

Modeling Match Performance in Elite Volleyball Players: Importance of Jump Load and Strength Training Characteristics

Seminário 3 – Aprendizado Descritivo
2025.1



Historiador

DANIEL SCHLICKMANN BASTOS

GABRIEL CASTELO BRANCO ROCHA

GUILHERME BUXBAUM MARINHO GUERRA

JOSE EDUARDO DUARTE MASSUCATO

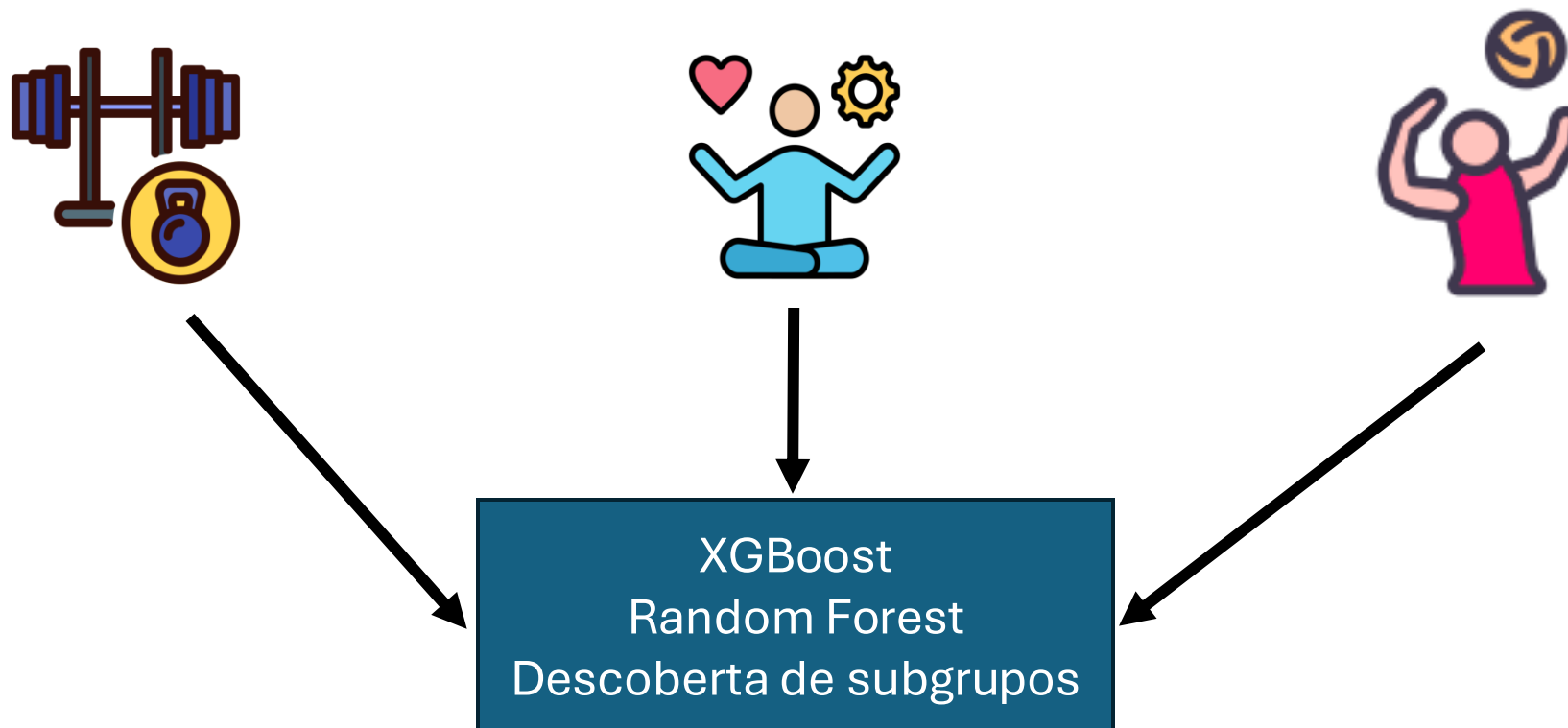
LEONARDO CAETANO GOMIDE

LUCAS MESQUITA ANDRADE

VINICIUS LEITE CENSI FARIA

Objetivo

- Investigar as relações entre **carga de treinamento**, **bem-estar** e **desempenho** de atletas profissionais de vôlei utilizando **XGBoost**, **Random Forest** e **descoberta de subgrupos**



