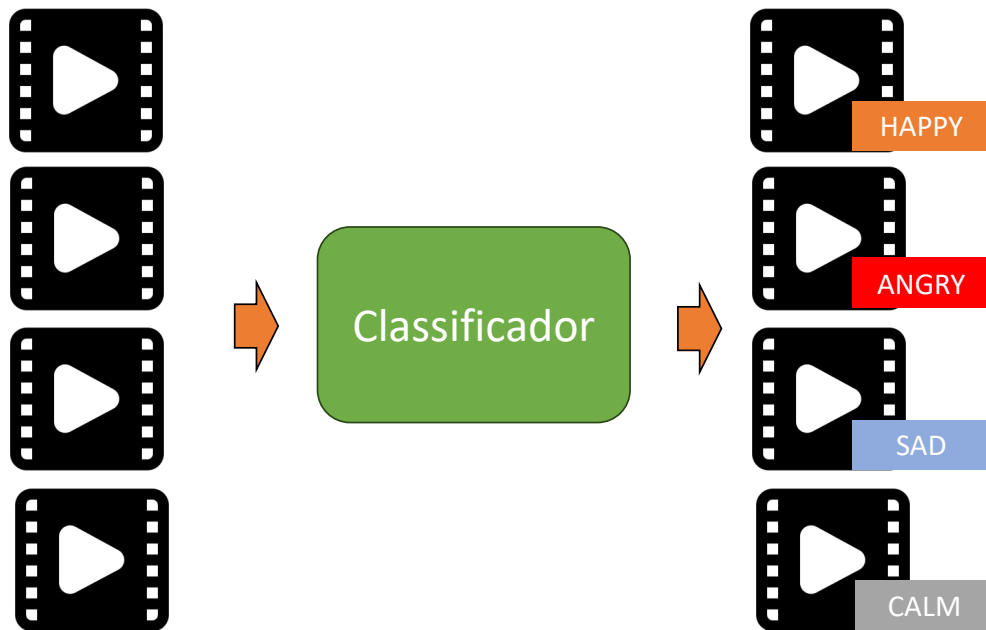


Tópicos de Ciência de Dados

Trabalho Pratico nº1

Análise de Expressões Faciais em Vídeo



Introdução

Período de execução: 10 semanas

Esforço extra-aulas previsto: 32h

Datas de Metas:

- **Entrega da componente A: 10 de Novembro 2023, 23h59 inforestudiante**
- **Teste prático componente A: 15 de Novembro 2023, aula PL**
- **Entrega da componente B: 7 de Dezembro 2023, 23h59 inforestudiante**
- **Defesa da componente B: 12 de Dezembro 2023, aula PL**

Objetivo: O objetivo central deste trabalho prático é que o aluno exercite conceitos centrais de um pipeline de análise de dados, passando pelas fases de preparação de dados, a sua limpeza, a extração de características descritivas, a sua seleção/redução e a avaliação.

Trabalho Prático

O problema proposto no presente trabalho prático é um problema típico de análise de requisitos que comumente se deparam os Data Scientists. O contexto do exercício proposto é o reconhecimento de diferentes expressões faciais em vídeos. Este é um contexto com uma importância crescente em múltiplas situações, abrangendo, por exemplo, a área do marketing ou *user experience*, onde através da captura da expressão do utilizador/cliente se consegue aferir a sua recetividade ao produto/aplicação.

Independentemente do problema específico e das suas potenciais aplicações, o presente contexto irá permitir exercitar e interiorizar conceitos centrais em qualquer *pipeline* de análise dados com que um *data scientist* se confronta: dado um volume (elevado) de dados

reais, analisar e identificar um conjunto de atributos que permitam a identificação de um conjunto de estados distintos.

No presente trabalho iremos usar uma variação do dataset RAVDESS – Ryerson Audio-Visual Database of Emotional Speech and Song¹, nomeadamente o RAVDESS Facial Landmark Tracking². Este dataset foi adquirido através da gravação vídeo de 24 atores a dizerem ou cantarem 2 frases representando 8 emoções diferentes (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised), com dois níveis de intensidade diferente (exceto para a emoção neutra).

