UNIVERSIDADE FEDERAL DO ESTADO DO RIO GRANDE DO NORTE

Disciplina / Área de conhecimento:

ECT2702 - TÓPICOS AVANÇADOS EM INFORMÁTICA I - T01 (2019.2)

SVM - Support Vector Machine

Luís Fernando Tavares

Natal, 2019

**Sumário**

1 Introdução…………………………………………...…………...…………..............................………......03

## 2 Metodologia.……………...……………………….......................................................….……......04

## 3 Códigos……………………....................................................................................………......05

## 4 Experimentos………………..........................................................…....................………......08

02

**1 Introdução**

O problema abordado neste relatório é: uma base de dados que contem registro de coordenadas do plano xy, contendo a posição de partes do corpo de uma pessoa, o objetivo é saber em que posição o corpo está posicionado (ex.: sentado e levantado).

A base de dados possui diversos dados coletados a partir de um programa que estipula a pose de uma pessoa, tais como:

* nose\_x, nose\_y: Posição do nariz.
* leftEye\_x, leftEye\_y: Posição do olho esquerdo.
* rightEye\_x ,rightEye\_y: Posição do olho direito.
* leftEar\_x, leftEar\_y: Posição da orelha esquerda.
* rightEar\_x, rightEar\_y: Posição da orelha direita.
* leftShoulder\_x, leftShoulder\_y: Posição do ombro esquerdo.
* rightShoulder\_x, rightShoulder\_y: Posição do ombro direito.
* leftElbow\_x, leftElbow\_y: Posição do cotovelo esquerdo.
* rightElbow\_x, rightElbow\_y: Posição do cotovelo direito.
* leftWrist\_x, leftWrist\_y: Posição do pulso esquerdo.
* rightWrist\_x, rightWrist\_y: Posição do pulso direito.
* leftHip\_x, leftHip\_y: Posição do quadril esquerdo.
* rightHip\_x, rightHip\_y: Posição do quadril direito.
* leftKnee\_x, leftKnee\_y: Posição do joelho esquerdo.
* rightKnee\_x, rightKnee\_y: Posição do joelho direito.
* leftAnkle\_x, leftAnkle\_y: Posição do tornozelo esquerdo.
* rightAnkle\_x, rightAnkle\_y: Posição do tornozelo direito.

03

**2 Metodologia**

O modelo de Machine Learning utilizado para resolver este problema é a Máquina de Vetores de Suporte (SVM - Support-Vector Machines) são modelos de aprendizado supervisionados com algoritmos de aprendizado associados que analisam dados usados ​​para análise de classificação e regressão.

Dado um conjunto de exemplos de treinamento, cada um marcado como pertencente a uma ou a outra de duas categorias, um algoritmo de treinamento SVM constrói um modelo que atribui novos exemplos a uma categoria ou a outra, tornando-o um classificador linear binário não probabilístico.

Um modelo SVM é uma representação dos exemplos como pontos no espaço, mapeados para que os exemplos das categorias separadas sejam divididos por uma lacuna clara o mais ampla possível. Novos exemplos são mapeados no mesmo espaço e prevê-se que pertençam a uma categoria com base no lado da lacuna em que caem.

O objetivo do algoritmo da máquina de vetores de suporte é encontrar um hiperplano em um espaço N-dimensional (N - o número de recursos) que classifica distintamente os pontos de dados.

Para separar as duas classes de pontos de dados, existem muitos hiperplanos possíveis que podem ser escolhidos. O objetivo é encontrar um plano com a margem máxima, ou seja, a distância máxima entre os pontos de dados das duas classes. A maximização da distância da margem fornece algum reforço para que os pontos de dados futuros possam ser classificados com mais confiança.

Além de executar a classificação linear, os SVMs podem executar com eficiência uma classificação não linear usando o que é chamado de truque do kernel, mapeando implicitamente suas entradas em espaços de recursos de alta dimensão.

Quando os dados não são rotulados, o aprendizado supervisionado não é possível e é necessária uma abordagem de aprendizado não supervisionado, que tenta encontrar o agrupamento natural dos dados em grupos e, em seguida, mapear novos dados para esses grupos formados.

O algoritmo de cluster de vetores de suporte aplica as estatísticas de vetores de suporte, desenvolvidos no algoritmo de máquinas de vetores de suporte, para categorizar dados não rotulados, e é um dos algoritmos de clustering mais amplamente utilizados na indústria.

Os atributos escolhidos para o SVM são todas as coordenadas possíveis de pontos específicos citada anteriormente, serão todos utilizados para compara-los entre si, no objetivo de detectar se em uma foto a pessoa está sentada ou levantada.

04

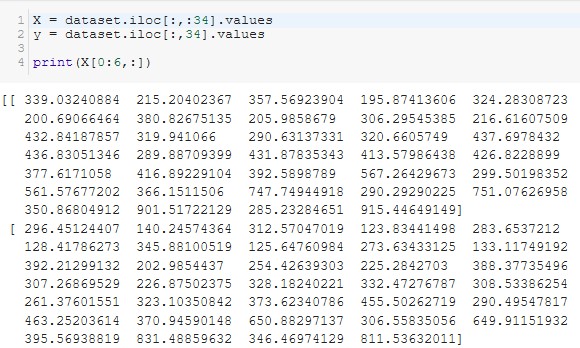
**3 Códigos**

**1º Importação:**



Esse código faz a importação dos dados da base de dados, no caso em .csv, denominada de Pose Estimation. O dataset.head mostra uma certa quantidade de dados contidos em cada linha da base.