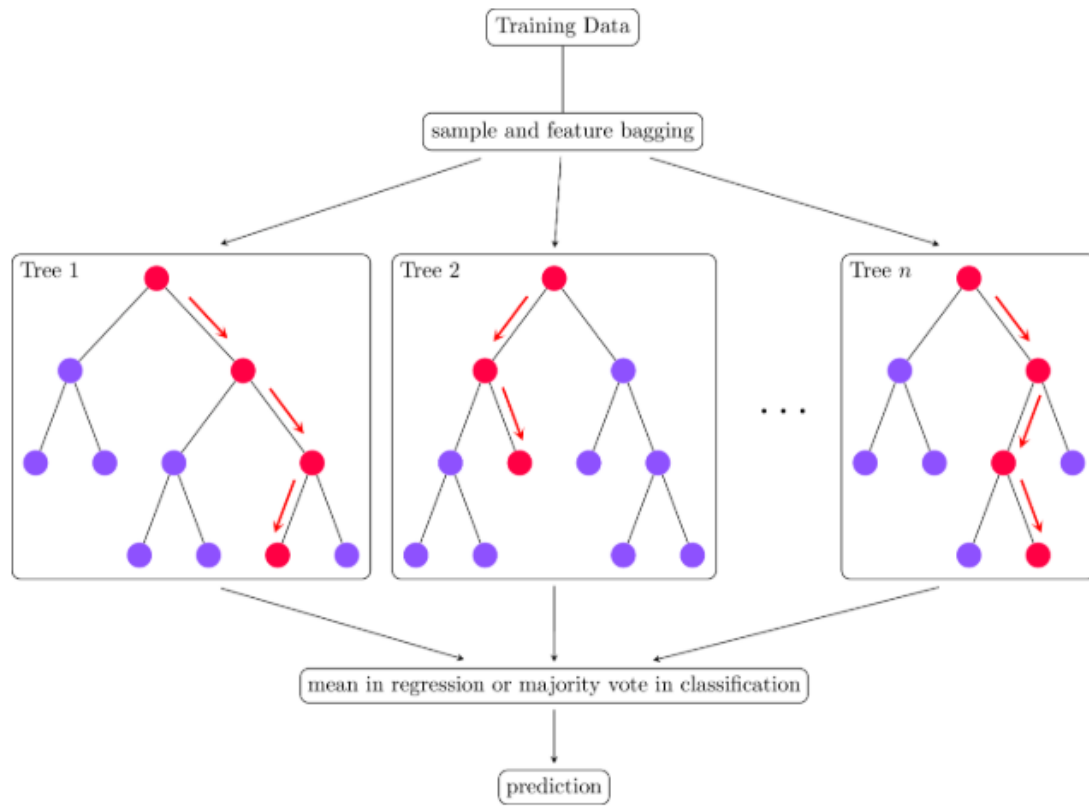
Trabalhos anteriores

- Estudo anteriores investigaram a relação entre **indicadores de performance e desempenho**.
- [2] Indicadores de performance (saque, bloqueio) -> Ranking na liga
- [3, 5] Indicadores de performance -> Chance de vitória
- [4] Indicadores de performance -> Eficiência do ataque (tempo do ataque, tipo do ataque, número de bloqueadores)
- [6] Performance do levantador -> Eficiência do ataque

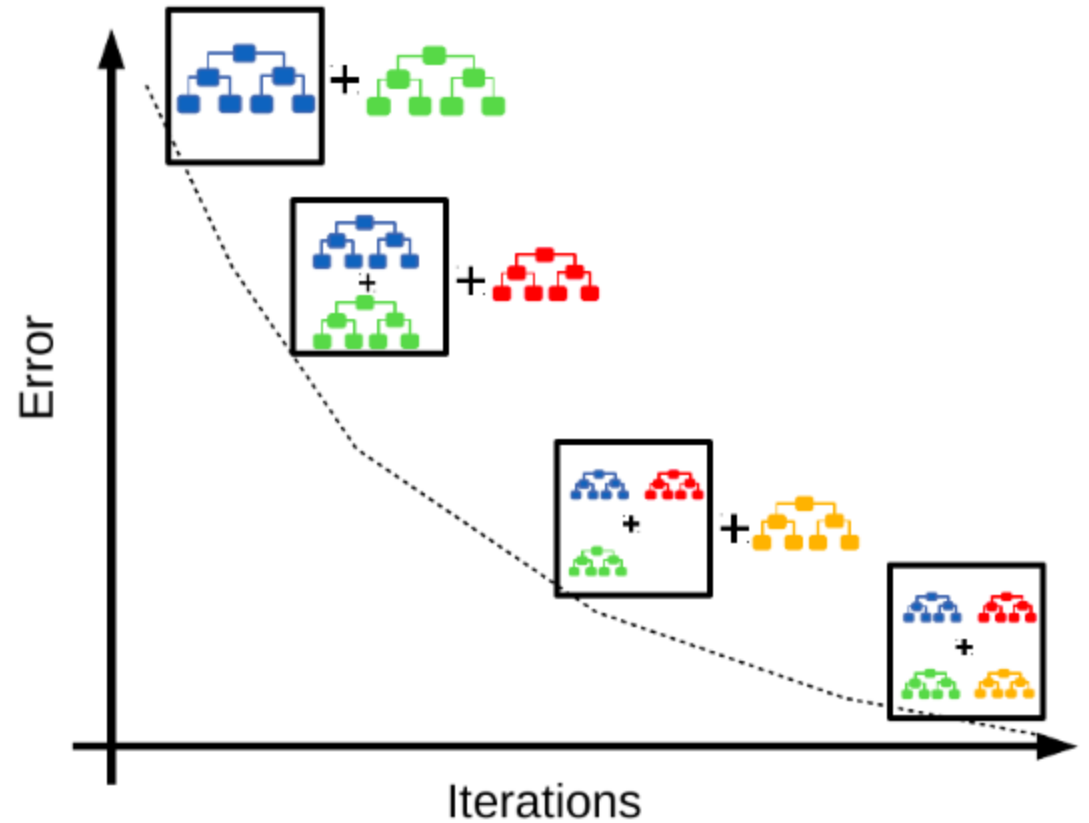
Nenhum estudo avaliou a carga de treinamento, o bem-estar e desempenho juntos

