Física 1 – Ciências Moleculares

Caetano R. Miranda

crmiranda@usp.br

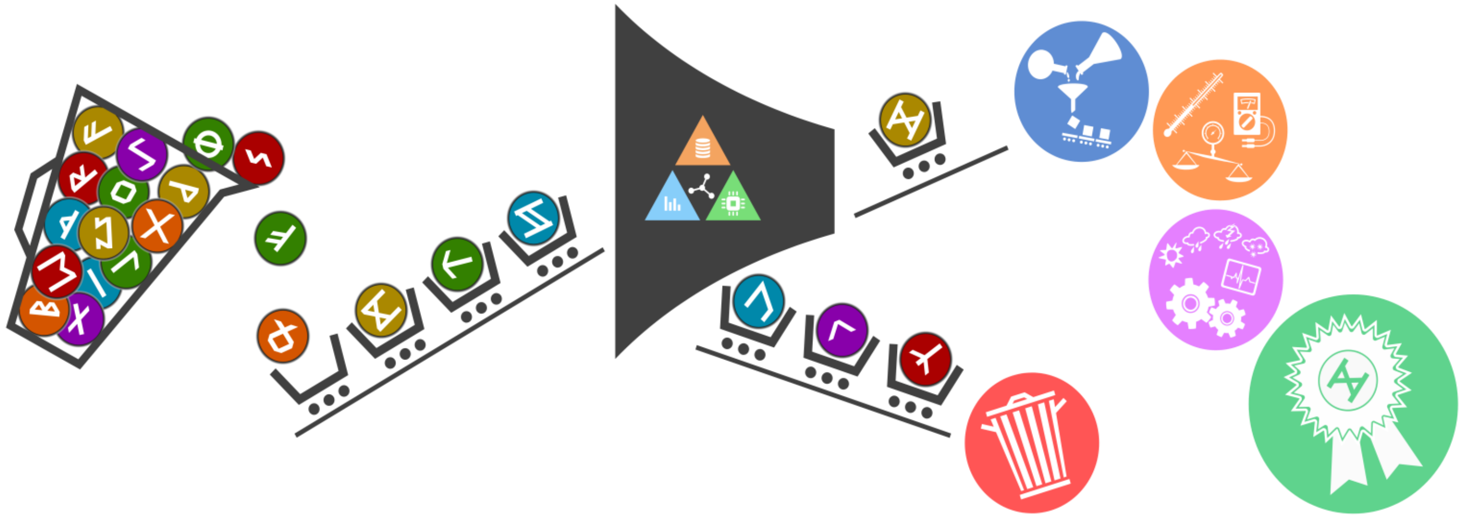
AULA 30 – 21/12/2023

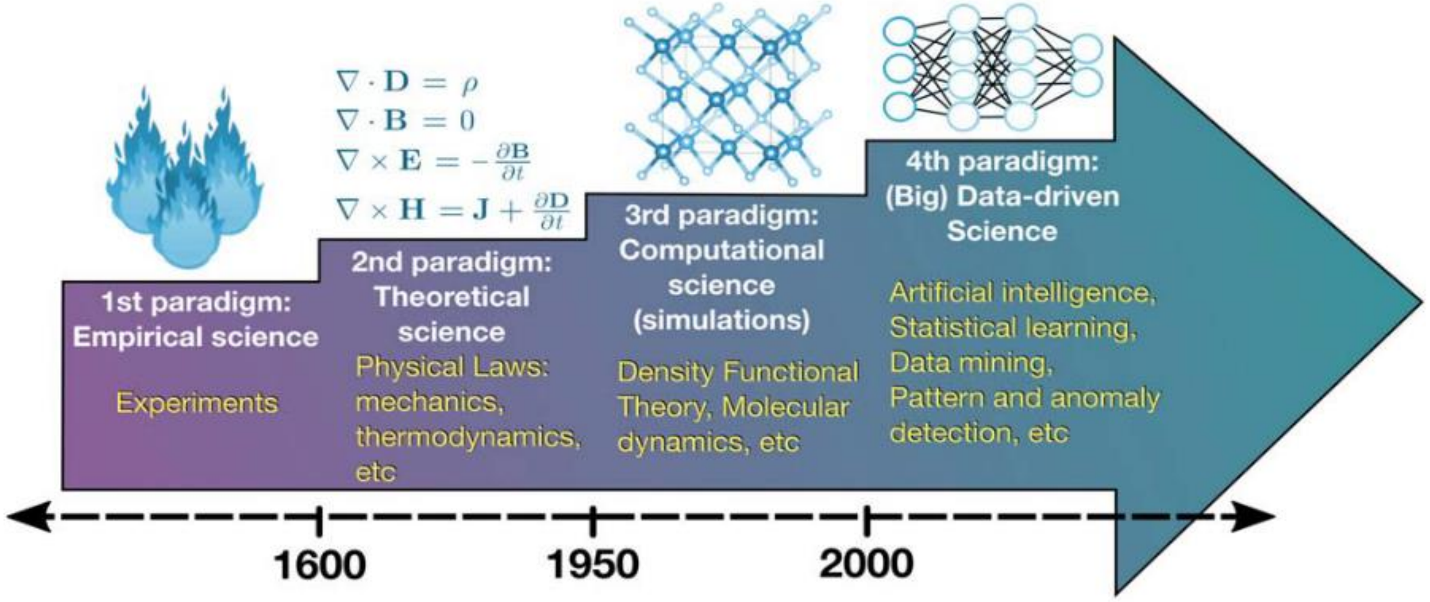
Movimento de Rolamento
Aprendizado de Máquina

Strategy

<u>Trial and Error – Multidimensional optimization– Parameters/Properties unknown</u>

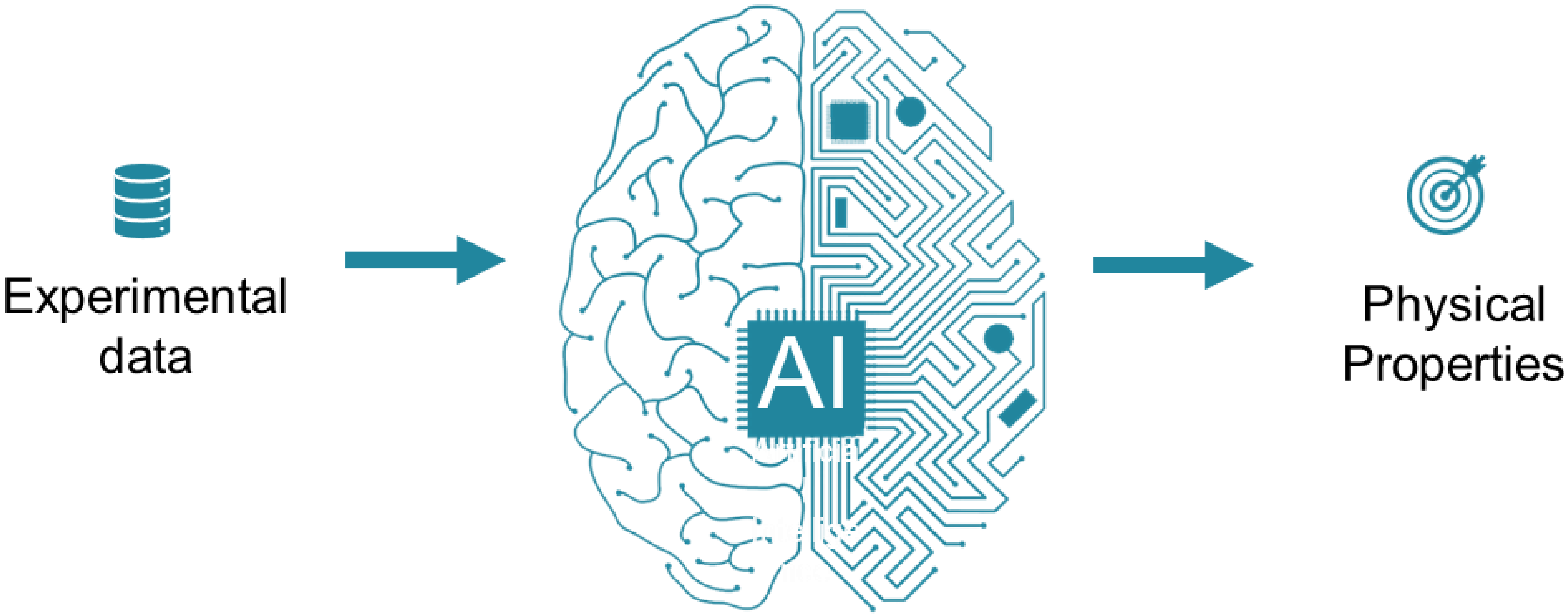
***In silico* Approach**





Schleder et al., Journal of Physics: Materials 2 (3)