**2º Seleção dos atributos:**



Esse código permite a escolha dos atributos (X) que são usados para comparação do parâmetro (Y), sendo este as coordenadas de certas partes pré-selecionadas para a base de dados, no caso será usada todas elas em comparação ao estado da pessoa, sentada ou levantada. O print mostra uma certa quantidade de dados dos atributos contidos em cada linha.

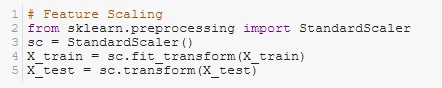
**3º Treinamento e teste:**



Esse código inicia a separação dos atributos para serem treinados e testados. O test\_size é a porcentagem de atributos que irão ser utilizados para o teste (No caso é 35%).

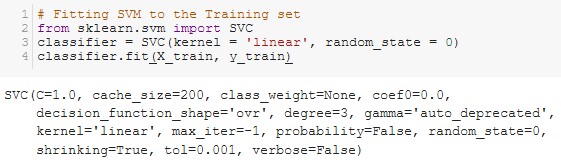
**05**

**4º** Escala de recursos**:**



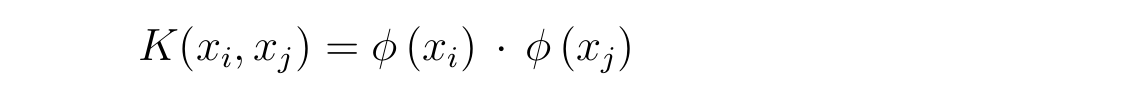
Ele transformará seus dados de forma que sua distribuição tenha um valor médio 0 e desvio padrão de 1. Dada a distribuição dos dados, cada valor no conjunto de dados terá o valor médio da amostra subtraído e depois será dividido pelo desvio padrão de todo o conjunto de dados, dessa forma serão atribuidos na categorização dos dados para saber se estão sentados ou levantados.

5**º Treinamento:**

****

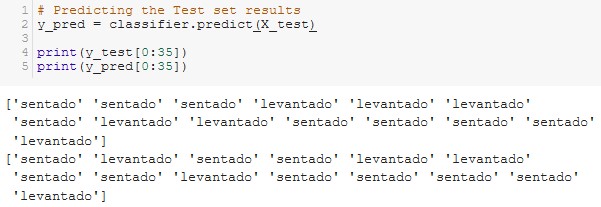
Esse código inicia o treinamento no SVM que permite a estimativa do gradiente, utilizando a função Kernel linear\* que reduz a complexidade de encontrar a função de mapeamento. Portanto, a função Kernel define o produto interno no espaço transformado.

**\***



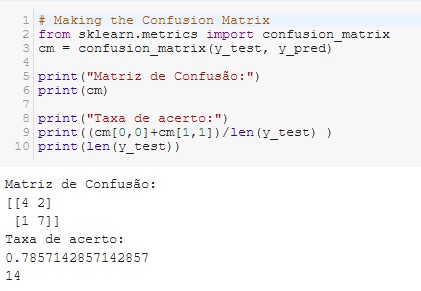
06

6**º Previsões:**

****

Esse código mostra a previsão dos dados feitos pelos testes em comparação com o trainamento.

7**º Taxa de acerto e Matriz de confusão:**

****

Calcula e mostra a taxa de acerto e os valores da matriz de confusão (verdadeiro positivo, falso negativo, falso positivo e verdadeiro negativo).

07

**4 Experimentos**

Os tipos de testes executados foram para, de acordo com os dados fornecidos, ter um gral de certeza que, de acordo com as coordenadas fornecidas em relações as partes do corpo humano que ele está sentado ou levantado , a porcentagem de atributos que irão ser utilizados para o teste é de 35% (test\_size), e para descobrir a estimativa do gradiente, foi utilizado a função Kernel linear e por fim os testes são comparados com o trainamento e é construida a construção da matriz de confusão que mostra quantidades de verdadeiro positivo, falso negativo, falso positivo e verdadeiro negativo, que são respectivamente quanto dos dados bateram, bateram mas não foram considerados, não bateram mas foram considerados e os que não bateram.

Parâmetro avaliado foi o estado de sentado ou levantado, o qual foi comparado com os atributos de todas as coordenadas citadas enteriormente.

Os resultados obtidos foram a matriz de confusão, a qual demonstrava os acertos, erros e falhas obtidas através da comparação entre as taxas dos testes e do que foi previsto, a qual nos permitiu calcular a taxa de acerto, a qual mostrava o quanto o programa era certeiro para calcular os alunos que iriam ser reprovados ou não, no caso possui uma taxa de acerto de 78.5%, ou seja foi possível prever o estado das pessoas (levantado ou sentado) em quase 79 % dos casos, assim esse programa pode ser usado para prever se uma pessoa está sentada ou levantada em uma foto.

08