Os clips de vídeo representados pelos atores foram depois processados de forma a extrair, para cada frame, as coordenadas x e y de um conjunto de 68 landmarks faciais, conforme ilustrado na Figura 1. Note que no dataset existem outras informações que poderiam ser de interesse para um sistema deste tipo, como a direção do olhar, inclinação da cabeça, etc, mas o trabalho vai focar explicitamente sobre estes 68 landmarks.

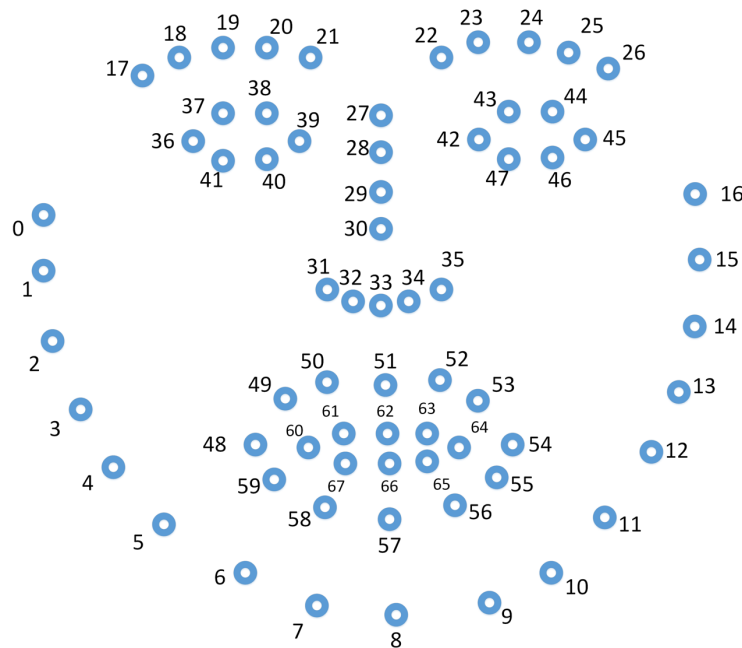


Figura 1 - Visualização dos 68 landmarks faciais

¹ <https://zenodo.org/record/1188976>

² <https://zenodo.org/record/3255102>

Faça download do dataset disponível no link:
https://zenodo.org/record/3255102/files/FacialTracking_Actors_01-24.zip?download=1

O dataset é composto por um total de 2452 ficheiros CSV, cada um relativo a um vídeo de uma expressão facial. No ficheiro CSV, cada linha corresponde a um *frame*/instante de tempo do vídeo e as colunas correspondem às métricas extraídas do vídeo. Das cerca de 700 variáveis disponíveis, vamos focar o trabalho apenas nas coordenadas x e y (2D) dos 68 landmarks faciais, representadas pelas colunas com o nome x_0 a x_{67} e y_0 a y_{67} . O binómio (x_k, y_k) representa as coordenadas xy do landmark k, para $0 \leq k \leq 67$.

O nome dos ficheiros CSV representa um identificador numérico com 7 partes, respetivamente³:

- Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
- Vocal channel (01 = speech, 02 = song).
- Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
- Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.
- Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
- Repetition (01 = 1st repetition, 02 = 2nd repetition).
- Actor (01 to 24. Odd numbered actors are male, even-numbered actors are female).

Por exemplo, o ficheiro 02-01-06-01-02-01-12.mp4 apresenta as seguintes características:

1. Video-only (02)
2. Speech (01)
3. Fearful (06)
4. Normal intensity (01)
5. Statement "dogs" (02)
6. 1st Repetition (01)
7. 12th Actor (12)
 - Female, as the actor ID number is even.

³ Adaptado da descrição do dataset em <https://zenodo.org/record/1188976>.

No fundo, os identificadores de interesse são apenas o terceiro (emotion, a nossa classe de interesse) e o ator (necessário para alguns cuidados adicionais na divisão dos dados).

