

Universidade Federal do ABC
Programa de Iniciação Científica - UFABC

Um Estudo de Aprendizado de Máquina para Classificar Imagens de Redes Sociais e Extrair Informações que Identifiquem Engajamento

Relatório final - Iniciação Científica - Modalidade bolsista

Edital N° 02/2017

Aluno: Erick Hotta Orsi
RA: 11201721564
Bacharelado em Ciência & Tecnologia
erickorsig@gmail.com

Orientador: Alexandre Noma
Centro de Matemática, Computação e Cognição
alexandre.noma@ufabc.edu.br

Santo André, 31 de Agosto de 2018

CARTA DE AVALIAÇÃO

Ao(a) Ilustríssimo(a) Pró-Reitor(a) de Pesquisa,

Encaminho o relatório final do aluno Erick Hotta Orsi referente ao projeto de pesquisa junto ao programa de Iniciação Científica, edital N° 02/2017.

Durante todo o período de trabalho, o aluno: (1) apresentou uma ótima autonomia acadêmica, e (2) teve um excelente desempenho no desenvolvimento das atividades consideradas no Projeto de Iniciação Científica.

Atenciosamente,

Alexandre Noma (orientador)
Professor Adjunto - CMCC

Resumo

Muitas instituições utilizam cada vez mais as redes sociais para se aproximar de pessoas. Elas postam mensagens para despertar interesse e engajamento. O engajamento considera a quantidade de curtidas, comentários, visualizações e compartilhamentos. Para as instituições, é fundamental conhecer quais postagens geram maior engajamento. Isto permite direcionar futuras postagens, para comunicar de maneira assertiva, fornecendo postagens mais adequadas ao interesse das pessoas. Para medir o engajamento, é necessário extrair dados de redes sociais e identificar quais características são interessantes para classificação. O objetivo deste projeto de pesquisa é preparar o aluno para estudar conceitos de aprendizado de máquina (*machine learning*). Especificamente, este projeto inicia com uma extração de informação de imagens postadas por uma instituição em uma rede social, como o *Facebook*. No caso de uma imagem, é interessante saber o seu conteúdo para medir o engajamento. Por exemplo, a imagem pode ter uma paisagem, pessoas e/ou objetos específicos. Além disso, poderíamos ter vários tipos de pessoas: homem, mulher, criança, idoso. Similarmente, vários tipos de paisagens e objetos. Após o aluno pesquisar ferramentas para extrair imagens de redes sociais e classificar o conteúdo delas, ele construiu um conjunto rotulado de dados para um estudo com classificação supervisionada. A ideia foi permitir, por exemplo, prever a quantidade de curtidas ou compartilhamentos teríamos para uma nova imagem de entrada, baseado nos dados históricos composto pelo conjunto de dados rotulados. Desta forma, o aluno teve oportunidade de experimentar conceitos importantes em ciência da computação. Isto possibilitou uma base sólida para o aluno, incentivando-o a continuar seus trabalhos com pesquisa científica.

Palavras-chave: aprendizado de máquina, redes sociais, classificação supervisionada de imagens, engajamento.

Sumário

1	Introdução	5
2	Resumo	5
3	Metas e Objetivos	6
4	Metodologia	8
5	Resultados	11
6	Cronograma	13
7	Discussão	14
	Referências	16

1 Introdução

Atualmente, as redes sociais são importantes canais de comunicação, podem influenciar a vida de muitas pessoas no mundo inteiro, e serem usadas como uma forma de conexão pessoal, como um recurso empresarial de *marketing* e de atendimento ao consumidor [18, 16]. Uma das importâncias deste meio de comunicação é possibilitar tomada de decisões a partir das reações e respostas do público.

As interações do público com uma entidade podem se manifestar através da quantidade de curtidas, comentários, compartilhamentos, etc. Neste trabalho, chamaremos este tipo de interação como **engajamento**.

Este projeto tem a finalidade de usar algoritmos para facilitar uma transformação de dados em conhecimento. Idealmente, deseja-se automatizar o processo desde a extração de informações de redes sociais até a classificação dos dados em plataformas de aprendizado de máquina [15] para medir o engajamento do público com a entidade.