Only the best candidates have to be synthesized, characterized and tested, resulting in <u>large cost and time savings</u>


Experimental data
AI
Physical Properties

Can machines discover emergence phenomena in Physics ?

nature physics
ARTICLES
https://doi.org/10.1038/s41567-020-1018-2
Check for updates

Machine learning the thermodynamic arrow of time

Alireza Seif[1,2], Mohammad Hafezi[1,2,3] and Christopher Jarzynski[1,4]

"...we have used various techniques to interpret what the network learns. In particular, by examining the optimized parameter values that emerge from the training, we have been able to identify which physical quantities the network uses to guess the direction of time's arrow. In this sense, ***our study represents a step toward AI driven discovery of physical concepts."***

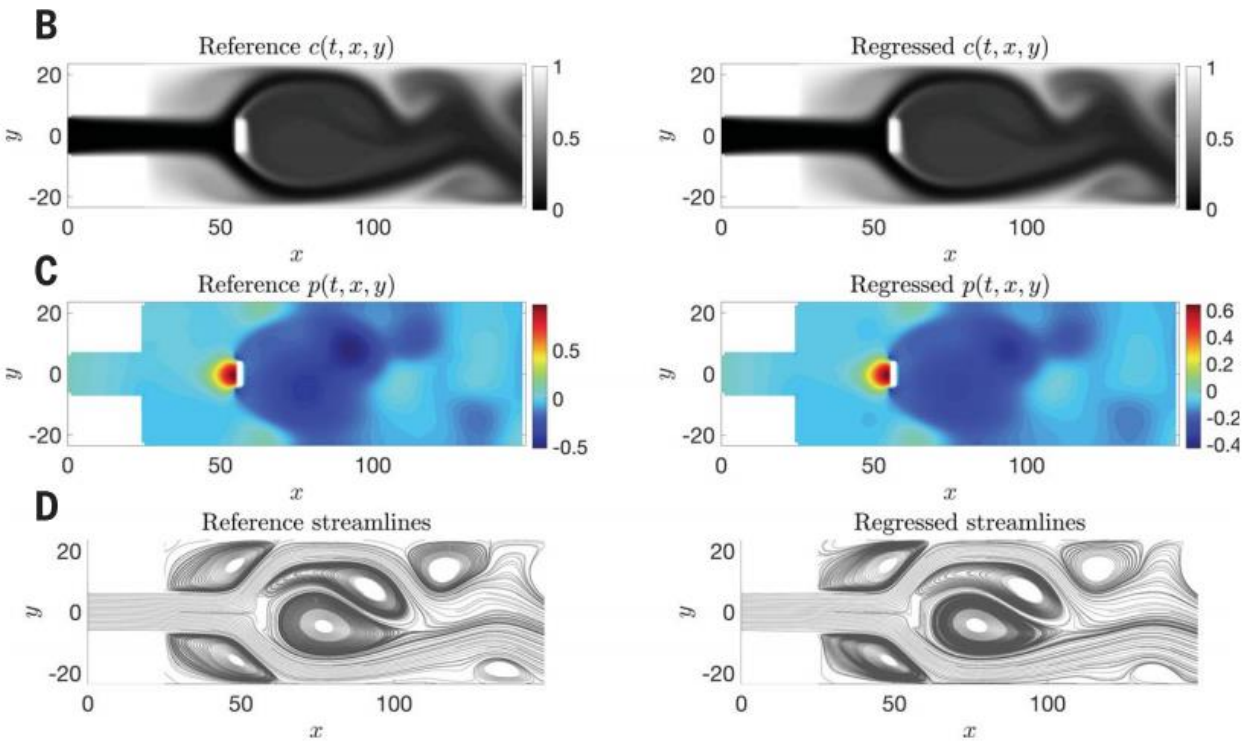
Seif et al., Nature Physics (September 2020)
https://doi.org/10.1038/s41567-020-1018-2

FLUID DYNAMICS

Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations

Maziar Raissi[1,2]*†, Alireza Yazdani[1], George Em Karniadakis[1]†

Raissi et al., Science 367, 1026–1030 (2020)



Can machines learn Physics without knowing the laws of Physics?

nature physics

ARTICLES

https://doi.org/10.1038/s41567-020-1018-2

Check for updates

Machine learning the thermodynamic arrow of time

Alireza Seif[1,2], Mohammad Hafezi[1,2,3] and Christopher Jarzynski[1,4]

"...we have used various techniques to interpret what the network learns. In particular, by examining the optimized parameter values that emerge from the training, we have been able to identify which physical quantities the network uses to guess the direction of time's arrow. In this sense, ***our study represents a step toward AI driven discovery of physical concepts."***

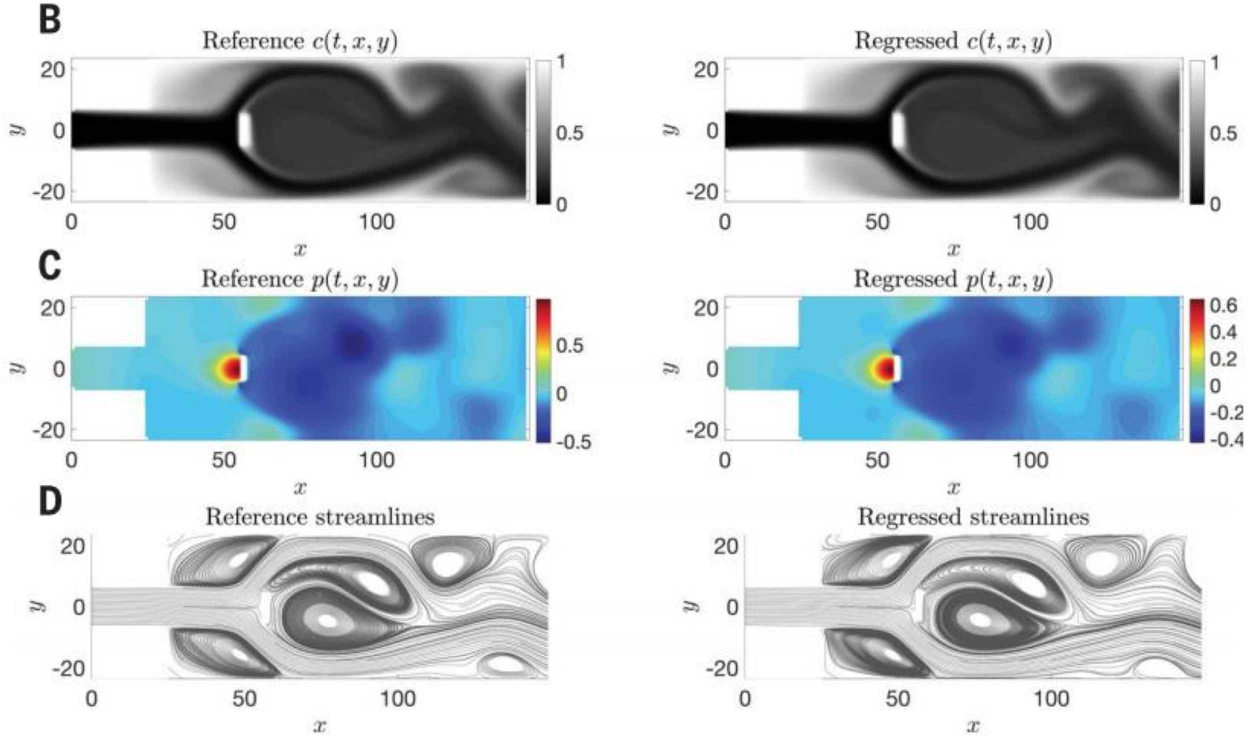
Seif et al., Nature Physics (September 2020)
https://doi.org/10.1038/s41567-020-1018-2

