusão para esse algoritmo, onde podemos ver os acertos e erros para cada objeto. Observa-se que o aro foi o menos confundido com os outros objetos (7 erros em 65, 10.8%), seguido do cilindro (4 erros em 32, 12.5%) e da esfera (16 erros em 65, 24.6%) – esses valores são obtidos pelas somas nas linhas da matriz. Uma possível explicação para o aro ser o menos confundido é o fato do seu coeficiente β ser o mais distante dos outros objetos (a diferença é de 0.5

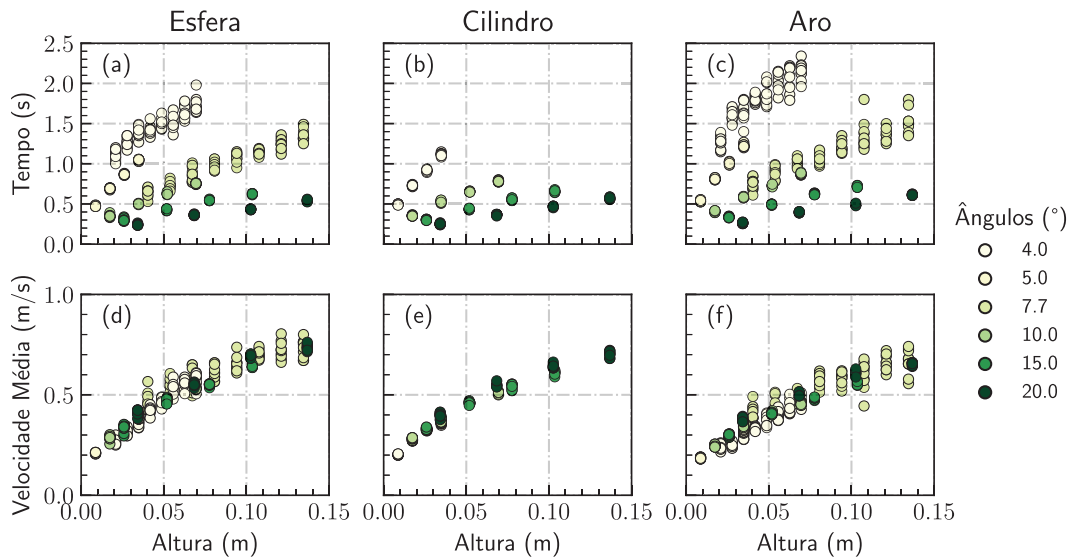


Figura 5: Gráficos de dispersão dos dados medidos: (a), (b) e (c) são a dispersão do tempo em função da altura inicial para esfera, cilindro e aro, respectivamente; (d), (e) e (f) são a dispersão da velocidade média em função da altura inicial para esfera, cilindro e aro, respectivamente.

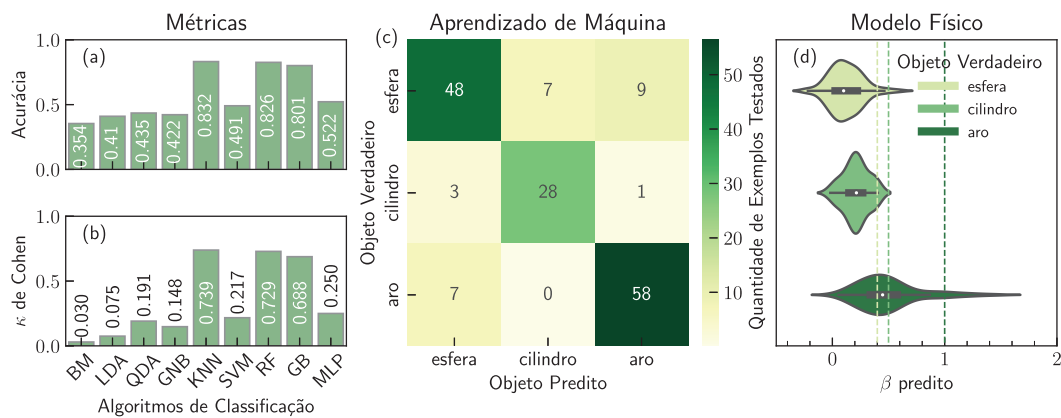


Figura 6: Resultados da Avaliação de Desempenho da Classificação para o conjunto de teste x_{teste} : (a) Acurácia e (b) Coeficiente κ de Cohen para os algoritmos de classificação usados; (c) Matriz de Confusão para o algoritmo KNN; (d) Distribuição dos coeficientes β preditos pelo modelo físico para os dados de teste (em tracejado estão os valores esperados pela teoria).