Neste trabalho, o foco foi sobre o *Facebook*, uma rede social bastante popular, com ferramentas computacionais disponíveis para facilitar a extração dos dados via programação usando-se bibliotecas (por exemplo, [7]). Especificamente, neste projeto de PDPD, foi desenvolvido um programa para extrair as imagens de uma rede social (por exemplo, o perfil [12]), e posteriormente, este foi modificado para extrair também o conteúdo destas imagens em forma de *tags* (por exemplo, usando a biblioteca [2]).

Após extrair os dados das imagens (*tags*, quantidade de curtidas, etc), construiu-se um conjunto rotulado de amostras para treinamento e obtenção de um classificador. Desta forma, a ideia foi classificar uma nova imagem (dada como entrada ao classificador) entre duas classes: "muitas" ou "poucas" curtidas (usando a biblioteca [3]).

2 Resumo

Na primeira etapa, o foco do projeto foi uma extração automática de imagens do *Facebook*, assim como os elementos visuais de cada imagem, usando-se duas bibliotecas de programação: Facebook Graph API [7] e Computer Vision API do Microsoft Azure [2].

Primeiramente, imagens foram extraídas da página com perfil público da AACD [12] por um programa criado em linguagem de programação Python, versão 3. Outras informações referentes a cada imagem, como quantidade de curtidas e data de publicação, entre outras, também foram extraídas e serão ilustradas mais adiante.

Posteriormente, foi implementado no mesmo código em Python, uma extração de elementos para descrever o conteúdo de cada imagem. Uma maneira simples de descrever os componentes visuais de uma imagem é usando-se *tags* (Figura 1).

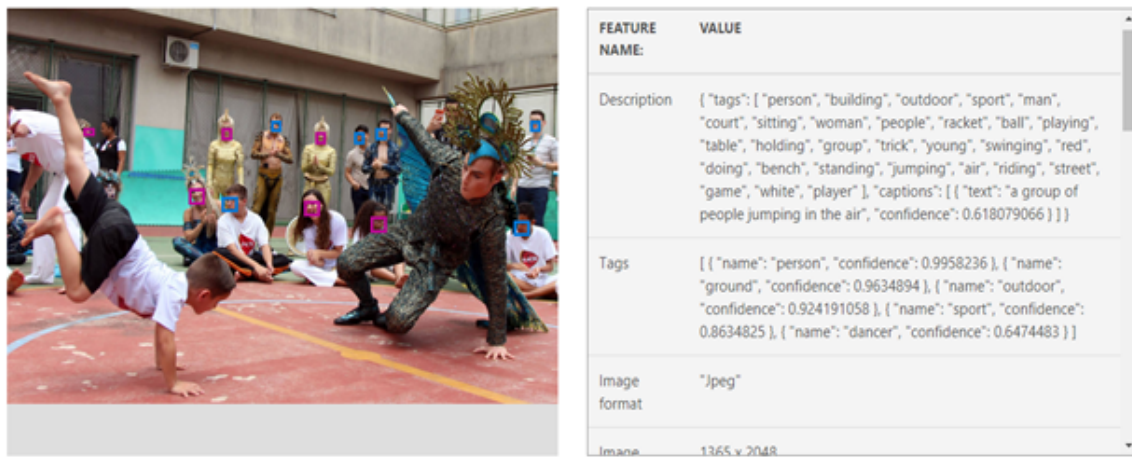


Figura 1: Extraída da página da Computer Vision API do Microsoft Azure [2].

A Figura 1 ilustra algumas *tags* que foram extraídas automaticamente para descrever o conteúdo da imagem: pessoa, prédio, quadra, grupo, dança, movimento, etc. No exemplo, na aba *Tags*, é apresentada a confiança de classificação de cada elemento e *person* tem confiança de mais de 99%. Neste caso, podemos afirmar que a Figura 1 contém (pelo menos) uma pessoa, com uma grande margem de segurança.

Na segunda etapa, o foco foi estruturar os dados extraídos. Posteriormente, efetuou-se uma análise exploratória dos dados [17]. Em seguida, foi necessário aprender diferentes algoritmos de classificação, usando recursos da ferramenta *scikit-learn* [3]. Desta forma, foi possível identificar quais *tags* são mais "promissoras" para classificação: quais poderiam discriminar melhor durante a classificação? Basicamente, escolhemos aquelas com boxplots com aparências "diferentes", como ilustrado na Figura 2.