FLUID DYNAMICS

Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations

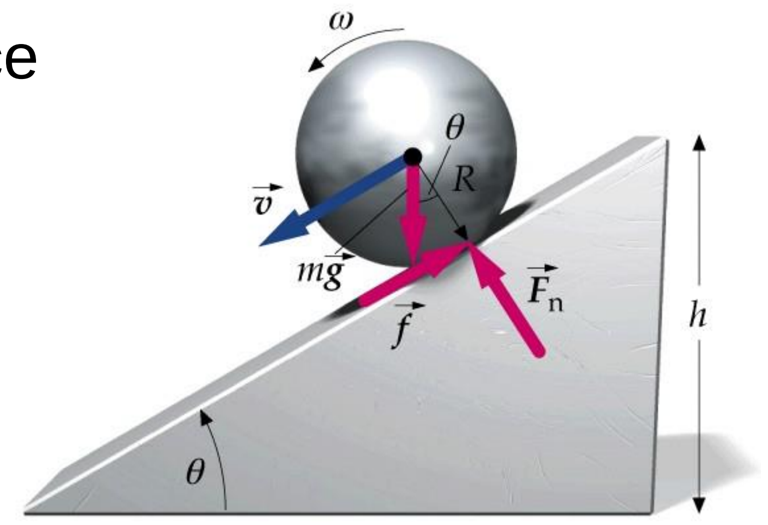
Maziar Raissi[1,2]*†, Alireza Yazdani[1], George Em Karniadakis[1]†

Raissi et al., Science 367, 1026–1030 (2020)



Rolamento sem deslizamento num plano inclinado

Objeto com seção transversal circular de raio R e massa m, desce rolando um plano inclinado com ângulo θ, sem deslizar



Aplicando a Segunda Lei de Newton:

rotação:

$$\tau_{ext} = I_{cm}\alpha = F_{at}R$$

$$F_{at} = \frac{I_{cm}}{R^2}a_{cm}$$

$$\boxed{\alpha = \frac{a_{cm}}{R}}$$

translação:

$$F_{res} = mgsen\theta - F_{at} = ma_{cm}$$

$$mgsen\theta - \frac{I_{cm}}{R^2}a_{cm} = ma_{cm}$$

$$\boxed{a_{cm} = \frac{gsen\theta}{1 + \frac{I_{cm}}{mR^2}}}$$

Rolando de um plano inclinado: qual chega primeiro?

- Esfera: $I_{cm} = \frac{2}{5} mR^2$

- Cilindro: $I_{cm} = \frac{1}{2} mR^2$

- Anel: $I_{cm} = mR^2$

$$a_{cm} = \frac{g sen\theta}{1 + \frac{I_{cm}}{mR^2}} = g sen\theta \cdot k$$

$$\boxed{\begin{array}{l} esfera \mapsto k = 0.71 \\ cilindro \mapsto k = 0.66 \\ anel \mapsto k = 0.5 \end{array}}$$

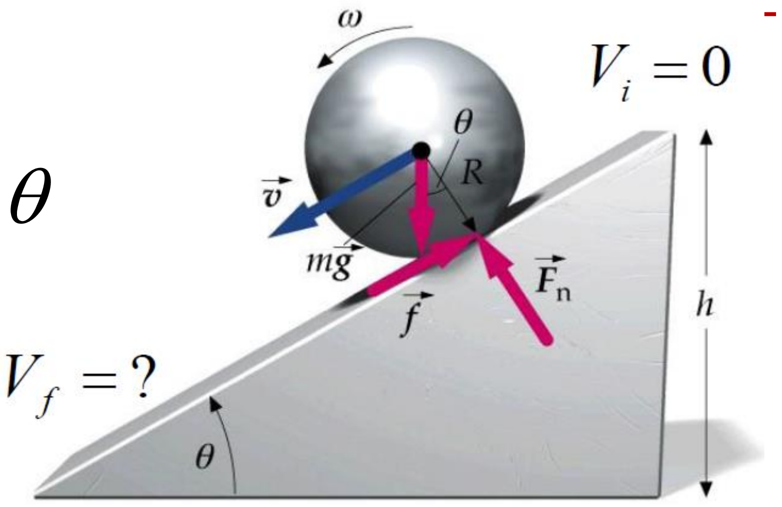
Esfera chega primeiro!

Qual a velocidade e energia cinética no final da rampa?

$$V_f^2 - V_i^2 = 2a\Delta S$$

$$V_f^2 = 2\frac{g\cancel{\sin\theta}}{1+\frac{I_{cm}}{mR^2}}\frac{h}{\cancel{\sin\theta}}$$

$$h = \Delta S.\sin\theta$$



$$\boxed{V_f^2 = 2gh.k}$$

$$\boxed{k = \frac{1}{1+\frac{I_{cm}}{mR^2}}}$$

$$\boxed{I_{cm} = (\frac{1}{k} - 1)mR^2}$$

$$\boxed{\omega = \frac{V_f}{R}}$$

$$T = \tfrac{1}{2}mV_f^2 + \tfrac{1}{2}I_{cm}\omega^2$$

$$T = \tfrac{1}{2}mV_f^2 + \tfrac{1}{2}(\frac{1}{k} - 1)m\cancel{R^2}(\frac{V_f^2}{\cancel{R^2}})$$

$$T = \cancel{\tfrac{1}{2}mV_f^2} + \tfrac{1}{2}\frac{1}{k}mV_f^2 - \cancel{\tfrac{1}{2}mV_f^2}$$

$$\boxed{T = mgh}$$

Força de atrito não realiza trabalho. No ponto de contato não há deslocamento.

Energia cinética final = Energia Potencial

Qual seria o maior ângulo para que não houvesse deslizamento?

$$\tau_{ext} = I_{cm}\alpha = F_{at}R$$

$$\boxed{a_{cm} = R\alpha}$$

$$\boxed{I_{cm} = Mk^2}$$

$$\boxed{a_{cm} = \frac{gsen\theta}{1 + \frac{k^2}{R^2}}}$$

$$F_{at} = \frac{I_{cm}}{R^2}a_{cm} = \frac{Mk^2}{R^2}\frac{gsen\theta}{\left(1 + \frac{k^2}{R^2}\right)} = Mgsen\theta\frac{k^2}{(k^2 + R^2)}$$

Por outro lado: $F_{at} = \mu_e N$

$$\cancel{M}gsen\theta\frac{k^2}{(k^2 + R^2)} = \mu_e\cancel{M}gcos\theta \Rightarrow \boxed{tg\theta = \frac{(k^2 + R^2)}{k^2}\mu_e}$$

Revista Brasileira de Ensino de Física, vol. 44, e20220214 (2022)
www.scielo.br/rbef
DOI: https://doi.org/10.1590/1806-9126-RBEF-2022-0214

Artigos Gerais

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 Junior[3], W. Espinosa-García[4], E. Novais[1], J.N.B. Rodrigues[1], G.M. Dalpian[1]

[1]Universidade Federal do ABC, Centro de Ciências Naturais e Humanas, Santo André, SP, Brasil.
[2]Faculdade de Informática e Administração Paulista, São Paulo, SP, Brasil.
[3]Universidade Federal do Oeste da Bahia, Centro das Ciências Exatas e das Tecnologias, Barreiras, BA, Brasil.
[4]Universidad de San Buenaventura-Medellín, Facultad de Ingenierías, Medellín, ANT, Colombia.

