



Relatório do projeto de Aplicações Avançadas de Instrumentação Biomédica

Mestrado Integrado em Engenharia Biomédica

Ana Lopes nº 51880

David Trindade nº 62268

Docentes: Professor Hugo Gamboa

Professor Pedro Vieira

2022/2023

Índice

1. Introdução e contextualização do projeto	3
2. Aquisição dos dados de treino	3
3. Extração de Características	3
4. Seleção de características	4
5. Classificador	5
7. Interface de visualização dos dados	6
8. Capacidades e limitações do sistema	6
9. Conclusão	6

1. Introdução e contextualização do projeto

No âmbito da unidade curricular de Aplicações Avançadas de Instrumentação Biomédica foi proposta a implementação de uma aplicação integrada de instrumentação, capaz de adquirir medidas físicas para serem processadas, analisadas e classificadas através de *Machine Learning* (ML). Pretendia-se, adicionalmente, que a comunicação do sistema fosse realizada à distância e que a interação com o utilizador fosse executada em ambiente móvel.

O principal objetivo do projeto foi desenvolver um classificador ML capaz de distinguir entre três tipos diferentes de movimentos de futebol, realizados com uma bola de futebol, designados *pentear*, *dentro pé* e *fora pé* (também existia uma classe de rejeição).

2. Aquisição dos dados de treino

Os dados para este projeto foram adquiridos através da aplicação *Sensors Streamer* disponível para Android. Os sensores utilizados foram o acelerómetro (*accx*, *accy*, *accz*) e o giroscópio (*gyrx*, *gyry*, *gyrz*) do *smartphone*, tendo sido seleccionadas as suas componentes nas direções x, y e z. Na aplicação foi selecionado um pacote de dados com a taxa de amostragem de 60 ms (*medium packet*) que corresponde a uma frequência de aquisição de aproximadamente 16,7 Hz.

Durante a aquisição das medidas, o telemóvel fica em contacto com a anca da pessoa, na vertical e com o ecrã voltado para fora. Foram recolhidas 25 amostras de cada um dos 3 movimentos mais 25 amostras para a classe de rejeição, na qual o sujeito se encontra parado. O código do projeto usado para aquisição de dados encontra-se nos ficheiros *dentro_pe.py*, *fora_pe.py* e *pentear.py*, em que cada ficheiro diz respeito à aquisição de uma maneira diferente de condução de bola.

3. Extração de Características

Para a definição das características a utilizar é importante ter em conta que o número de amostras recolhidas não é elevado. O uso de um número excessivo de características poderá conduzir a um mau desempenho do modelo, uma vez que a relação entre a quantidade de informação a interpretar e os dados existentes seria desproporcional.

Inicialmente, procedeu-se a uma observação dos dados obtidos com o objetivo de compreender quais as características que possivelmente seriam mais adequadas e relevantes, para adicionar valor ao classificador. Depois, foram selecionadas, entre o grupo de características disponíveis na biblioteca *tsfel*, as características consideradas mais relevantes. Esta etapa é concretizada através da edição do dicionário *python*, guardado depois no ficheiro *features.json*.

No total foram usadas 12 características dos domínios estatístico e temporal, aplicadas às 6 séries temporais *accx*, *accy*, *accz*, *gyrx*, *gyry*, *gyrz*, o que resulta num total de 72 características passíveis de serem fornecidas ao modelo no processo de treino. Estas são as seguintes:

- Domínio temporal
 - Máximo
 - Mínimo
 - Média
 - Mediana
 - Variância
 - Desvio Padrão
 - Curtose
 - Assimetria
- Domínio estatístico
 - Energia Absoluta
 - Derivada
 - Autocorrelação
 - Distância Pico a Pico

4. Seleção de características

É muito importante determinar quais as características cuja informação fornecida, acresce mais valor ao classificador. Esta etapa tem um elevado impacto no desempenho do modelo, reduzindo o risco de *overfitting* ao conjunto de treino (menor redundância nos dados), tempo de treino (menor complexidade computacional do algoritmo) e o erro.