3 Metas e Objetivos

Muitas instituições utilizam as redes sociais como um importante canal para se aproximar do seu público alvo. Um exemplo de uma instituição beneficente é a página com perfil público da AACD [12]. Por este canal de comunicação, é possível interagir com a entidade, compartilhar notícias, divulgar eventos, fazer comentários, etc.

Para ampliar sua influência sobre o público, é interessante para a entidade medir o grau de interação ou engajamento das pessoas, seja pelas quantidades de curtidas, comentários, compartilhamentos, etc.

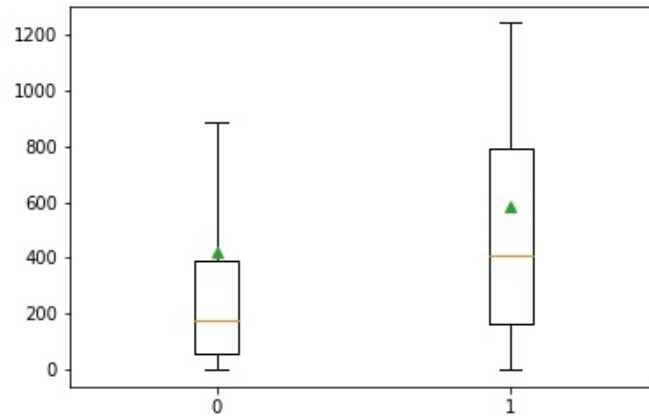


Figura 2: Um exemplo de boxplot para a tag "shirt".

Em *marketing*, isto permite identificar quais postagens são mais adequadas ao seu público alvo para que nas postagens futuras a entidade invista nos tipos de postagens que podem gerar maior repercussão.

Um dos objetivos deste projeto foi criar ferramentas para identificar quais tipos de postagens que geram mais engajamento. Para isso, foram propostas as seguintes tarefas:

1. Extração de imagens de redes sociais.

- Neste primeiro passo, o objetivo foi pesquisar ferramentas de programação para efetuar uma extração automática das imagens disponíveis em um perfil de uma rede social.

2. Classificação do conteúdo das imagens para extração de *tags*.

- Neste passo, o objetivo foi pesquisar ferramentas de programação para classificar o conteúdo das imagens extraídas no passo anterior.

3. Construção de um conjunto de dados rotulados para testar diferentes algoritmos de classificação supervisionada.

- Neste último passo, o objetivo foi criar um conjunto de amostras rotuladas, cada amostra com um vetor de características e um vetor de respostas para identificar engajamento. Desta forma, foi possível testar diferentes algoritmos de classificação supervisionada.

Os dois primeiros passos do projeto foram realizados na primeira etapa do projeto. O terceiro passo foi executado na segunda etapa. Cada um dos passos é detalhado a seguir.

4 Metodologia

No primeiro passo do projeto, o aluno pesquisou ferramentas de programação para extrair, de maneira automática, dados disponíveis de um perfil público da rede social *Facebook*. Dentre as ferramentas encontradas, as que se destacaram foram: Buzzmonitor [1], Scup [14], Seeker [6], Live Buzz [11] e Facebook Graph API [7].

O Facebook Graph API foi a biblioteca escolhida para extrair as imagens do perfil público da AACD [12]. Esta biblioteca é gratuita, funciona com a linguagem de programação Python, possui uma documentação de fácil acesso, com diversos exemplos (veja [10]).

Durante os experimentos, um programa para extrair imagens foi desenvolvido em Python, versão 3. Para isto, foi necessário instalar duas bibliotecas, disponíveis no GitHub [9]: **facebook** e **urllib3**. Para ilustrar, abaixo encontram-se alguns exemplos de código.

```
import sys
sys.path.append("C:/Users/Erick/Documents/PDPD/Projeto/Extracao de Dados/python/bibs/facebook-sdk-master")
import facebook
import urllib3
import requests
import urllib

token = ''

graph = facebook.GraphAPI(access_token=token, version = '2.10')
page = graph.request('ajudeaacd?fields=albums,description')

albumsList = page['albums']
albumsDataList = albumsList['data']
timelineID = albumsDataList[0]['id']

timelinePhotos = graph.get_object(id=timelineID,fields='count,photos')

count = timelinePhotos['count']
photos = timelinePhotos['photos']
photosList = photos['data']
photoID = photosList[0]['id']

photoLinks = graph.get_object(id=photoID,fields='picture')

picture = photoLinks['picture']

print (picture)
```

Figura 3: Exemplo para extrair uma imagem com a biblioteca Facebook Graph API [7].