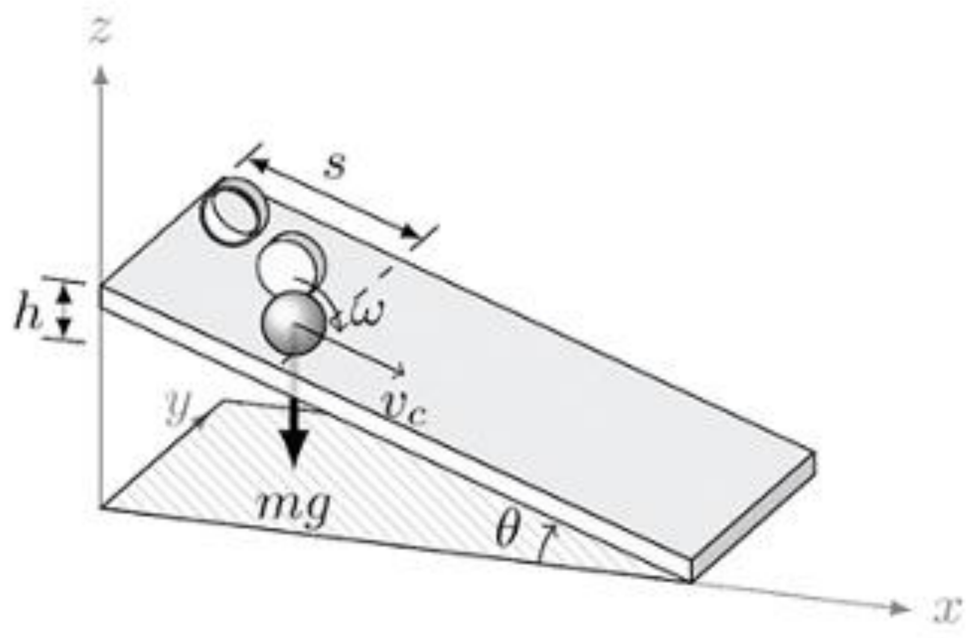


Figura 1: Ilustração do rolamento de um aro, um disco e uma esfera sobre um plano inclinado de um ângulo θ com relação ao plano horizontal. Na esfera são destacadas variáveis físicas de interesse (análogas para os outros objetos), como a força peso mg, a velocidade linear v_c do centro de massa, a velocidade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

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

$$I = \int_C r^2 dm = \beta m R^2, \tag{1}$$

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, $\beta = 1$.

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 predizer v_{med}, ou, dado (h, θ, v_{med}) predizer β. Usaremos exatamente essas duas possibilidades para explorar a aplicação de aprendizado de máquina supervisionado para predizer v_{med} (tarefa de regressão) e predizer o objeto, o que é análogo a predizer 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 $v_t = f v_c$, 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:

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

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:

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

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

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.

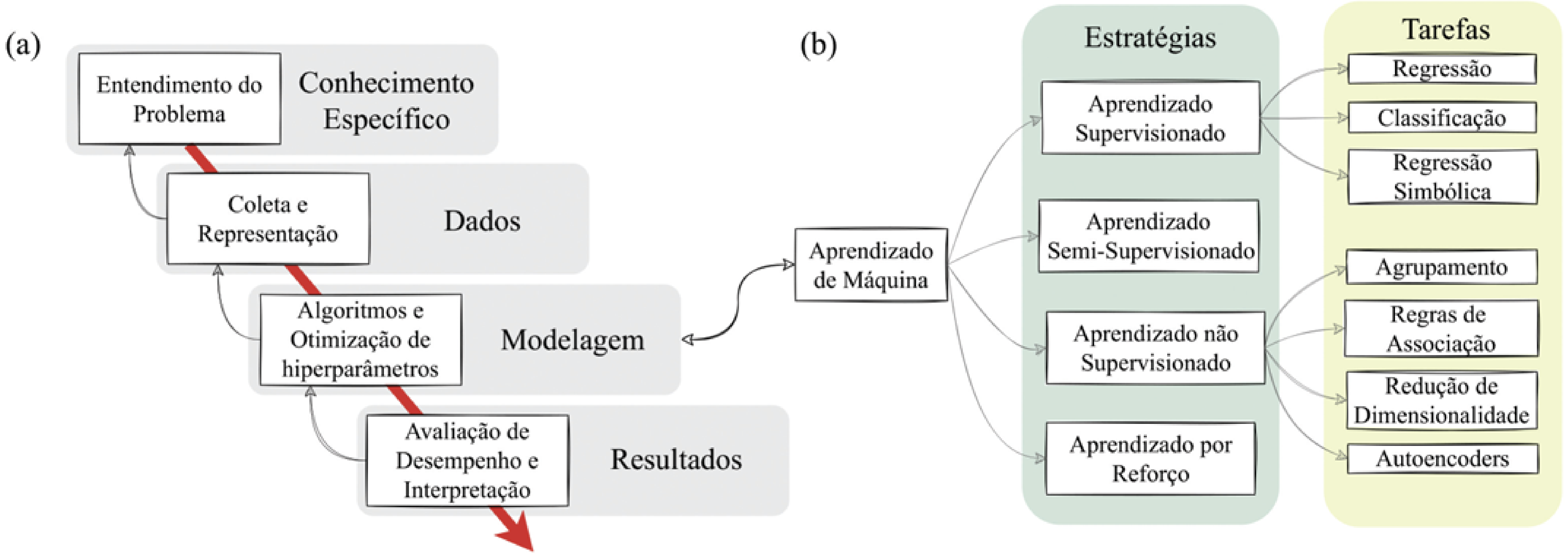


Figura 2: Aprendizado de Máquina: (a) Etapas para a utilização de técnicas de aprendizado de máquina (a seta vermelha simboliza o fluxo direto de trabalho e as setas em preto, o retorno às etapas anteriores visando o aperfeiçoamento); (b) Particionamento das técnicas de aprendizado de máquina que são 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).

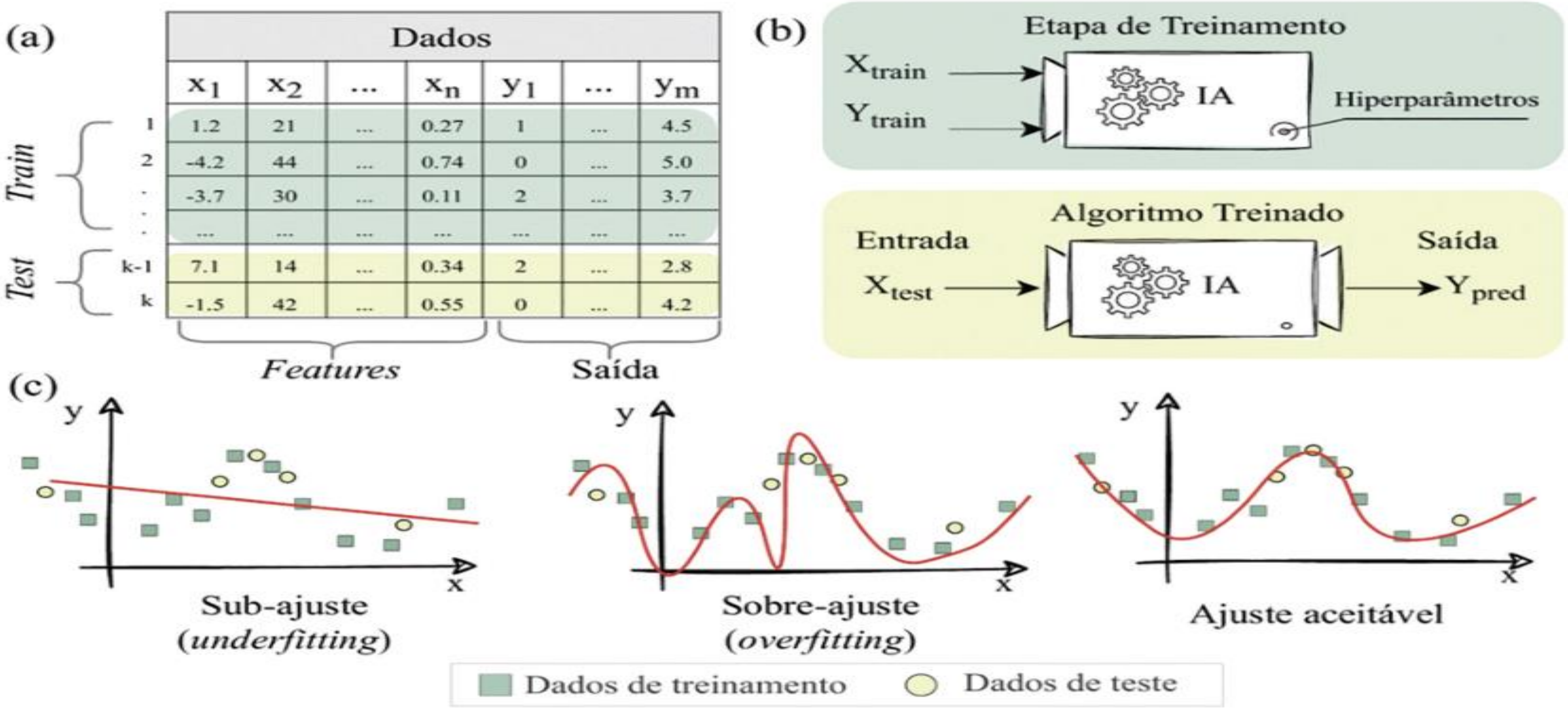


Figura 3: Aprendizado de Máquina Supervisionado: (a) Dados em representação tabular, com destaque para uma separação arbitrária entre conjunto de treinamento (em verde) e teste (em amarelo); (b) Uso de algoritmo supervisionado onde primeiro são usados dados para o treinamento (em verde), seguido do uso 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.