Aprendizado de máquina - Regressão

- Random Forest



- XGBoost



Descoberta de subgrupos

- Descobrir os seletores que melhor distinguem entre boa e má performance
- Vantagem: Funciona bem para datasets pequenos com alta dimensionalidade
 - Times e atletas não querem fornecer seus dados
 - Dataset do artigo: 17 Atletas
- [23] Padrão do ritmo de corredores -> Performance
- [24] Treino de skatistas profissionais -> Performance
- [25] 1 Atleta de tênis -> Ponto ganho
 - Ex: Chance maior de sucesso quando evita backhand

Formalizador

Bernnardo Serafim

Diná Xavier

Gabriel Fadoul

Kênia Gonçalves

Oluwatoyin Joy

Samuel Kfuri

Contexto - Coleta dos dados

Hipótese: os **preditores relacionados a saltos altos** são importantes para o desempenho, uma vez que temos diferentes posições dentro de uma equipe de vôlei e que a altura do salto impacta a efetividade no ataque e o bloqueio é decisivo para o resultado das partidas.

Na coleta de dados: Reuniram dados **físicos, subjetivos e técnicos** para investigar a relação entre **carga de treino, bem-estar e desempenho técnico**

Participantes:

- 25 atletas acompanhados durante **24 semanas** (1 temporada).
- **17 atletas** foram utilizados na aplicação do estudo (dados completos).
- **8 atletas excluídos** por dados insuficientes.

Tipo de Dado	Origem / Ferramenta
Carga de treino externa	Dispositivos de monitoramento (G-VERT)
Carga de treino interna	Questionários subjetivos (ex.: RPE)
Bem-estar percebido	Auto relatos diários dos atletas
Desempenho técnico em jogo	Software <i>Data Volley 4</i> + analistas de vídeo

Preparação dos dados

Ações avaliadas:

Saques	Passes	Ataques	Bloqueios
--------	--------	---------	-----------

- Cada ação dos jogadores foi transformada em **dados quantificáveis (notas)**, para representar a qualidade de cada execução em quadra;
- Cálculo de **probabilidade** que representa a chance de a ação resultar em ponto e gerar vitória no rally = n° de vezes que ocorreu cada ação / total de ocorrências

Para representar o volume, a variação e o impacto dos treinos no corpo do atleta, foram calculados:

Carga Total	$\text{soma}(\text{Peso} \times \text{Esforço (RPE)})$
Monotonia	Variação diária da carga
Strain	$\text{Carga Total} \times \text{Monotonia}$

Ações	Attack	0	0	5	7	8	10
	Block	0	0	4	5	7	10
	Attack Reception	0	4	6	9	9	10
	Freeball	0	5	7	9	9	10
	Pass	0	4	6	8	9	10
	Serve	0	6	4	5	6	10
	Set	0	5	7	9	9	10
		=	/	-	!	+	#
		Notas					



Análise dos dados

- O desempenho de cada jogador foi calculado por **tipo de ação**, a partir da **média das notas recebidas** para aquela ação específica.
- As notas foram utilizadas para avaliar quais características são mais importantes para o desempenho de cada ação, em cada posição, por isto, utilizou-se o **one-hot-encoding** para binarizar a posição dos jogadores;

A construção das variáveis explicativas (preditivas), que representam o contexto físico e psicológico dos atletas antes do jogo, foram **agregadas em diferentes janelas de tempo** e calculadas utilizando informações sobre:

Categoria	Quantidade de Preditores
Saltos (número e altura) - G-VERT	72
Exercícios de força (carga e repetição max.)	81
Bem-estar percebido (fadiga, sono, humor)	48
Carga de treino (total, monotonia, strain)	27
Frequência de sessões de treino	9

No total, foram construídos 237 preditores combinando dados de **carga interna, carga externa e bem-estar.**

Análise Preditiva e Descoberta de Padrões

Objetivo: Modelar o **desempenho técnico** dos atletas (ofensivo e defensivo), identificar os **fatores mais relevantes** e extrair padrões que ajudem a explicar variações de desempenho.

