

TRABALHO DE INTELIGÊNCIA ARTIFICIAL

***IA QUE FAZ TREINOS POR
SEMANA***

Etapas da inteligência artificial

Seleção do dataset:

O dataset usado como base seria o dataset “Top 50 Exercises for Your Body” do site [Kaggle](#). Nele, são apresentadas colunas de ‘Nome’, ‘Repetições’, ‘Séries’, ‘Benefícios’, ‘Calorias queimadas a cada meia hora’, ‘Grupo muscular’, ‘Equipamento’ e ‘Dificuldade’.

Objetivo do trabalho:

O objetivo inicial seria um site onde o usuário digitaria qual o nível dele e qual grupo muscular ele gostaria de trabalhar. Por questões de tempo, o objetivo foi simplificado para montar um treino semanal de 5 dias focado na queima de calorias.

Como ele faz a seleção dos exercícios:

3.1 O modelo:

É utilizada para essa IA uma rede neural com três camadas ocultas, com 64, 64 e 32 neurônios, utilizando a *Mean Squared Error* (MSE) para a validação de perda.

O treinamento é feito em 1000 épocas, com um *batch size* de 5.

3.2 Adaptação do modelo:

Poucas adaptações foram feitas no modelo. Foi necessário substituir os valores nulos por 0 para evitar erros ao passar pelo modelo.

3.3 X e Y:

O X é a nossa entrada, ou seja, para esse caso, nome e calorias.

O Y seria o valor-alvo (calorias).

Os valores de X, caso não sejam numéricos, são tratados como NaN, convertidos para 0 e passados para *float32*, para evitar erros no modelo.

A variável Y é transformada em um vetor NumPy contendo os valores-alvo no formato de números *float32*.

Gerar o planejamento da semana:

4.1 Objetivo:

A função `generate_week_plan` tem como objetivo criar um plano semanal de exercícios, considerando os valores previstos de calorias queimadas para cada exercício. O plano é estruturado para cinco dias, com cinco exercícios por dia, sem repetições.

4.2 Adicionar previsões ao DataFrame:

Os valores previstos de calorias queimadas são adicionados ao DataFrame na coluna `Predicted Calories Burned`, priorizando aqueles que queimam mais calorias.

4.3 Ordenação dos exercícios:

O DataFrame é ordenado em ordem decrescente com base na coluna `Predicted Calories Burned`.

4.4 Seleção das colunas relevantes:

Apenas as colunas essenciais para o plano semanal são extraídas do DataFrame. Essas colunas incluem:

- Nome do exercício;
- Grupo muscular trabalhado;
- Séries e repetições;
- Calorias queimadas previstas;
- Nível de dificuldade.

4.5 Construção do plano diário:

Para cada dia:

- Um plano diário é inicializado com uma lista vazia;
- Os exercícios são iterados em ordem de calorias queimadas;

- Apenas exercícios que ainda não foram usados são adicionados ao plano do dia;
- O nome do exercício é registrado para evitar repetições nos dias seguintes;
- O plano diário é finalizado quando atinge o limite de exercícios definido (cinco por dia).

4.6 Retorno do plano:

Após gerar o plano semanal para os cinco dias, a função retorna:

- **week_plan**: Lista contendo os exercícios organizados por dia;
- **df_sorted**: DataFrame ordenado por calorias queimadas, útil para exibir informações complementares.

Display de informações:

Essa função seria responsável apenas por mostrar quais seriam os exercícios do dia, trazendo todas as informações sobre eles.

Gráfico:

Essa função é responsável por montar o gráfico de calorias gastas durante toda a semana, trazendo a informação diária de calorias.

Criação do menu principal:

A função `main_menu` é o ponto de entrada do sistema, responsável por integrar as etapas anteriores em uma interface interativa para o usuário.

7.1 Boas-vindas e configuração inicial:

O sistema exibe uma mensagem de boas-vindas e define o caminho do arquivo CSV que contém os dados dos exercícios.

Variáveis e caminhos:

- **file_path**: Local do dataset. Deve ser ajustado conforme o ambiente do usuário.

7.2 Carregamento e preparação dos dados:

A função `load_and_prepare_data` é chamada para processar o dataset, retornando:

- **df**: DataFrame processado;
- **X**: Entradas do modelo (características dos exercícios);
- **y**: Saída-alvo (calorias queimadas).

7.3 Divisão dos dados para treinamento e teste:

Os dados são divididos em conjuntos de treinamento e teste usando `train_test_split`, com 20% dos dados reservados para teste.

7.4 Treinamento do modelo:

A função `build_and_train_model` é usada para criar e treinar a rede neural no conjunto de treinamento, retornando o modelo treinado.

7.5 Geração de previsões:

O modelo treinado é usado para prever as calorias queimadas por exercício no conjunto completo de dados.

7.6 Criação do plano semanal:

A função `generate_week_plan` utiliza as previsões para criar o plano semanal de exercícios, retornando:

- **week_plan**: Lista estruturada com os exercícios organizados por dia;
- **df_sorted**: DataFrame ordenado por calorias previstas.

7.7 Interface interativa:

O menu principal apresenta quatro opções para o usuário:

- **Opção 1**: Visualizar o plano de treino semanal, chamando a função `display_week_plan` para exibir o plano;

- **Opção 2:** Exibir os cinco exercícios com maior queima calórica, mostrando a tabela com as informações mais relevantes;
- **Opção 3:** Exibir um gráfico de calorias queimadas por dia, gerado pela função `plot_calories_burned`;
- **Opção 4:** Encerrar o programa.

7.8 Tratamento de erros e controle de fluxo:

- O sistema valida a entrada do usuário;
- Para entradas inválidas, exibe uma mensagem de erro e retorna ao menu principal.

7.9 Encerramento:

Quando a opção "4" é selecionada, o programa exibe uma mensagem de despedida e encerra o loop, finalizando o sistema.