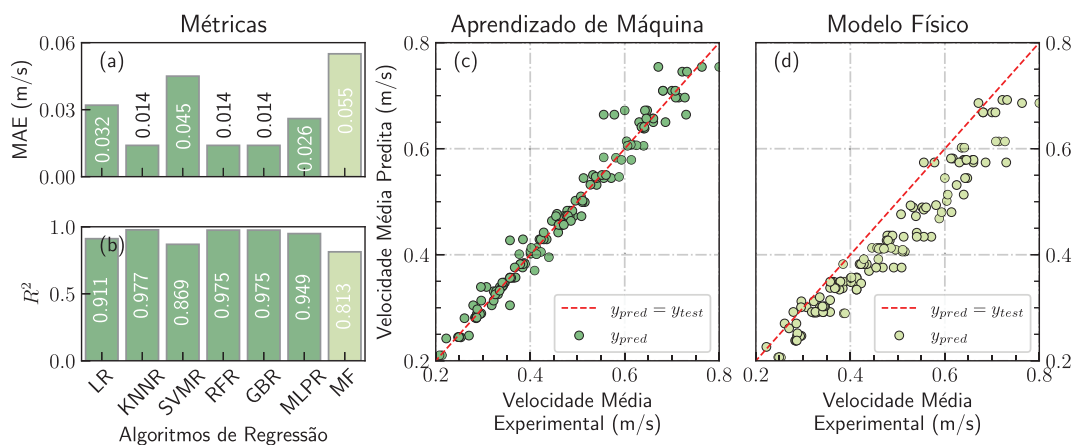


Figura 7: Resultados da Avaliação de Desempenho da Regressão para o conjunto de teste x_{teste} : (a) Erro Absoluto Médio, MAE; (b) Coeficiente de Determinação R^2 para os algoritmos de regressão usados; Dispersão da velocidade média predita y_{pred} pelo em função da velocidade média experimental y_{true} para o KNNR (c) e para o modelo físico (d).

entre β_{aro} e $\beta_{cilindro}$ e de 0.6 entre β_{aro} e β_{esfera} , enquanto a diferença entre $\beta_{cilindro}$ e β_{esfera} é de 0.1).

Como o modelo físico não prevê discretização de classes, obtivemos através dele um coeficiente β contínuo associado ao momento de inércia de cada objeto. Assim, o gráfico da Figura 6 (d) representa a distribuição dos valores de β para cada objeto verdadeiro, obtidos através da Eq. 5. O gráfico em (d) funciona de maneira análoga à matriz de confusão em (c), sendo cada distribuição a classe real do objeto e cada pico de distribuição o valor mais provável de β predito (ou seja, a classe predita). Observa-se pela tendência central das distribuições que cada objeto possui um momento de inércia característico. Entretanto, não é possível afirmar exatamente qual é cada objeto uma vez que ocorre sobreposição entre as distribuições observadas. É evidente que o modelo físico proposto possui limitações inerentes às hipóteses de sua elaboração e as restrições impostas para a comparação com os métodos de aprendizado de máquina (ambas abordagens trabalham com o mesmo conjunto de dados e variáveis). Apesar disso, a finalidade do contraste entre ambas as técnicas é demonstrar que a máquina foi capaz de aprender a distinguir os objetos partindo da mesma informação física fornecida ao modelo físico (*features* de entrada). Uma limitação que surge deste contraste é que a máquina não estabelece que a propriedade que permite distinguir os objetos seja uma propriedade física dos objetos (o momento de inércia), se restringindo a reproduzir o padrão aprendido dos dados. Em outras palavras, embora a máquina tenha sido capaz de induzir a física que reproduz os dados ela não é capaz de deduzir as propriedades físicas que dão origem a esses padrões.