Subgroup Discovery

- Identificar **padrões interpretáveis** que distinguem bons e maus desempenhos;
- Aplicado separadamente para cada tipo de ação (ataque, passe, saque e bloqueio)
- Para garantir validade dos achados, foram usados:
- **Z-score**
- **Teste t**
- **Cohen's d** (tamanho do efeito)

Modelos Preditivos de Aprendizado de Máquina

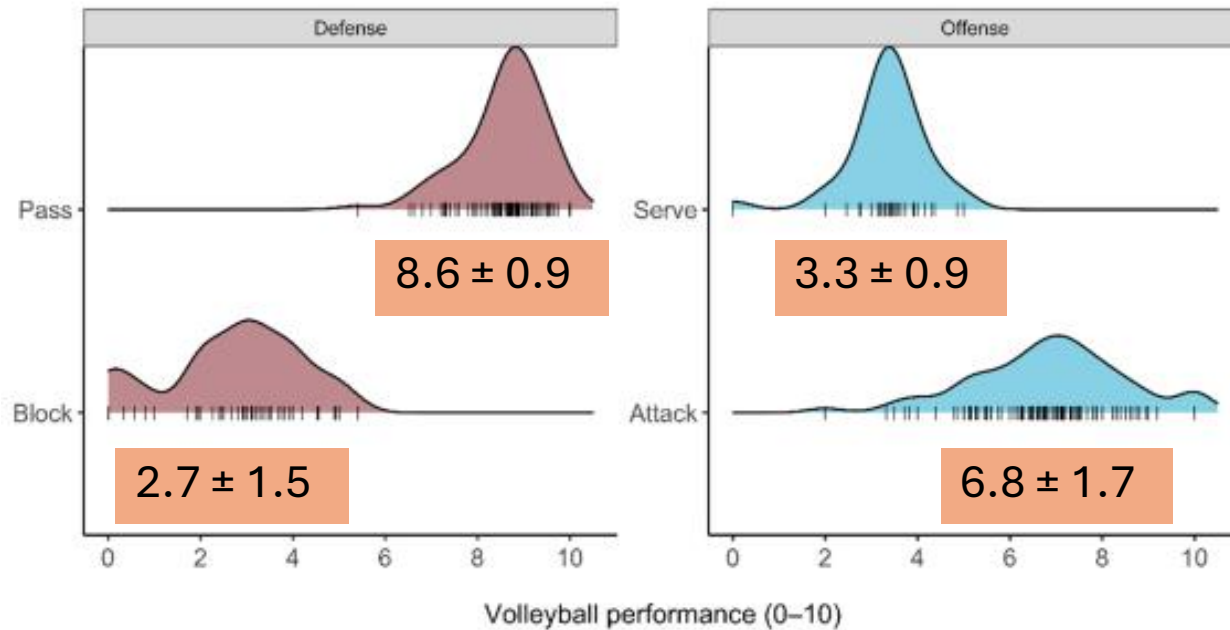
- **Random Forest** e **XGBoost**
- Ajudaram a identificar os **preditivos mais influentes**
- Modelagem dividida em:
 - **Ofensivo:** ataque e saque
 - **Defensivo:** recepção e bloqueio

Avaliação dos Modelos

- Foi utilizado o **Erro Absoluto Médio (MAE)** para medir a performance das previsões;
- **Correlação de Spearman** também foi aplicada para verificar possíveis aumentos ou quedas no desempenho ao longo do tempo.

~~Metodologista~~ > Resultados e discussão

Distribuição da performance média das diferentes categorias de ação:



Avaliação de mudanças de performance ao longo do tempo:

Spearman's rank correlation (entre performance da partida e o índice da partida)

Defensivo: -0.02 ($p = 0.19$)

Ofensivo: -0.19 ($p = 0.02$)

Modelos de aprendizado de máquina

Métricas da qualidade das predições para cada tipo de ação (ofensiva ou defensiva):