Com recurso à biblioteca *tsfel* esta fase foi implementada do seguinte modo:

- 1) Remoção de características com elevada correlação (Método de *Pearson*)
- 2) Remoção de características com reduzida variância (apresentam um *output* semelhante para todas as amostras e por isso não são úteis para a distinção entre classes).

Posteriormente a esta seleção, das 72 características iniciais apenas 49 são consideradas relevantes para o treino do algoritmo ML, tendo sido removidas 23 características.

5. Classificador

O modelo desenvolvido permite identificar 3 classes e uma classe de rejeição.

- 1) Condução da bola com a parte interna do pé (*dentro pé*)
- 2) Condução da bola com a parte externa do pé (*fora pé*)
- 3) Pentear a bola
- 4) Outro

Neste projeto já se sabia *à priori* a classe de cada amostra, logo foi treinado um classificador de aprendizagem supervisionada.

Cada classe é formada por 25 amostras, existindo no total 100 amostras. A constituição dos conjuntos de treino e teste resulta respetivamente de 80% e 20% dos elementos de cada classe. Logo, o conjunto de treino tem 80 amostras e o de teste tem 20 amostras.

Foram treinados dois tipos de classificadores: *Random Forest* e *Decision Tree*, ambos com recurso à biblioteca *scikit-learn*.

Ambos os classificadores obtiveram uma *accuracy* de 100%. O classificador *Random Forest* foi obtida uma *accuracy* de 100.0%, enquanto para o classificador *Decision Tree* foi obtida uma *accuracy* de 100% (*max_depth* de 5). Esta métrica representa a proporção de amostras do conjunto de teste que foram corretamente classificadas. A elevada *accuracy* dos classificadores treinados pode ser explicada devido ao baixo número de amostras adquiridas.

O classificador escolhido para este trabalho foi o *Random Forest*. Este é um meta classificador que treina um determinado número de Árvores de Decisão em várias

subamostras do conjunto de treino e usa a média das classificações para melhorar a performance e prevenir *over-fitting* do modelo.

Uma forma melhor de avaliar o desempenho de um classificador é através da observação da matriz de confusão, que permite, por exemplo, determinar o número de vezes que a instância da classe pentear foi classificada como classe *dentro_pé*. Através da matriz de confusão é possível obter os *true positive (TP)* e *true negative (TN)*, que correspondem às classificações corretas, e os *false positive (FP)* e *false negative (FN)*, que correspondem às classificações incorretas. O acesso a esta informação permite compreender onde o classificador falha mais e que classes ele está a confundir.