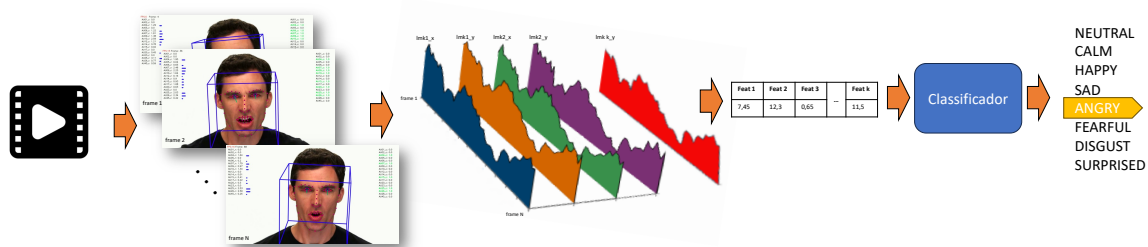



Figura 2 - Pipeline de análise e classificação de um vídeo. Do vídeo original foram identificados os landmarks para os diferentes frames, obtendo-se uma timecourse de cada coordenada de cada landmark. Esses timecourses são tratados e dos mesmos extraídos features identificativas do vídeo. Essas features, depois de reduzidas e selecionadas, são depois fornecidas a um classificador que atribui uma emoção ao vídeo com base nas features identificadas.

Nota: Além dos ficheiros CSV com estas métricas, podem visualizar os vídeos com os marcadores sobrepostos no link <https://zenodo.org/record/3255102>, nos ficheiros “Tracked_Video_Actor_*.zip”. Apenas para visualização, os vídeos não farão parte da análise levada a cabo neste projeto.



Módulo A: Elaboração de um conjunto de scripts e funções em Python, NumPy e SciPy para realizar as tarefas de preparação dos dados e *Feature Engineering*.

1. Crie um IPython notebook com o nome ‘XXXX_YYYY_EA2023.ipynb’, em que XXXX e YYYY devem ser substituídos pelos números de aluno dos elementos constituintes do grupo. Neste notebook devem colocar todo o código bem como as respostas / análises.
2. Prepare o dataset
 - 2.1. Prepare, para cada sujeito do dataset, um dicionário ou array que contenha, para cada emoção, a lista de ficheiros desse sujeito e emoção. Nota: sugere-se o recurso às funções `listdir()` do módulo `os` e da função `string.split()` do python.
 - 2.2. Elabore uma rotina que carregue os dados relativos a um indivíduo (ou todos) e os devolva num Array NumPy, no formato `[nr_de_videos x`

- nr_de_landmarks*coordenadas x nr_de_frames]. Note que vídeos diferentes têm durações (nr de frames) diferentes, pelo que deve usar o nr de frames do maior vídeo e usar valores NaN para completar os vídeos de duração menor.
- 2.3. Crie também uma variável Emotion com o formato [nr_de_videos x 1] onde guarda, para cada vídeo, um valor representante da emoção representada nesse vídeo (presente no nome do ficheiro CSV).
 - 2.4.  Implemente uma função de visualização das expressões, que, recebendo as coordenadas dos landmarks relativas a um frame, gere uma imagem semelhante ao representado na figura 1.
 - 2.5. Adapte a função anterior para permitir a visualização, em sobreposição, de vários frames (por exemplo, todos os frames de um vídeo).
3. **Análise e tratamento de *Outliers*:** o objetivo será identificar e tratar *outliers* no *dataset* usando diferentes abordagens univariável e multivariável. Para o efeito iremos juntar os frames dos vários vídeos e considerar cada frame como uma amostra, criando um dataset transformado com o formato [nr_total_de_frames x nr_de_landmarks*coordenadas].
 - 3.1. Normalize os valores das coordenadas dos landmarks em cada frame em função do landmark de referência 27, da seguinte forma:







$$x_k = \frac{x_k - x_{ref}}{x_{max}}$$

$$y_k = \frac{y_k - y_{ref}}{y_{max}}$$

Considere x_{max} como sendo 1280 e y_{max} como sendo 720, as dimensões máximas do vídeo, em pixéis, e x_{ref} e y_{ref} como sendo os valores das variáveis x_{27} e y_{27} (coordenadas do landmark de referência 27), nesse mesmo frame.
 - 3.2.  Elabore uma rotina que apresente simultaneamente o *boxplot* de cada emoção (8 emoções – eixo horizontal) relativo a **todos os sujeitos** e a um dos landmarks (coordenada x ou y), por exemplo, variável x_{54} . Sugere-se o uso da biblioteca *matplotlib*.
 - 3.3.  Analise e comente a **densidade de *Outliers*** existentes no *dataset*, isto é, para cada variável em função da emoção. Observe que a densidade é determinada recorrendo

$$d = \frac{n_o}{n_r} \times 100$$

em que n_o é o número de pontos classificados como *outliers* e n_r é o número total de pontos. Esta análise deve ser feita por variável e por emoção, reportando depois o valor médio e desvio padrão das densidades d obtidas entre as diferentes variáveis para cada emoção.