Action Type	Model	MAE (95% CI)	Difference in MAE	<i>p</i> -Value	Cohen's <i>d</i> (95% CI)	Effect Size
Offense	Random Forest	0.91 (0.62–1.19)	–46.8%	$p < 0.001$	0.79 (0.47–1.18)	Medium
	XGBoost	1.09 (0.78–1.41)	–36.3%	$p < 0.01$	0.58 (0.23–0.99)	Medium
	Action Model	1.04 (0.75–1.32)	–39.2%	$p < 0.001$	0.66 (0.35–1.04)	Medium
	Baseline	1.71 (1.38–2.05)				
Defense	Random Forest	1.15 (0.79–1.51)	–59.4%	$p < 0.001$	1.47 (1.10–1.98)	Large
	XGBoost	0.75 (0.50–1.00)	–73.5%	$p < 0.001$	2.09 (1.63–2.78)	Large
	Action Model	0.79 (0.58–1.00)	–72.1%	$p < 0.001$	2.14 (1.63–2.88)	Large
	Baseline	2.83 (2.47–3.20)				

MAE (Mean Absolute Error): Erro absoluto médio, a distância entre o valor predito e o valor real da performance de uma partida.

Busca de subgrupos

Action Type	Description Subgroup	Z-Score Sign	Size	p-Value	Cohen's <i>d</i> (95% CI)	Effect Size
Passes	Jumps_above65_ std7 \geq 9.75	-	17.2%	$p = 0.01$	1.03 (0.48–1.62)	Large
	Jumps_above65_ avg14 \geq 11.6	-	10.3%	$p = 0.01$	1.33 (0.64–2.08)	Large
Blocks	No significant results					
Serve	No significant results					
Attacks	LowerWeight_ firstquantile28 \geq 90	+	15.6%	$p = 0.01$	0.83 (0.35–1.34)	Large
	JumpHeight_ thirdquantile28 \leq 59	-	12.3%	$p = 0.001$	1.01 (0.47–1.58)	Large
	FullbodyWeight_ std28 \geq 17.6	-	23.0%	$p = 0.001$	0.76 (0.35–1.19)	Medium
	WeightPrct_ Upperbody_ avg28 \geq 0.90	-	28.7%	$p = 0.001$	0.67 (0.30–1.07)	Medium
	Jumps_above65_ std28 \leq 2.24	-	10.7%	$p = 0.02$	0.99 (0.42–1.59)	Large
	WeightPrct_ Upperbody_ firstquantile28 \geq 0.87	-	32.0%	$p = 0.03$	0.56 (0.20–0.94)	Medium

Altura dos pulos

Pesos na
musculação

Discussão

Preditores mais importantes:

- **Melhor ataque:** Carga de treinamento de força de membros inferiores (4 sem.) (**maiores pulos**)
- **Pior ataque:** Altura dos saltos (4 sem.) (**menores pulos**) e Excesso de carga em parte superior (**menos controle**)
- **Pior passe:** Altura média dos saltos (1 semana) ("**reduced freshness**")
- Bem estar não teve impacto.

Limitações e trabalhos futuros:

- Número baixo de jogadores para coletar os dados
- Os melhores modelos de aprendizado performaram tão bem quanto modelos que se baseavam na média dos *scores* do tipo da ação
- Avaliar variações nos cronogramas de treinamento
- Pesos para importância da partida ou nível do oponente

Assessor Social

Aplicação: Esportes

O artigo mostra a aplicação de métodos de descoberta de subgrupos para o melhor desempenho de atletas, por meio de:

- Personalização de treinos
- Análise de performance
- Delimitação de treinos úteis e menos úteis
- Medidas de previsão de desempenho em partida a partir do treino

Como dito no resumo do artigo:

"Differences in findings with respect to passing and attack performance suggest that elite volleyball players can improve their performance if training schedules are adapted to the position of a player. "

Aplicação: Desempenho em geral

Nota-se que não necessariamente precisamos limitar a análise para análise de esportes. Podemos, por exemplo, fazer as seguintes trocas:

- Treino (peso, repetições, etc) -> Estudos; Tempo e forma de trabalho;
- Resultados na partida -> Entregas de trabalhos e tarefas

O algoritmo pode ser generalizado para ambientes de estudos, trabalhos e hobbies pessoais, permitindo uma análise de quais as melhores formas de dedicação para alcance de melhores resultados.

"Empregados que passaram 20% das horas em repouso, 55% das horas trabalhando e 15% das horas estudando tiveram entregas mais pontuais."

"Empregados que variavam bastante o seu horário de entrada e saída tendem a terem resultados piores na avaliação das entregas."