Os modelos de regressão da Figura 7 também demonstram que os algoritmos de aprendizado de máquina foram capazes de reproduzir o padrão dos dados. Pelo contraste do desempenho entre o modelo físico e o melhor modelo de aprendizado de máquina (KNNR), Figuras 7 (b) e (c), observa-se que as limitações do modelo físico proposto trazem maior incerteza à predição da velocidade média de descida do objeto (erro médio de 0.055 m s^{-1} para o modelo físico e 0.014 m s^{-1} para o KNNR). A velocidade média predita pelo modelo físico é sempre menor que a velocidade experimental medida, o que novamente indica as limitações do modelo devido a inadequação das hipóteses assumidas de que o objeto rola sem deslizar ou que a velocidade inicial é nula. O modelo de aprendizado de máquina não se baseia nessas hipóteses do modelo físico, atendo-se a aprender o padrão dos dados, e conseguindo um desempenho melhor como modelo fenomenológico de predição.

Um último detalhe que gostaríamos de destacar é que as *features* escolhidas para esse problema foram determinadas pelo conhecimento do domínio de física. Ou seja, apesar dos modelos de aprendizado de máquina não saberem física internamente, a física foi adicionada a eles pela escolha consciente dos dados de entrada (altura, ângulo e tempo). Por exemplo, se assumíssemos que a

cor dos objetos fosse um atributo relevante, o modelo de aprendizado de máquina permitiria que essa informação fosse inserida. Se ainda, por descuido, escolhêssemos um objeto de cada cor, então a cor seria atributo suficiente para o modelo de aprendizado de máquina classificar corretamente os dados. Todavia, sabemos que a cor do objeto de nada importa neste problema. Para invalidarmos a escolha dessa *feature* precisaríamos aumentar a amostragem dos nossos dados, de forma a ter objetos iguais de cores diferentes. Só assim o modelo aprenderia estatisticamente que a cor não é relevante para o problema proposto. É a esse tipo de amostragem do espaço de dados que nos referimos ao destacar que o aprendizado de máquina requer dados em quantidade e qualidade suficiente para funcionar adequadamente. No problema aqui tratado, poderíamos adicionar informações da massa m e do raio R de cada objeto ou do atrito μ entre o objeto e o plano. Com essa informação adicional nos dados, é provável que a performance dos algoritmos fosse ainda melhor, o que não significa que o modelo aprendeu mais física com os dados, apenas que um conjunto maior de atributos permite distinguir melhor entre três objetos, já que a chance de um atributo ser diferente é maior. Contudo, precisaríamos garantir que houvesse repetições para objetos diferentes com a mesma massa ou o mesmo raio, para evitar que o modelo utilizasse essa informação como única suficiente para separar os dados. Esse desafio de amostragem dos dados está relacionado à *maldição da dimensionalidade*: quanto maior o espaço de entrada, mais dados são necessários para que tenhamos uma amostragem significativa o suficiente para que o modelo não fique enviesado para o subespaço amostrado pelos dados – o que pode facilmente ser impossibilitado pela explosão combinatória. Dessa forma, aqui tratamos apenas de um subespaço do problema geral do rolamento no plano inclinado. Se aumentarmos a mistura dos nossos experimentos para ter mais condições diferentes de atrito, o desempenho do algoritmo irá progressivamente diminuir, à medida que não possui *features* para “aprender” o atrito. Essa é uma sutileza que gostaríamos de enfatizar, observando que realizamos experimentos em duas condições de atrito diferentes (Figura 4), sem comprometer o desempenho, uma vez que os algoritmos aprendem a separação dos objetos pelos ângulos, facilitando a predição. Apesar do algoritmo se valer disso de maneira indireta, lembramos que a dispersão dos dados de β tem uma origem física, já que a medida à qual temos acesso experimental é de γ (Eq. 13), na situação na qual a hipótese de vínculo entre a velocidade do centro de massa e a velocidade tangencial não é respeitada ($f \neq 1$).

4. Conclusão

Neste trabalho mostramos que os modelos de aprendizado de máquina aqui vistos conseguem “aprender física” no sentido restrito de reproduzir com muita