- 3.4.  Escreva uma rotina que receba um *Array* de amostras de uma variável e identifique os *outliers* usando o teste Z-Score para um k variável (parâmetro de entrada).
- 3.5.  Usando o Z-score implementado assinale todos as amostras consideradas *outliers* nas variáveis do dataset. Apresente *plots* em que estes pontos surgem a vermelho, enquanto que os restantes surgem a preto. Use $k=3, 3.5$ e 4 .
- 3.6.  Compare e discuta os resultados obtidos em 3.2 e 3.5.
- 3.7.  Elabore uma rotina que implemente o algoritmo k-means para identificar n (valor de entrada) clusters, recebendo agora ambas as coordenadas de um landmark (x e y).
- 3.8.  Determine os *outliers* no *dataset* transformado usando o k-means. Experimente diferentes números de *clusters* e compare com os resultados obtidos em 3.5. Ilustre graficamente os resultados usando plots 2D.
 - 3.8.1. Bónus: poderá realizar um estudo análogo usando o algoritmo DBSCAN (sugere-se que recorra à biblioteca *sklearn*⁴)
- 3.9.  Implemente uma rotina que injete outliers com uma densidade igual ou superior a $x\%$ nas amostras de variável fornecida. Para o efeito deverá:
 - A calcular a densidade de outliers existente no *Array* fornecido com n_r pontos; observe que a densidade d é obtida por

$$d = \frac{n_o}{n_r} \times 100$$

em que

⁴ <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html#sklearn.cluster.DBSCAN>


$$n_o \equiv \#\{p \notin [\mu - k\sigma, \mu + k\sigma]\}$$

- Se a densidade d for inferior a x , então deverá sortear $(x-d)\%$ dos pontos não *outliers* de forma aleatória e para cada ponto selecionado deverá transformá-lo tal que


$$p = \mu + s \times k \times \sigma + q$$

em que μ e σ representam, respectivamente, os valores da média e o desvio padrão da amostra, k é o limite especificado no ponto 3.3, $s \in \{-1, 1\}$ é uma variável escolhida de forma aleatória usando uma distribuição uniforme e q é uma variável aleatória uniforme no intervalo $q \in [0, z]$ em que z é a amplitude máxima do *outlier* relativamente a $\mu \pm k\sigma$.

Nota: a alteração de valores da amostra mudam os parâmetros estatísticos da mesma e, como tal, a densidade de outliers vai variando a cada perturbação. Deve repetir o processo acima descrito de forma iterativa até atingir a percentagem desejada.

- 3.10.  Elabore uma rotina que determine o modelo linear de ordem p . Para o efeito, a sua rotina deverá receber n amostras de treino de um vetor de dimensão p , isto é, $(x_{i,1}, x_{i,2}, x_{i,2}, \dots, x_{i,p})$ e a respetiva saída y_i . A sua rotina deverá determinar o melhor vector de pesos β tal que

$$\underset{\beta}{\operatorname{argmin}} \sum_{i=1}^p \left(y_i - \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots + \beta_p x_{i,p} \right)^2 = \underset{\beta}{\operatorname{argmin}} \|Y - X\beta\|^2$$

- 3.11.  Determine o modelo linear para o a coordenada y do landmark 57 considerando uma janela com p valores anteriores. Usando a rotina desenvolvida no ponto 3.9, injete 5% de *outliers* nessa variável. Elimine os *outliers* injetados e substitua-os pelos valores previstos pelo modelo linear. Analise o erro de predição apresentando i) a distribuição do erro e ii) exemplos de plots contendo o valor previsto e real. Determine o melhor p para o seu modelo.