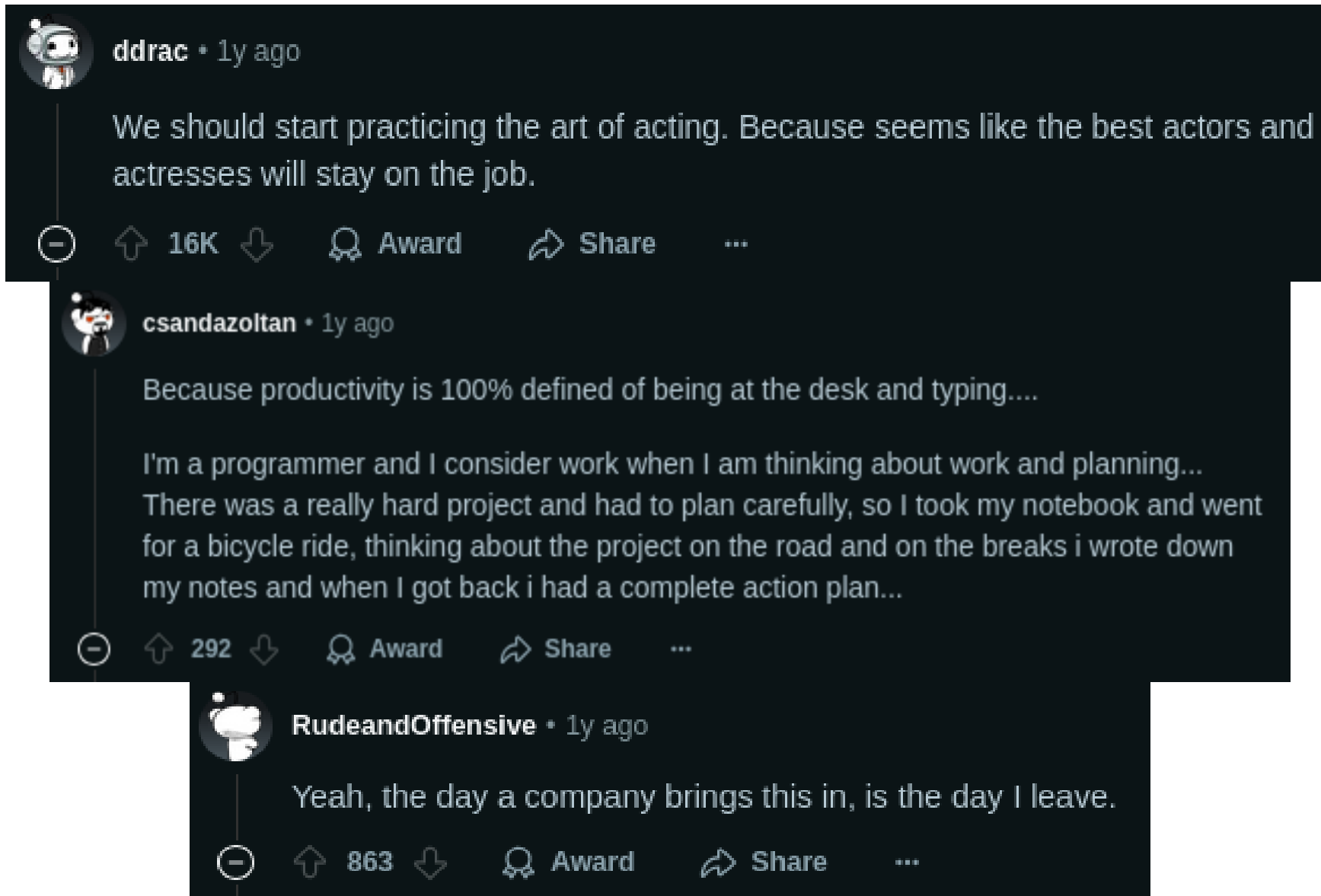
Risco: Controle

Na busca de terem os dados mais atualizados, empresas e organizações podem acabar usando os estudos analíticos como desculpa para controlarem funcionários e atletas. Considere o caso à direita como exemplo:

Isso leva ao descontentamento de funcionários, considere os seguintes comentários:



Risco: Controle



Risco: Privacidade

O monitoramento constante das pessoas pode afetar a sua privacidade, especialmente quando:

- Os dados forem usados sem consentimento claro ou transparência.
- Houver coleta em excesso (dados sensíveis, pessoais).
- Os dados monitoram a pessoa de forma a remover a privacidade em ambiente privados.
- Houver vazamento de dados pessoais.

Hacker

Henrique Rotsen

Alexandre Cassimiro Silva Araújo

Wesley Marques Daniel Chaves

Caíque Bruno Fortunato

Arthur Yochio R. Codama

Tentando obter os dados

O artigo menciona que os dados não podem ser compartilhados publicamente, pois são propriedade da NeVoBo (Federação Holandesa de Voleibol). No entanto, os dados estão disponíveis mediante solicitação ao autor correspondente.

- No artigo tem somente o e-mail do primeiro autor: **Arie-Willem de Leeuw**
 - O professor mudou de universidade, então o e-mail não existe mais
- Procuramos o e-mail atual do Arie-Willem de Leeuw: ainda sem resposta

O detentor dos dados é o segundo autor: **Rick van Baar**, ele era treinador, mas saiu da instituição NeVoBo.