acurácia os padrões nos dados fornecidos, baseando-se na mesma informação física dos modelos teóricos (mesmo conjunto de variáveis). Porém, tais modelos não são capazes de generalizar além dos padrões aprendidos nem intuir sobre as limitações do modelo. Pelo fato dos algoritmos serem agnósticos aos conceitos físicos, é muito desafiador extrair aprendizados mais profundos de suas predições (o que configura uma área própria de pesquisa em inteligência artificial [20, 21]). Alguns modelos possuem formulações matemáticas características que podem ser úteis para uma compreensão mais ampla, normalmente nos denominados regressores paramétricos (como a regressão linear, que assume uma função linear como padrão nos dados). Uma vez que se descobrem os coeficientes da função previamente assumida que melhor descrevem os dados, pode-se estabelecer relação deles com a realidade física. Um exemplo clássico disso é a primeira Lei de Ohm, na qual a relação fenomenológica entre a queda de tensão V e a corrente elétrica I que circula por um fio metálico é uma função linear do tipo $V = RI$, sendo que a constante R encontrada pela regressão linear de V e I tem significado físico de resistência elétrica. Por outro lado, modelos não-paramétricos podem alcançar desempenhos extremamente satisfatórios nas tarefas realizadas sem explicitamente entregarem nenhuma informação adicional (como redes neurais profundas). Independentemente dessas características, é importante notar que todos os algoritmos de aprendizado de máquina aprendem a reproduzir o padrão dos dados através dos próprios dados. Isto significa que em nenhum dos casos aqui discutidos foi feito um esforço deliberado para introduzir física aos modelos, como ocorreria em um paradigma de ciência computacional tradicional – toda a física está contida na seleção das *features* de entrada dos algoritmos.

Uma vez que nos limitamos aqui a aplicação dos algoritmos como caixa-pretas, devemos destacar as utilidades e limitações de suas aplicações. Pelo contraste entre os modelos teóricos e de aprendizado de máquina, um pesquisador poderia concluir que as mesmas variáveis físicas selecionadas podem ser usadas para gerar um modelo mais acurado, o que é um teste indicativo sobre a validade das hipóteses assumidas no modelo físico – mas que exige cuidados com relação as instâncias de dados usadas para treinamento e teste (cuja separação pode gerar certo vazamento de informação entre as etapas, emulando mais informação do que o modelo deveria ter acesso). De uma perspectiva diferente, pesquisas que envolvam a análise de muitos dados (que podem ser rotulados por humanos em quantidade muito menor do que por um algoritmo automatizado) podem se valer dessas técnicas, por exemplo: a detecção de exoplanetas, [30]; a predição de ordem magnética em materiais bidimensionais [31]; e a análise de experimentos de altas energias [32, 33]. Em qualquer aplicação, como ferramentas de aprendizado automatizado, os algoritmos também podem falhar em suas previsões, seja devido ao

sub-ajuste ou ao sobre-ajuste, ou pela própria natureza estatística de seu funcionamento.

Os procedimentos metodológicos e as ferramentas aqui apresentadas são de fácil acesso e compreensão. Acreditamos ser muito importante para o estudante de física familiarizar-se com tais técnicas, acumulando assim os saberes dos diferentes paradigmas de se fazer ciência.

Ensinar física às máquinas é ainda uma jornada em seu início. Diversos novos desdobramentos estão sendo pesquisados e construídos dessa interação interdisciplinar entre computação e física. Por enquanto podemos treinar o computador para ter um desempenho semelhante ou melhor do que Galileo teve ao analisar os dados do rolamento no plano inclinado. Contudo, as tarefas em que podemos fazer isso são apenas fragmentos de tudo aquilo que Galileo fez neste experimento. Dessa forma, apesar de máquinas poderem aprender física, elas ainda não são capazes de substituir físicos. Em vez disso, o aprendizado de máquina vem se mostrando uma ferramenta auxiliar de grande importância para o trabalho deles.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001. W.F. Espinosa-García agradece à Universidad de San Buenaventura-Medellín pelo suporte (projeto 449011-22.01-01P). G. M. Dalpian e J. N. B. Rodrigues agradecem ao CNPq pelo financiamento. Este trabalho também foi realizado no âmbito do Projeto Temático FAPESP 17/02317-2. Os autores agradecem aos estudantes da turma de Física II da UFOB Bárbara Beatriz Pedroza Santos, Camila de Jesus Vargas, Geovanna Sousa Bonifácio Magalhães, Lorena Vieira Soares, Rafael Sales Souza Farias e William Philip Leite de Castro pelo suporte na realização dos experimentos.

Material suplementar

O seguinte material suplementar está disponível online: Apêndice – A. Da presença de conteúdo de computação em cursos de física; B. Do rolamento no plano inclinado por formalismo Lagrangiano; C. Do algoritmo KNN.

Referências

- [1] F. Ricci, L. Rokach e B. Shapira (eds), *Recommender systems handbook* (Springer, New York, 2015), 2 ed.
- [2] G.R. Schleder, A.C.M. Padilha, C.M. Acosta, M. Costa e A. Fazzio, *Journal of Physics: Materials* **2**, 032001 (2019).
- [3] O.N. Oliveira Jr e M.C.F. Oliveira, *Front. Chem.* **10**, 930369 (2022).