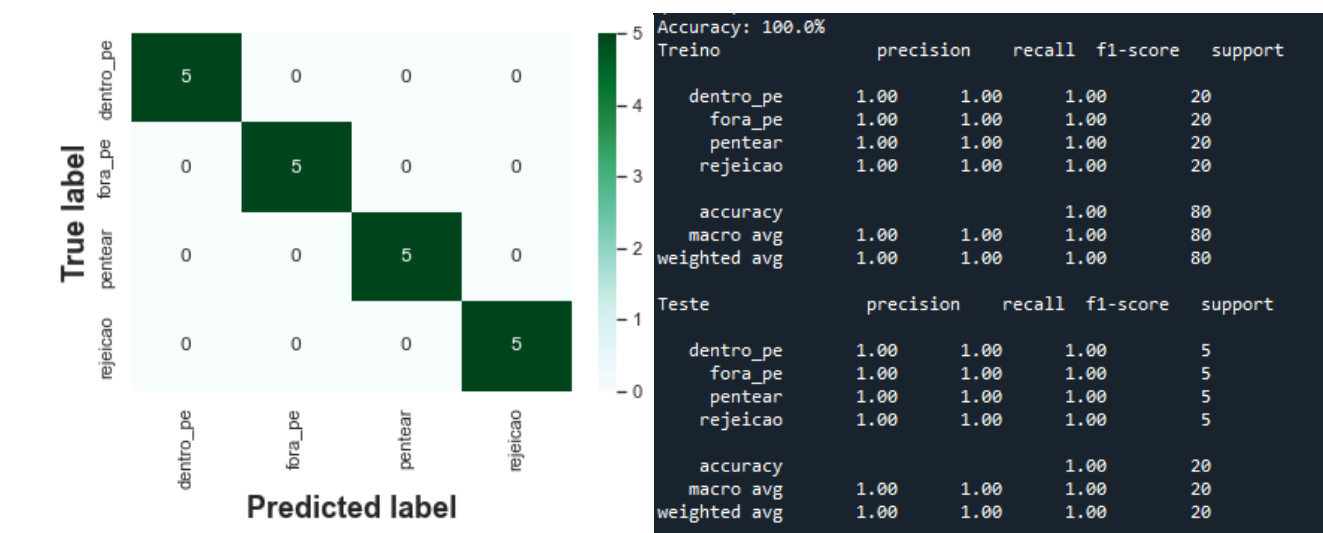


Figura 1 - Classificador: A - Random Forest; B - Matriz de confusão

7. Interface de visualização dos dados

A interface desenvolvida, representada na Figura 2, permite controlar o início da aquisição através do botão *start* e a classificação dos dados adquiridos através do botão *classify*, cujo resultado, a classe, é visível na interface. É possível, também, observar uma representação gráfica da magnitude do acelerómetro em função do tempo, durante o processo de aquisição de dados.

Classificação de movimentos de Futebol através de WebSockets com Flask

Projeto desenvolvido no âmbito da cadeira de AAIB

Raspberry IP:

"Movimento executado pelo sujeito: pentear"

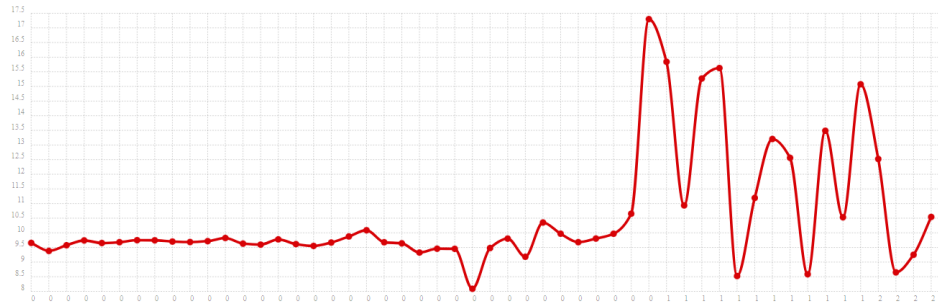


Figura 2 - Interface desenvolvida

8. Capacidades e limitações do sistema

Como já foi mencionado anteriormente, o sistema tem a capacidade de distinguir três tipos de movimentos diferentes, classificando-os e demonstra o gráfico referente aos dados recolhidos pelo acelerómetro, sendo estas as principais capacidades do sistema. Contudo, o sistema não consegue fazer mais que três aquisições seguidas, sendo esta a principal limitação do trabalho desenvolvido.

9. Conclusão

O programa tem a capacidade de identificar corretamente cada um dos três movimentos propostos (*pentear*, *dentro pé* e *fora pé*), bem como identificar a classe de rejeição, sendo que com isto é possível concluir que a comunicação foi feita de maneira correta e que o classificador utilizado teve uma *accuracy* elevada. Apesar da boa aquisição e classificação dos dados, o programa apenas consegue fazer três aquisições de seguida, pelo que este seria um aspeto que poderia ser melhorado. Outra melhoria, passaria por desenvolver uma interface mais detalhada e que permitisse ao utilizador mais interatividade.