- Após muita busca, foi encontrado o e-mail: ainda sem resposta

Tentando obter os algoritmos

Embora os dados brutos não estejam abertos, talvez o código-fonte para o método (os algoritmos de aprendizado de máquina e a engenharia de recursos) possa estar em um repositório. O artigo menciona o uso de **XGBoost**, **random forest regression** e **subgroup discovery**.

- Buscamos o repositório no GitHub, mas não obtivemos sucesso.

Porém, o uso dos algoritmos pode ser exemplificado:

Random Forest Artigo: [Link do artigo](#)

Git: [Link do git](#)

Biblioteca Py: *from sklearn.ensemble import RandomForestClassifier*

Exemplo de uso: [Link do git](#)

XGBoostArtigo: [Link do artigo](#)

Git: [Link do git](#)

Características dos dados originais

Desempenho de Partida:

- Avaliações de ações de voleibol (ataque, bloqueio, recepção...) em uma escala de **0 a 10**.

Carga de Salto:

- Número de saltos e alturas de salto (total, baixos < 50cm, médios entre 50-65cm, altos > 65cm).

Treinamento de Força:

- Pesos de exercícios (em kg absolutos ou % de 1-RM) divididos em corpo inteiro, parte inferior e parte superior do corpo.

Carga de Treinamento Percebida (RPE):

- Usando a escala CR10 para sessões de voleibol e força.

Bem-estar Percebido (Wellness):

- Medidas de fadiga, qualidade do sono, horas dormidas e humor em uma escala **Likert** de 10 pontos.

Períodos de Agregação:

- Dados agregados em janelas de **7, 14 ou 28** dias anteriores à partida, com funções como primeiro quartil, média, terceiro quartil e desvio padrão.

Como reproduzir?

Procure por conjuntos de dados de voleibol abertos e disponíveis publicamente que sejam similares em natureza aos dados usados no artigo.

O objetivo não é replicar exatamente os resultados, mas demonstrar como o método poderia ser aplicado e os tipos de padrões que poderiam ser encontrados.

Implementação do algoritmo não faz parte do escopo.