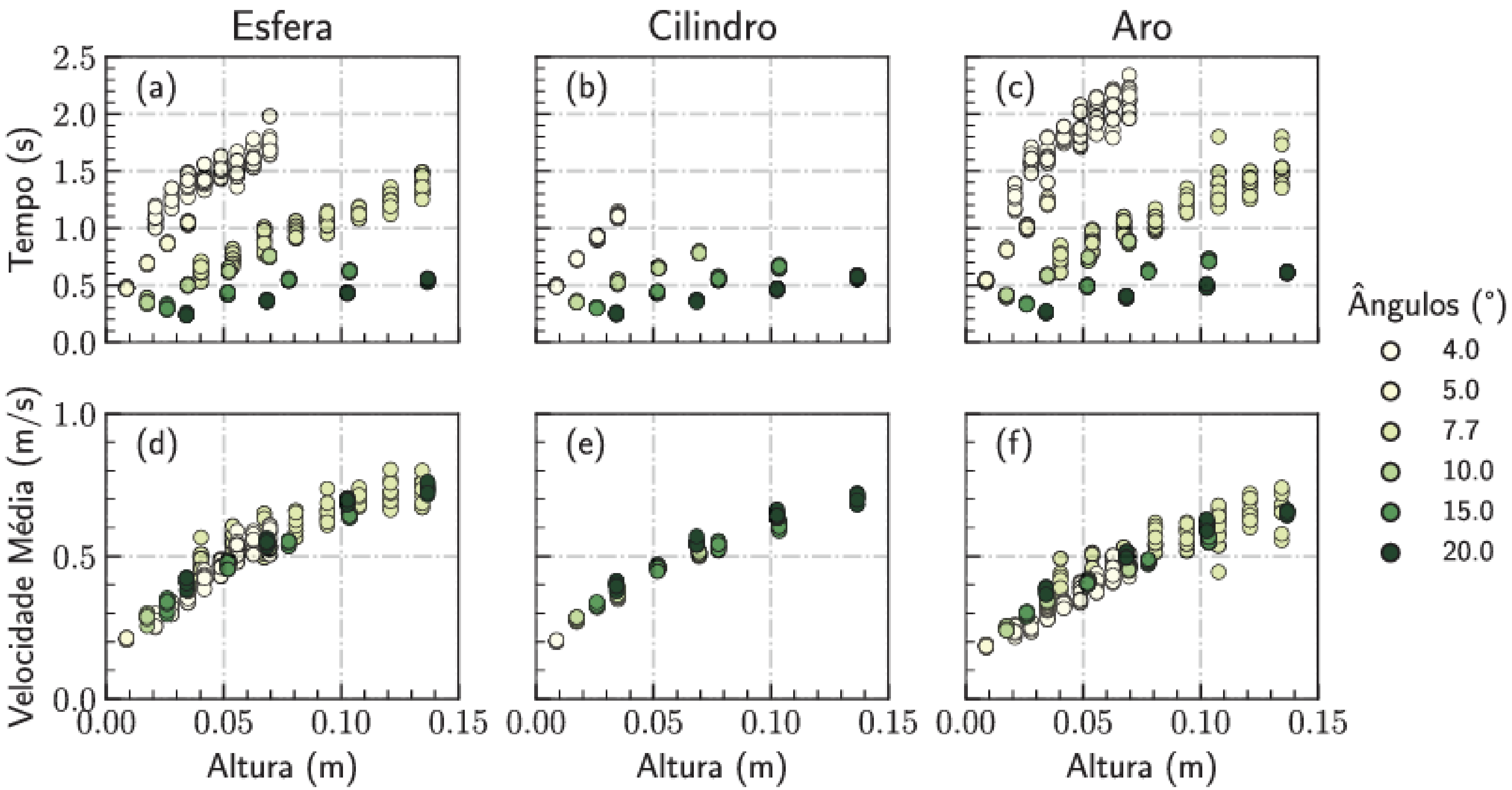
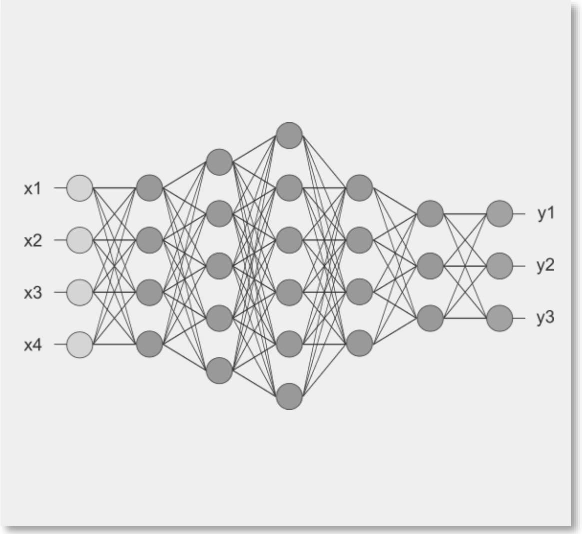


Figura 5: Gráficos de dispersão dos dados medidos: (a), (b) e (c) são a dispersão do tempo em função da altura inicial para esfera, cilindro e aro, respectivamente; (d), (e) e (f) são a dispersão da velocidade média em função da altura inicial para esfera, cilindro e aro, respectivamente.

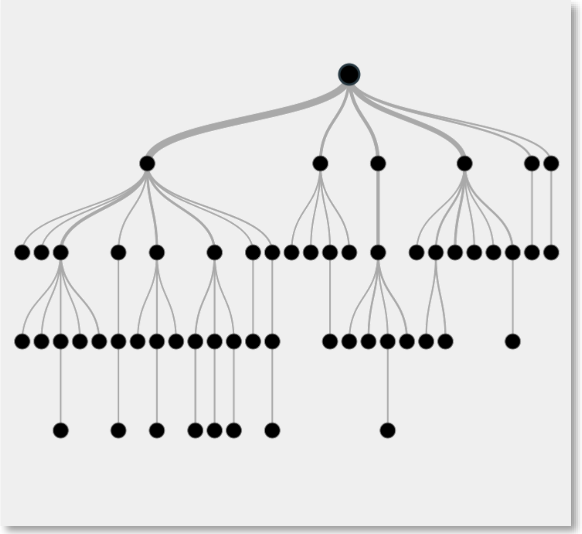
The regression models

Neural network
(MLPRegressor)



These are computational models inspired by a nervous system that are capable of machine learning as well as pattern recognition.