- [4] S. Schmidt. Escola Superior Ilum busca cientista do futuro, Revista FAPESP, junho de 2022.
- [5] G. Galilei, *Dialogues Concerning Two New Sciences* (Macmillan, New York, 1914).
- [6] C.M. Ramos e J.R. Bonjorno, *Física: História e Cotidiano* (FTD, São Paulo, 2005).
- [7] H.M. Nussenzveig, *Curso de Física Básica: Mecânica* (Blucher, Rio de Janeiro, 2013).
- [8] A. Chaves e J.F. Sampaio, *Física Básica: Mecânica* (LTC, Rio de Janeiro, 2011).
- [9] R.L. Chaplin e M.G. Miller, *American Journal of Physics* **52**, 1108 (1984).
- [10] R. Cross, *European Journal of Physics* **36**, 065047 (2015).
- [11] M. Newman, *Computational Physics* (Createspace Independent Publishing Platform, North Charleston, 2012).
- [12] D. Walker, *Computational Physics* (Mercury Learning & Information, Dulles, 2016).
- [13] P.O.J. Scherer, *Computational Physics* (Springer, Cham, 2017).
- [14] G.R. Schleder e A. Fazzio, *Revista Brasileira de Ensino de Física* **43**, e20200407 (2021).
- [15] A.L. Samuel, *IBM Journal of Research and Development* **3**, 210 (1959).
- [16] K. Faceli, A.C. Lorena, J. Gama, T.A. Almeida e A.C.P.L.F. Carvalho, *Inteligência Artificial – Uma Abordagem de Aprendizado de Máquina* (LTC, Rio de Janeiro, 2021).
- [17] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (O'Reilly Media, Sebastopol, 2019), 2 ed.
- [18] A. Burkov, *Machine learning engineering* (True Positive, Quebec, 2020).
- [19] M. Franklin e D. Lagnado, em: *HCI International 2022 Posters*, editado por C. Stephanidis, M. Antona e S. Ntoa (Springer, Cham, 2022).
- [20] M. Popolin Neto e F.V. Paulovich, *IEEE Trans. Vis. Comput. Graph.* **27**, 1427 (2021).
- [21] P.A. Angelov, E.A. Soares, R. Jiang, N.I. Arnold e P.M. Atkinson, *WIREs Data Mining and Knowledge Discovery* **11**, e1424 (2021).
- [22] S.M. Udrescu e M. Tegmark, *Science Advances* **6**, eaay2631 (2020).
- [23] I. Sucholutsky e M. Schonlau, *Proceedings of the AAAI Conference on Artificial Intelligence* **35**, 9739 (2021).
- [24] L.A. Torrey e J. Shavlik, em: *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques*, editado por E. Soria, J. Martin, R. Magdalena, M. Martinez e A. Serrano (IGI Global, Hershey, 2009), v. 1.
- [25] M. Kubat, *An introduction to machine learning* (Springer International Publishing, Basel, 2015), 1 ed.
- [26] H.F. Arruda, A. Benatti, C.H. Comin e L.F. Costa, *Revista Brasileira de Ensino de Física* **44**, e20220101 (2022).
- [27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., *Journal of Machine Learning Research* **12**, 2825 (2011).
- [28] G. van Rossum, *Python tutorial. Technical Report CS-R9526* (Centrum voor Wiskunde en Informatica, Amsterdam, 1995).
- [29] J. Cohen, *Educational and Psychological Measurement* **20**, 37 (1960).
- [30] H. Valizadegan, M.J.S. Martinho, L.S. Wilkens, J.M. Jenkins, J.C. Smith, D.A. Caldwell, J.D. Twicken, P.C.L. Gerum, N. Walia, K. Hausknecht et al., *The Astrophysical Journal* **926**, 120 (2022).
- [31] C.M. Acosta, E. Ogoshi, J.A. Souza e G.M. Dalpian, *ACS Applied Materials & Interfaces* **14**, 9418 (2022).
- [32] D. Guest, K. Cranmer e D. Whiteson, *Annual Review of Nuclear and Particle Science* **68**, 161 (2018).
- [33] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman e D. Shih, *Nature Reviews Physics* **4**, 399 (2022).