Simulando o Dataset

```
dadosArtigo5.ipynb
Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda
Comandos + Código + Texto ▶ Executar tudo

# Notebook de Análise com Random Forest e XGBoost
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, r2_score
from scipy.stats import ttest_ind
import matplotlib.pyplot as plt
import seaborn as sns

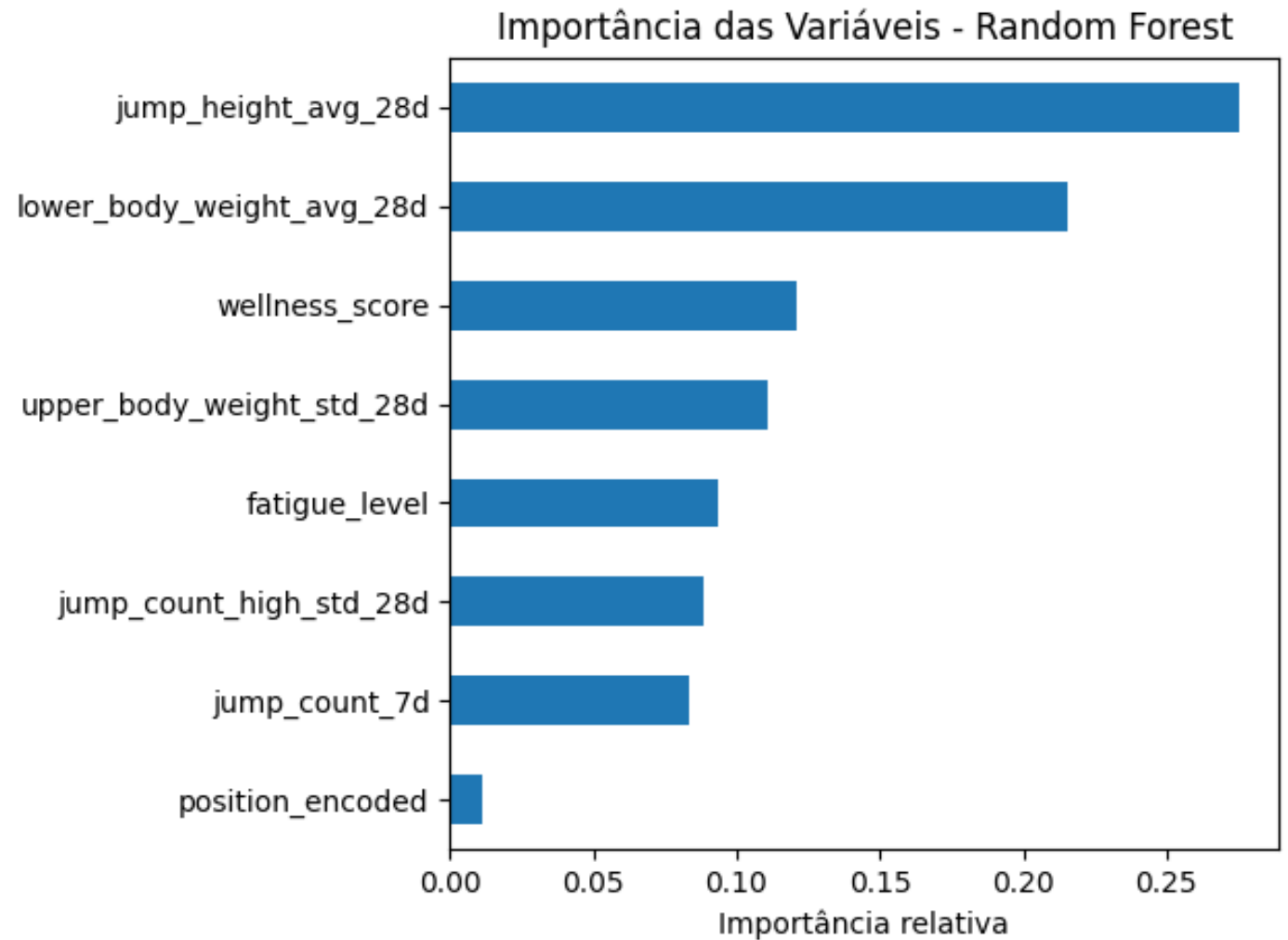
# Simulando dataset (baseado no artigo)
np.random.seed(42)
n = 300

data = pd.DataFrame({
    "player_id": np.random.randint(1, 10, size=n),
    "jump_count_7d": np.random.poisson(120, size=n),
    "jump_height_avg_28d": np.random.normal(62, 4.5, size=n),
    "jump_count_high_std_28d": np.random.normal(3.2, 1.1, size=n),
    "lower_body_weight_avg_28d": np.random.normal(95, 8, size=n),
    "upper_body_weight_std_28d": np.random.normal(6, 2, size=n),
    "wellness_score": np.random.uniform(5, 10, size=n),
    "fatigue_level": np.random.uniform(1, 10, size=n),
    "position_encoded": np.random.choice([0, 1], size=n)
})

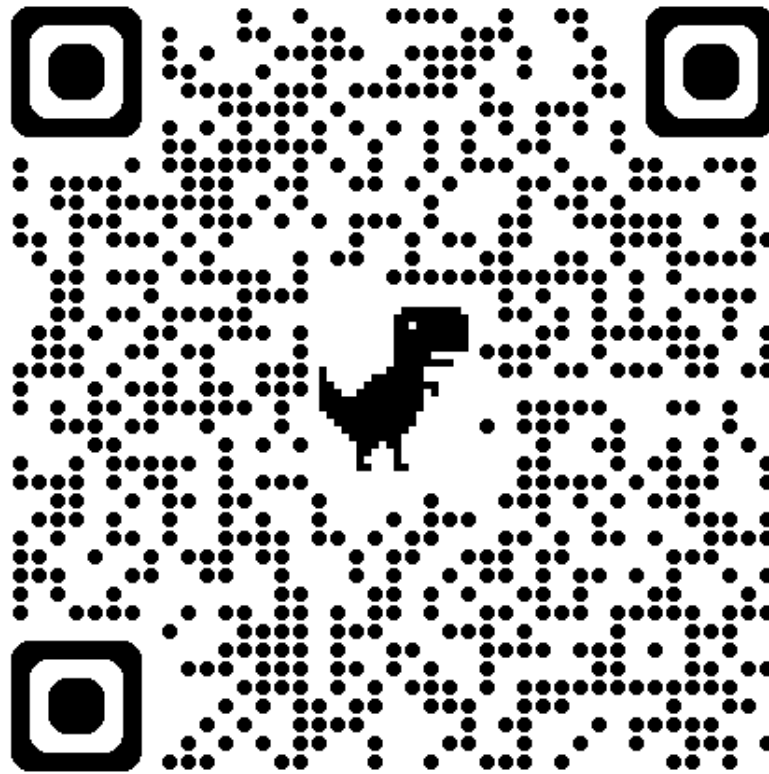
data["performance_score"] = (
    0.05 * data["jump_height_avg_28d"] +
    0.02 * data["lower_body_weight_avg_28d"] -
    0.03 * data["upper_body_weight_std_28d"] +
    0.04 * data["wellness_score"] -
    0.02 * data["fatigue_level"] +
    np.random.normal(0, 0.5, size=n)
)

data["performance_score"] = data["performance_score"].round(1)

# Separar features e target
```



Link para o Colab



Referências

Referências bibliográficas