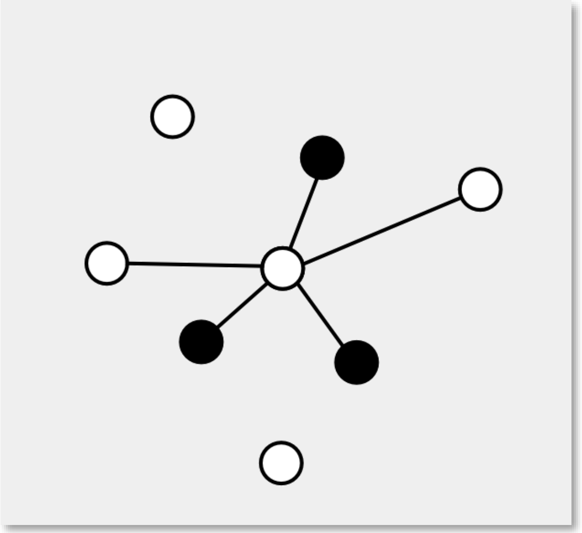
Decisions tree
(XGBoost)



It is a predictive a powerful modeling approach of achieving high accuracy.

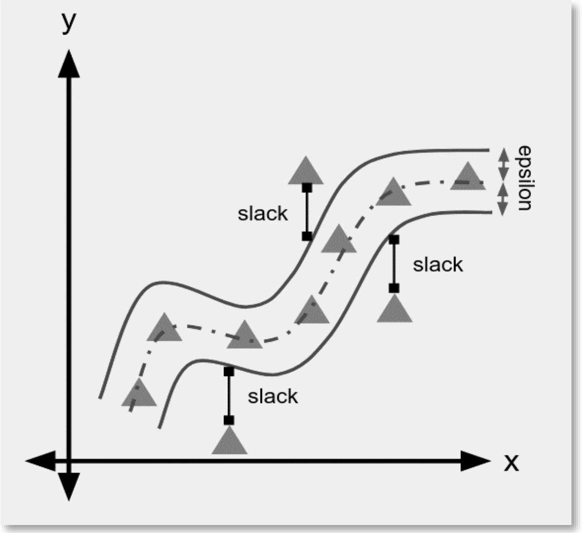
This model works with the idea of decision tree ensembles that use gradient boosting structures.

Euclidean distance
(KNN)



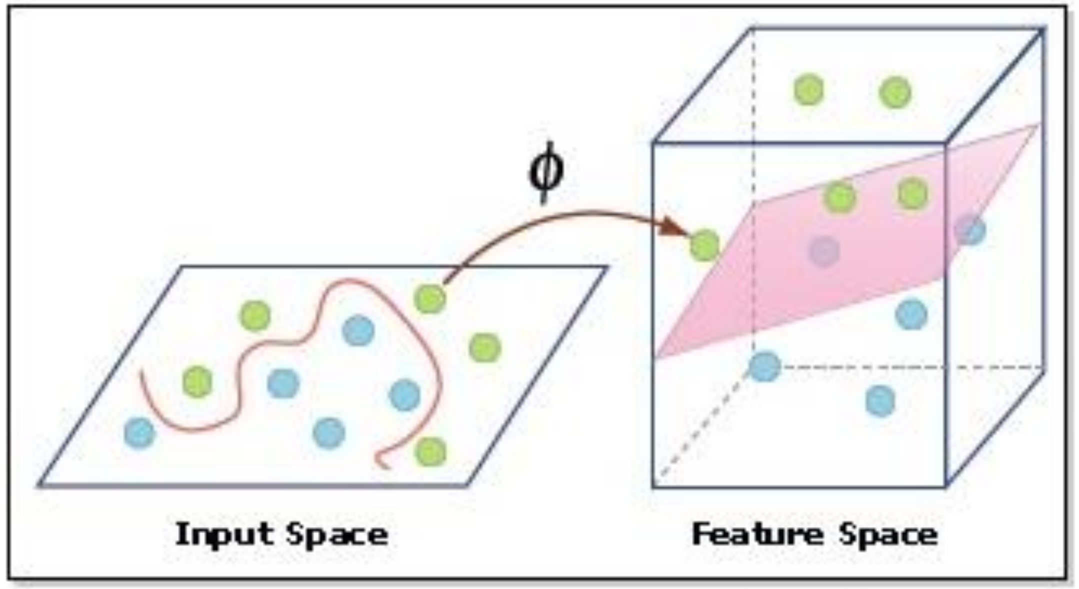
The KNN algorithm uses euclidian distance to predict the values of any new data points.

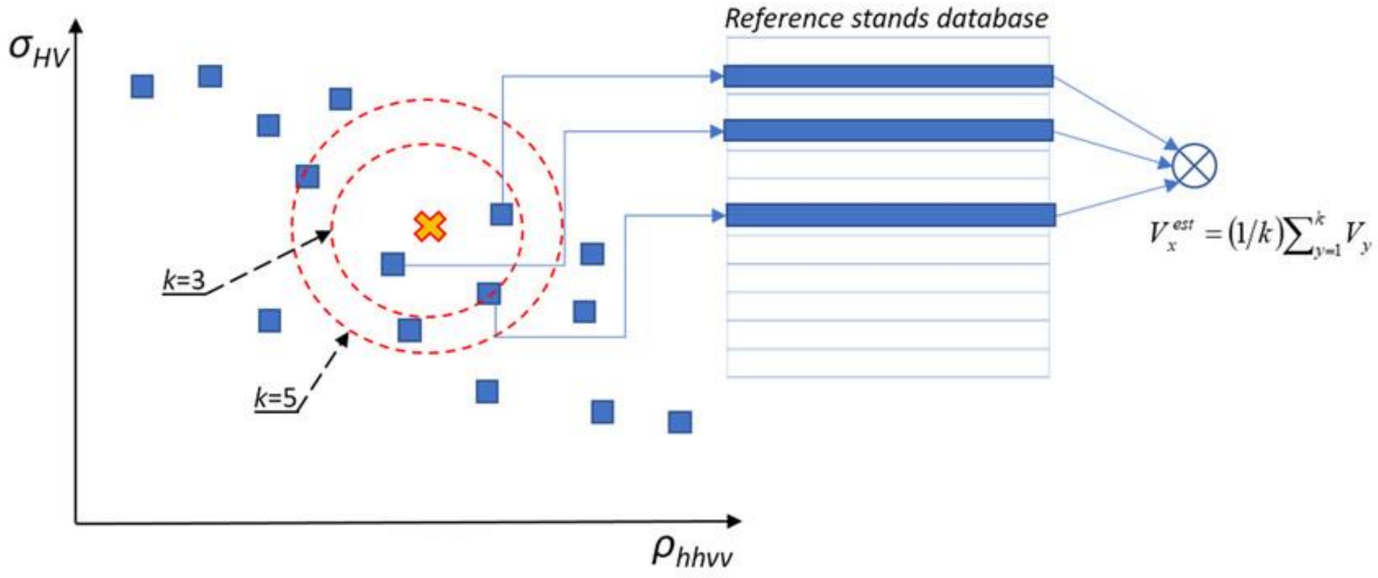
Support Vector Regressor
(SVR)

