

# Relatorio do Trabalho 3 de Introdução à Inteligência Artificial

Lucas Mendonça Macedo Amaral, 17/0149668

Lucas Veríssimo Botelho, 17/0109071

10 de novembro de 2019

Universidade de Brasília - Instituto de Ciências Exatas  
Departamento de Ciência da Computação - CIC 116572 - Introdução à Inteligência Artificial  
2019.1 - Turma A - Professor Dibio Leandro Borges  
Prédio CIC/EST - Campus Universitário Darcy Ribeiro  
Asa Norte 70919-970 Brasília, DF  
lucasmamaral@gmail.com  
lucas.vbotelho83@gmail.com

## Resumo

O relatório a seguir consiste em um breve resumo sobre o desenvolvimento do projeto proposto pela disciplina de Introdução à Inteligência Artificial, que fundamenta-se em um algoritmo de Floresta Randômica que tem como objetivo principal, analisar uma base de dados referente a atributos de selfies e seu grau de popularidade, e a partir desse ponto conseguir analisar o grau de importancia de cada atributo analisado para uma selfie "perfeita".

**Palavras-chave:** *Floresta Randômica, Árvores de Decisão*

## 1 Contextualização das partes do Trabalho

### 1.1 Floresta Randômica

O algoritmo de Floresta Randomica consiste em utilizar da tecnica de bagging para gerar varios sub-conjuntos aleatorios de dados apenas utilizando os dados iniciais, assim tambem sao criados juntos com os dados arvores de decisão que são utilizadas pelo algoritmo para chegar a um consenso sobre a resposta final do problema. Com a quantidade grande de arvores de decisão aleatorias se evita o problema de overfitting que é quando se treina um modelo ate um ponto alem do necessario.

### 1.2 Árvore de Decisão

O algortimo de Arvore de Decisão que foi implementado nesse trabalho consiste utilizar um modelo estatistico supervisionado no qual os dados ja estão marcados. Assim o algoritmo consegue dividir o problema em varios sub-problemas que consequentemente sao divididos também recursivamente, ate se marcar os dados em suas subseqüentes regiões.Chegando a conclusão.

## 2 Arquitetura do Sistema Desenvolvido

### 2.1 Visão Geral

O sistema foi desenvolvido em cima de python utilizando os modulos Pandas, Sklearn e IPython. A partir disso foi desenvolvido o programa no arquivos main.py, todo o algoritmo desenvolvido esta nesse arquivo, sendo o mesmo devidamente comentado passo por passo.

## 3 Descrição do Funcionamento do Programa

### 3.1 Grau de Importancia

```
DataFrame:
   features  importance
8        white  0.059039
24    brown hair  0.049666
17  wearing glasses  0.044634
14        smiling  0.044488
1         is female  0.043166
22    black hair  0.040805
12    round face  0.040130
27    straight hair  0.039478
11    oval face  0.039462
5         youth  0.038062
33    wearing hat  0.036208
4        teenager  0.035771
13    heart face  0.035323
15    mouth open  0.033887
26    curly hair  0.033841
21    duck face  0.032903
23    blond hair  0.029415
10        asian  0.029093
19    wearing lipstick  0.024629
31    using mirror  0.024472
30    using earphone  0.024472
29    showing cellphone  0.024350
20        tongue out  0.020687
9         black  0.019968
18    wearing sunglasses  0.018415
25        red hair  0.017710
6    middle age  0.017583
0    partial faces  0.017017
35    dim lighting  0.015269
34    harsh lighting  0.014856
3         child  0.014558
32        braces  0.012290
28    braid hair  0.011887
16    frowning  0.011406
2         baby  0.004153
7        senior  0.000906
Creating an html file with name (importancefeatures.html)
```

Figura 1: Grau de importancia de cada atributo listado

Para executar o algoritmo de floresta randômica com base na base de dados que foi proposta basta ir no terminal e rodar o programa na ultima versão do Python, a partir dai o algoritmo de árvore randômica sera executado, a floresta randômica em questão foi implementada gerando 100 árvores de decisão, sendo analisado todas os atributos do arquivo. Para conseguir fazer o parse de todos o atributos do

arquivo dado, foi utilizado a biblioteca “pandas”, que ajudou demais esse trabalho. Após isso, alguns problemas ocorreram, visto que a classe anteriormente usada não tratava parâmetros com “float point”, então utilizamos a classe “RandomForestRegressor” da biblioteca “Sklearn”. A partir disso, conseguimos treinar a nossa floresta com base nos dados propostos e geramos uma tabela, utilizando a biblioteca “pandas” novamente, para facilitar a visualização do usuário e para mostrar os dados na tela utilizamos a biblioteca “IPython”, depois que os dados foram mostrados, foi criado um arquivo “html”, que gerou a mesma tabela, que dessa vez, pode ser visualizada no browser de melhor escolha do usuário.

## 3.2 Tabela HTML

|    | features           | importance |
|----|--------------------|------------|
| 8  | white              | 0.059039   |
| 24 | brown_hair         | 0.049666   |
| 17 | wearing_glasses    | 0.044634   |
| 14 | smiling            | 0.044488   |
| 1  | is_female          | 0.043166   |
| 22 | black_hair         | 0.040805   |
| 12 | round_face         | 0.040130   |
| 27 | straight_hair      | 0.039478   |
| 11 | oval_face          | 0.039462   |
| 5  | youth              | 0.038062   |
| 33 | wearing_hat        | 0.036208   |
| 4  | teenager           | 0.035771   |
| 13 | heart_face         | 0.035323   |
| 15 | mouth_open         | 0.033887   |
| 26 | curly_hair         | 0.033841   |
| 21 | duck_face          | 0.032903   |
| 23 | blond_hair         | 0.029415   |
| 10 | asian              | 0.029093   |
| 19 | wearing_lipstick   | 0.024629   |
| 31 | using_mirror       | 0.024472   |
| 30 | using_earphone     | 0.024472   |
| 29 | showing_cellphone  | 0.024350   |
| 20 | tongue_out         | 0.020687   |
| 9  | black              | 0.019968   |
| 18 | wearing_sunglasses | 0.018415   |
| 25 | red_hair           | 0.017710   |
| 6  | middle_age         | 0.017583   |
| 0  | partial_faces      | 0.017017   |
| 35 | dim_lighting       | 0.015269   |
| 34 | harsh_lighting     | 0.014856   |
| 3  | child              | 0.014558   |
| 32 | braces             | 0.012290   |
| 28 | braid_hair         | 0.011887   |
| 16 | frowning           | 0.011406   |
| 2  | baby               | 0.004153   |
| 7  | senior             | 0.000906   |

Figura 2: Tabela gerada em HTML aberta em um browser

## 4 Conclusões

Em suma, o projeto, serviu para a familiarização com algumas bibliotecas do python para a solução sobre questões de análise de dados, assim como colocar em prática teorias aprendidas em sala de aula. Com as técnicas aprendidas sobre os algoritmos de floresta randômica e árvores de decisão foi possível utilizar os dados fornecidos pelo artigo "How to Take a Good Selfie?" para que fosse possível classificar as características que fornecem a melhor selfie. A partir do algoritmo que escrevemos conseguimos listar os dados que melhor classificam as selfies de uma forma fácil de se ver e interpretar.