- 3.12. ✎ Repita 3.11 usando uma janela de dimensão p centrada no instante a prever. Deverá usar não só os $p/2$ valores anteriores e seguintes da variável que pretende prever (y_{57}) mas também dos seus landmarks vizinhos (y_{56} e y_{58}) das variáveis disponíveis (módulos disponíveis). Compare com os resultados obtidos em 3.10.

4. **Extração de informação característica:** o objectivo será comprimir o espaço do problema, extraindo informação característica discriminante que permita implementar soluções eficazes do problema de classificação. Deixaremos de considerar os frames de forma independente, mas sim a sequência de frames que constituem cada vídeo. Assim, o objetivo é terminar com uma tabela de features em que cada linha representa um vídeo e cada coluna uma feature. A figura 3 representa o segmento do pipeline correspondente.

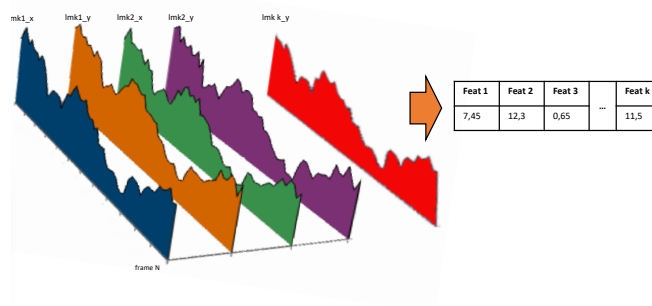


Figura 3 - Conversão dos timecourses das posições dos landmarks ao longo do vídeo para features.

- 4.1. ✎ Extraia features que pense que possam ser relevantes para a identificação das expressões no vídeo, justificando o seu raciocínio. Nota: a visualização dos vídeos fornecidos com o dataset, bem como a visualização dos vários frames sobrepostos implementada na alínea 2.5, podem ajudar na escolha.
- 4.1.1. Inclua obrigatoriamente a posição média da coordenada x e y de 10 landmarks de interesse, escolhidos por si.
 - 4.1.2. Sugestão 1: defina ângulos, áreas, ou distâncias entre landmarks de interesse frame a frame, extraindo depois métricas estatísticas desses valores entre todos os frames do vídeo (média, desvio padrão, etc).
 - 4.1.3. Sugestão 2: calcule variações entre frames de landmarks de interesse (passando de medidas de posição para velocidade),

calculando depois estatísticas desses valores entre todos os frames do vídeo (média, desvio padrão, etc).

- 4.2. ✎ Usando as features extraídas, determine a significância estatísticas da diferença dos seus valores médios entre as diferentes expressões. Observe que poderá aferir a gaussianidade da distribuição usando, por exemplo, o teste Kolmogorov-Smirnov (vide documentação do SciPy). Para rever a escolha de testes estatísticos sugere-se a referência⁵: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2881615/> . Comente.
- 4.3. ✎ Desenvolva o código necessário para implementar o PCA de um feature set.
- 4.4. ✎ Determine a importância de cada vetor na explicação da variabilidade do espaço de features. Note que deverá normalizar as features usando o z-score. Quantas variáveis deverá usar para explicar 75% do feature set?
 - 4.4.1. Indique como poderia obter as features relativas a esta compressão e exemplifique para um vídeo à sua escolha.
 - 4.4.2. Indique as vantagens e as limitações desta abordagem.
- 4.5. ✎ Desenvolva o código necessário para implementar o Fisher Score e o ReliefF.
- 4.6. ✎ Identifique as 10 melhores features de acordo com o Fisher Score e o ReliefF.
 - 4.6.1. Indique como poderia obter as features relativas a esta compressão e exemplifique para um instante à sua escolha.
 - 4.6.2. Indique as vantagens e as limitações desta abordagem.

⁵ Jean-Baptist du Prel, Dr. med.,¹ Bernd Röhrig, Dr. rer. nat.,² Gerhard Hommel, Prof. Dr. rer. nat.,³
³ Jean-Baptist du Prel, Bernd Röhrig, Gerhard Hommel, and Maria Blettner, Choosing Statistical Tests, Deutsches Arzteblatt, v107(19), 2010