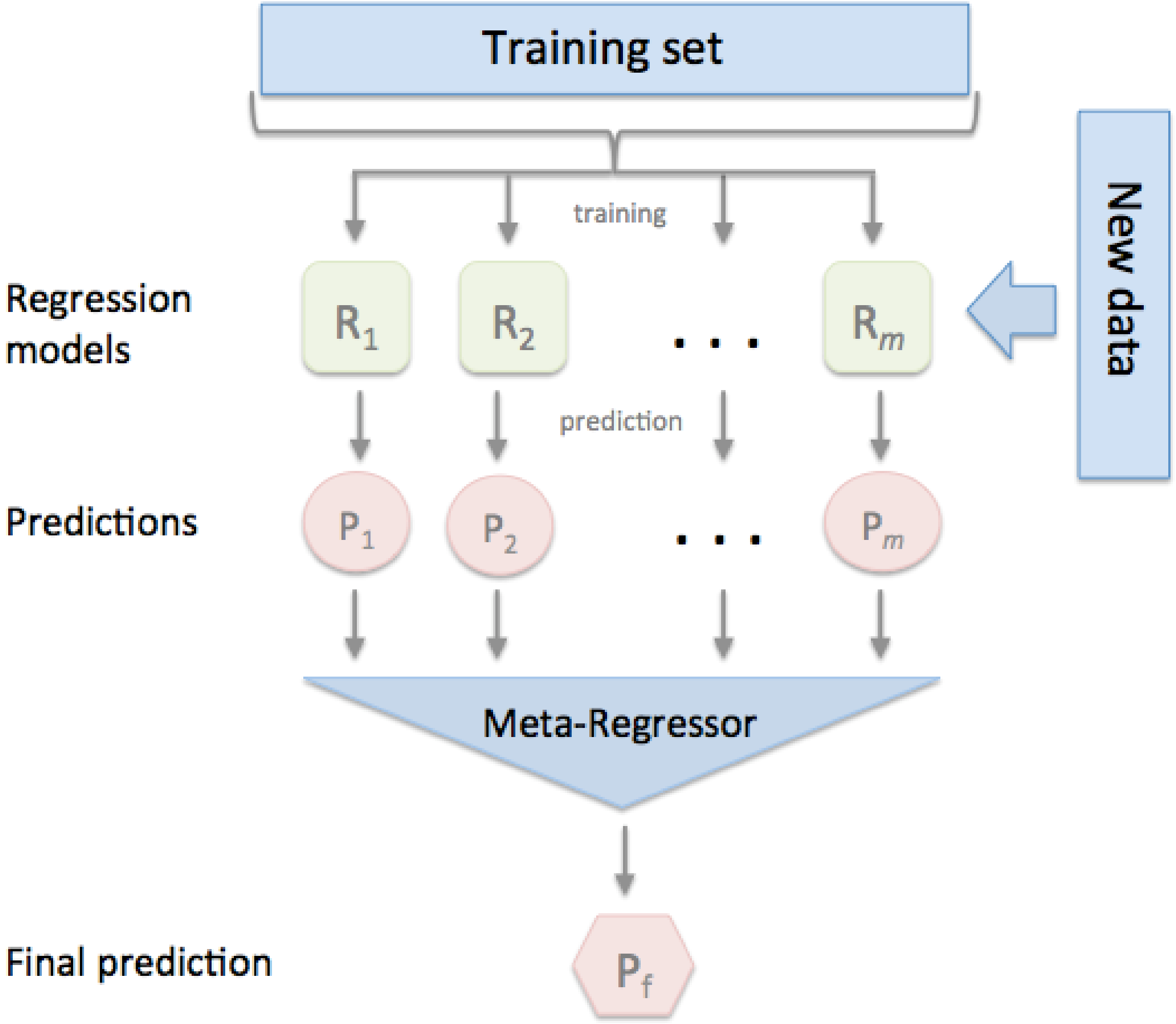


SVR tries to delimit data by analysing the boundary points through vectors that indicate which division best fit data.

division of the sample space


ϕ
Input Space
Feature Space


σ_{HV}
Reference stands database
k=3
k=5
$V_x^{est} = (1/k)\sum_{y=1}^{k} V_y$
ρ_{hhvv}


Training set
training
New data
Regression models
R_1
R_2
. . .
R_m
prediction
Predictions
P_1
P_2
. . .
P_m
Meta-Regressor
Final prediction
P_f

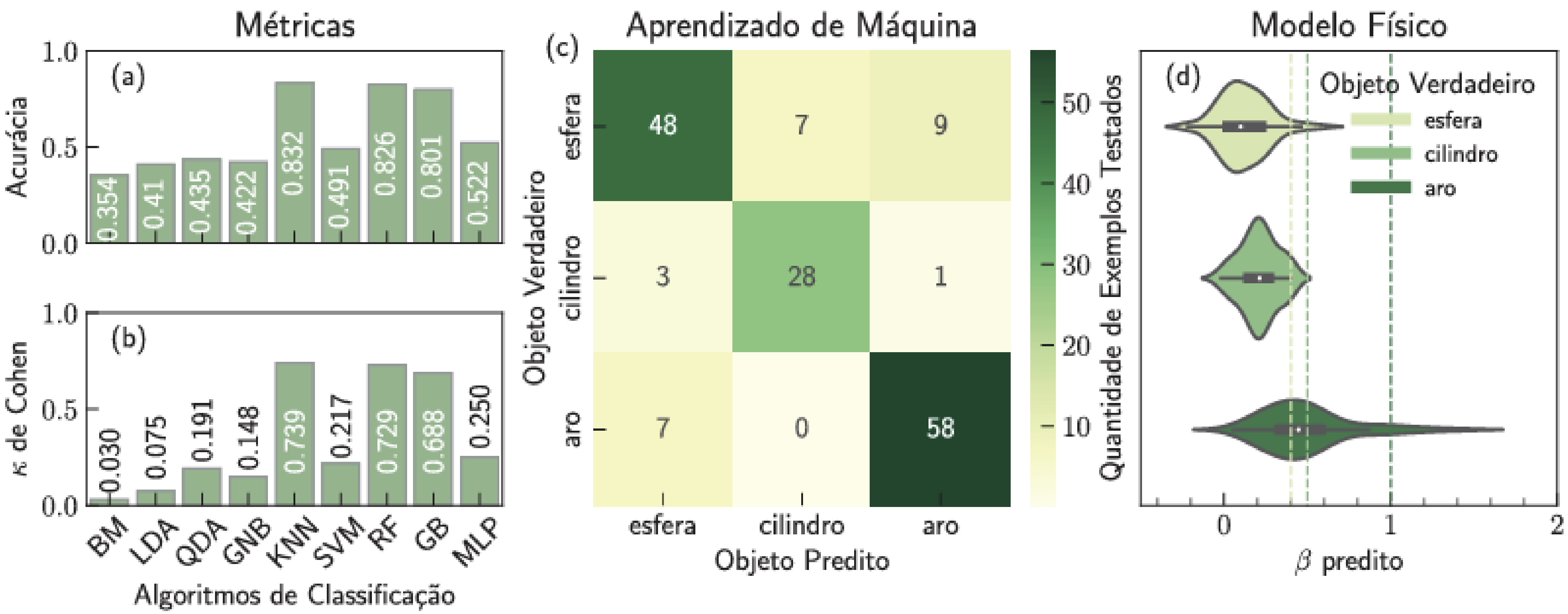


Figura 6: Resultados da Avaliação de Desempenho da Classificação para o conjunto de teste x_{test}: (a) Acurácia e (b) Coeficiente κ de Cohen para os algoritmos de classificação usados; (c) Matriz de Confusão para o algoritmo KNN; (d) Distribuição dos coeficientes β preditos pelo modelo físico para os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

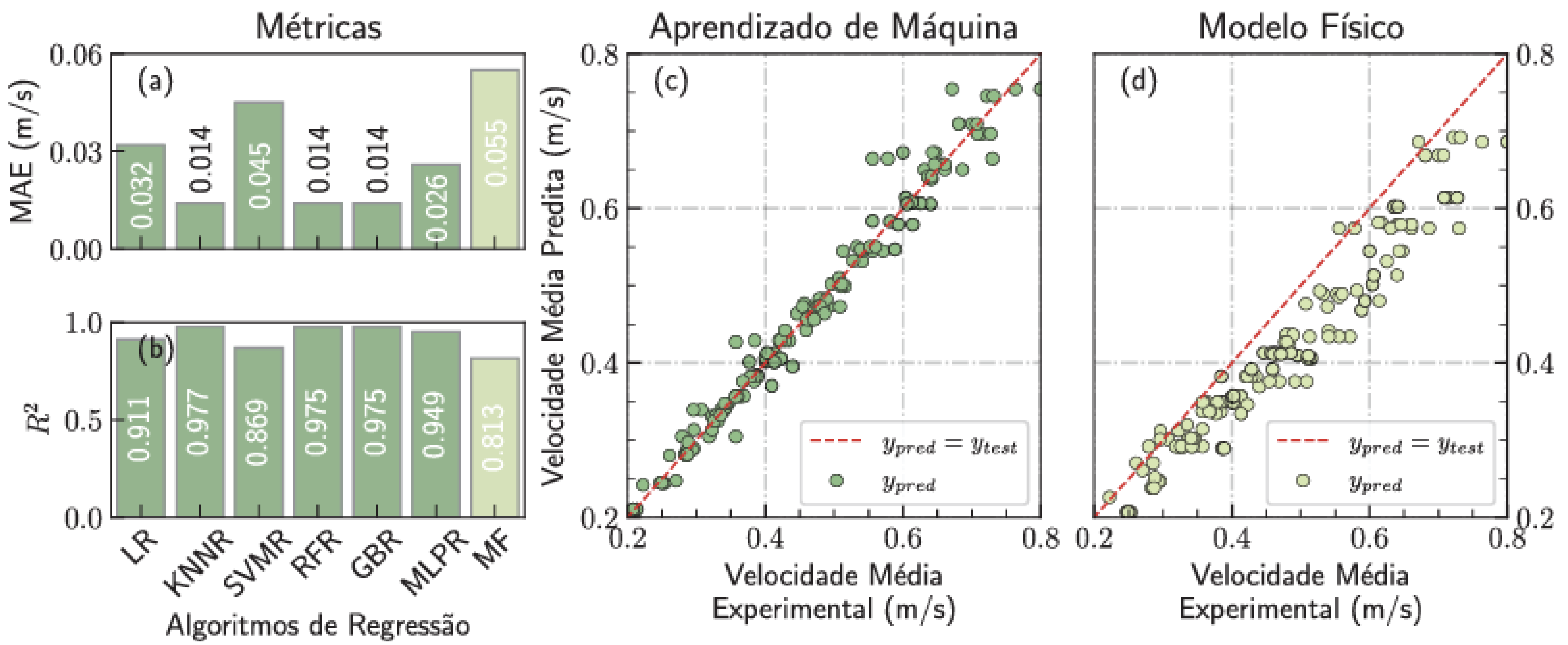


Figura 7: Resultados da Avaliação de Desempenho da Regressão para o conjunto de teste x_{test}: (a) Erro Absoluto Médio, MAE; (b) Coeficiente de Determinação R^2 para os algoritmos de regressão usados; Dispersão da velocidade média predita y_{pred} pelo em função da velocidade média experimental y_{true}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

Experimento

Compilar os dados:

```
# Colunas necessárias na tabela
colunas = ['Objeto', 'Ângulo (°)', 'Distancia (m)',
 'Altura (m)', 'Tempo (s)',
       'Velocidade Média (m/s)']
```

Acessar o notebook em Python

https://colab.research.google.com/drive/1lREMtTsMMiXomdEHq9NAjz5WN2lY9kd2#scrollTo=8HPhuGHDKpsj

ML_Physics.ipynb

IArpi

Introduzindo Aprendizado de Máquina no ensino de Física

Neste Jupyter Notebook são apresentados os passos para estudar um dataset de dados experimentais do rolamento em um plano inclinado. Concomitante ao uso de algoritmos de Aprendizado de Máquina (Machine Learning - ML), são realizados cálculos através de um modelo físico, discutido ao longo de cada tarefa.

- 1) Carregando os dados;
- 2) Realizando tarefa de Aprendizado Supervisionado de Classificação;
- 3) Realizando tarefa de Aprendizado Supervisionado de Regressão.

1) Carregando os dados

Para trabalhar com dados na forma de tabela, usamos a biblioteca **pandas**.

```
import pandas as pd  # Importando a biblioteca pandas com o nome pd
```

```
# tabela_dados = pd.read_csv('../datasets/rolling.csv', sep=';') # Carrega o arquivo .csv na variável tabela_dados.
tabela_dados = pd.read_csv('https://raw.githubusercontent.com/simcomat/IArpi/main/datasets/rolling.csv', sep=';') # Para abrir o .csv direto do github
```

Salve uma cópia no seu google drive

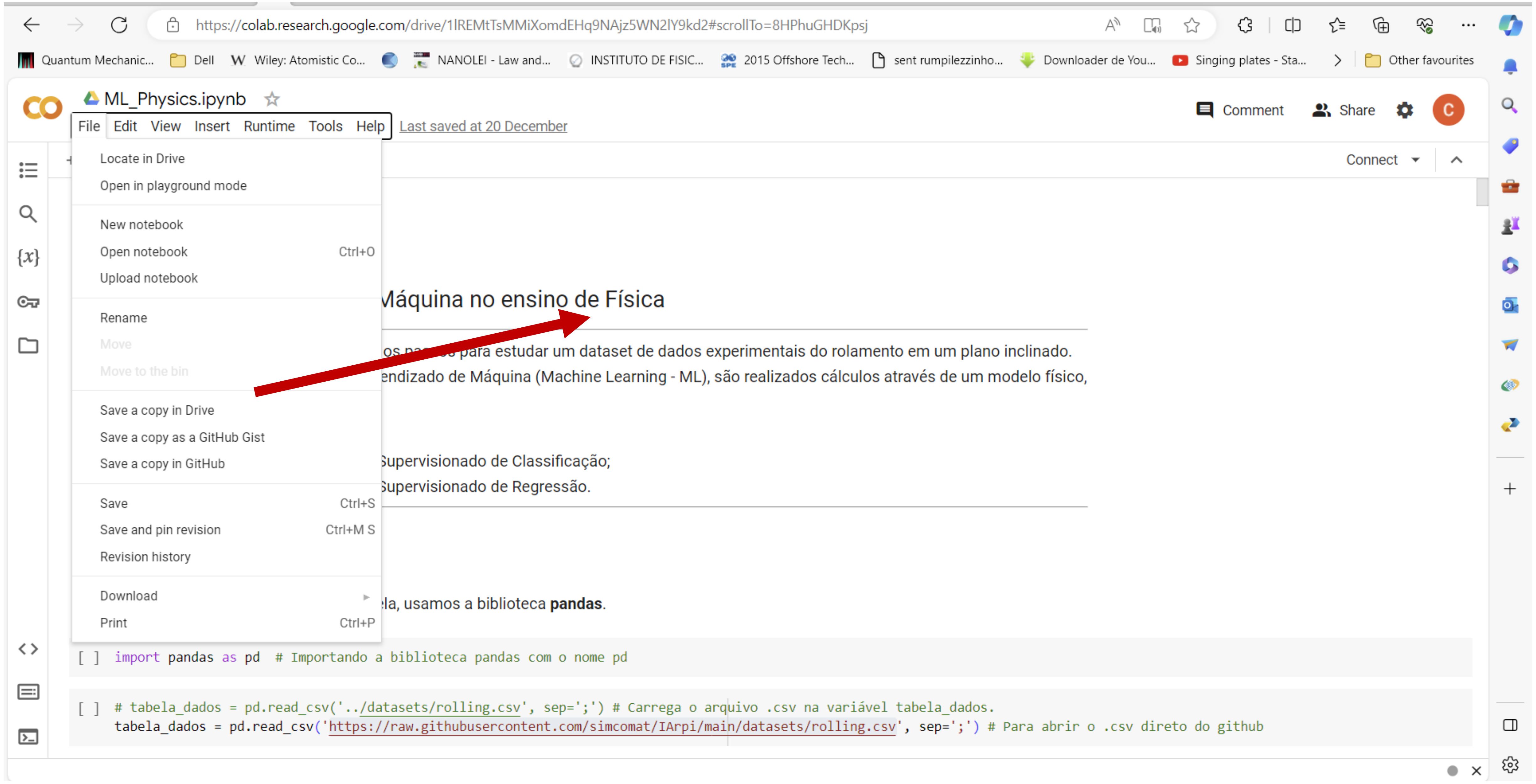


Tabela já existente

Objeto;Ângulo (°);Distancia (m);Altura (m);Tempo (s);Velocidade Média (m/s)
esfera;7.73;1.0;0.134505045;1.33;0.7518796992481203
esfera;7.73;1.0;0.134505045;1.49;0.6711409395973155
esfera;7.73;1.0;0.134505045;1.49;0.6711409395973155
esfera;7.73;1.0;0.134505045;1.4;0.7142857142857143
esfera;7.73;1.0;0.134505045;1.31;0.7633587786259542

Carregar a tabela de vocês

```
# exemplo/sugestão de como carregar seus dados,
 supondo 2 medidas
tabela_dict['Objeto']          = ['esfera', 'esfera']
tabela_dict['Ângulo (°)']      = [8.3, 7.73]
tabela_dict['Distancia (m)'] = [1.0, 1.0]
tabela_dict['Altura (m)']      = [0.25, 0.134505]
tabela_dict['Tempo (s)']       = [0.90, 1.33]
```