Na Figura 3, as primeiras linhas do código efetuam a importação de algumas bibliotecas. A mais importante delas é a biblioteca **facebook**, que corresponde biblioteca [7]. Esta biblioteca permite acessar os dados de um perfil através de um programa de computador. Para isto, o primeiro passo é ter uma conta no *Facebook* e criar um *token* de acesso. No exemplo acima, o *token* está vazio e deve ser substituído por um *token* válido.

Com este *token*, o programa pode acessar os dados do perfil por um período limitado. Desta forma, é necessário criar um novo *token* após a expiração do tempo de conexão. Um exemplo útil que foi usado

para criar este programa encontra-se em [10].

No segundo passo do projeto, o aluno pesquisou ferramentas para classificar o conteúdo das imagens. As ferramentas que se destacaram foram: IBM Watson [13], Google Cloud ML [5] e Microsoft Azure [2].

A biblioteca escolhida foi a Computer Vision API da Microsoft Azure. Esta biblioteca não é gratuita, mas tem um período de demonstração gratuito, funciona com a linguagem de programação Python e possui uma documentação de fácil acesso com diversos exemplos (veja [8]).

```
import http.client
import urllib.request
import urllib.parse
import urllib.error
import base64
import json
subscription_key = ''
uri_base = 'westcentralus.api.cognitive.microsoft.com'

headers = {
    'Content-Type': 'application/json',
    'Ocp-Apim-Subscription-Key': subscription_key,
}

params = urllib.parse.urlencode({
    'visualFeatures': 'Faces,Categories,Description,Color',
    'language': 'en',
})

url = ""
body = '{"url":"' + url + '"}'

conn = http.client.HTTPSConnection('westcentralus.api.cognitive.microsoft.com')
conn.request("POST", "/vision/v1.0/analyze?%s" % params, body, headers)
response = conn.getresponse()
data = response.read()
parsed = json.loads(data)
print("Response:")
result = json.dumps(parsed, sort_keys=True, indent=2)
print(result)
conn.close()
```

Figura 4: Exemplo para extrair tags de uma imagem com a biblioteca Computer Vision API da Microsoft Azure [2].

A Figura 4 ilustra um exemplo de código para extrair *tags* de uma imagem, especificada por uma *url*. Note que para usar a biblioteca [2], não é necessário importar uma biblioteca específica, pois as chamadas para a API são feitas via **http**.

Para ter acesso a esta biblioteca, é necessário se cadastrar para obter uma licença (*subscription key*). Note que é possível obter uma licença temporária gratuita. No exemplo acima, a licença está vazia e deve ser substituída por uma que seja válida. Um exemplo útil que foi usado para criar este

programa encontra-se em [8].

No terceiro passo do projeto, foi necessário estruturar os dados para facilitar futuras análises. Para isso, foram criados dicionários no Python que continham os dados organizados de diferentes formas (por imagem, por *tag*), o que facilitou o acesso às informações.

Em seguida, o aluno realizou uma análise exploratória dos dados [17], na qual foi possível visualizar as informações que seriam úteis para a classificação (quantidade de aparências de cada *tag*, quartis, percentis, etc.), extraídas por meio de boxplots, como ilustrado na Figura 3.

Para realizar uma classificação multivariada para encontrar uma variável resposta, foram utilizadas ferramentas do *scikit*, especificamente o *scikit-learn* [3] do Python, pois estas apresentam diversos algoritmos de classificação. Foram testados diferentes algoritmos de classificação: Bayes Ingênuo, Árvore de Classificação, Support Vector Classification, Nearest Neighbors Classification. Os códigos não foram inseridos neste documento, para evitar ultrapassar a limitação máxima de 20 páginas, mas um exemplo bem simples usando o classificador SVM é dado em [4].

O vetor de características foi composta pelas *tags*, representadas numericamente por 0 ou 1 em cada imagem, sendo 0 a ausência da *tag* e 1 a presença da *tag* na imagem. Por exemplo, na Figura 5, a imagem 1 tem apenas a *tag* "text". Já a imagem 2 contém várias *tags*: "spectacles", "goggles", "sunglasses", etc.

	text	spectacles	goggles	sunglasses	wearing	holding	screenshot	abstract	man	sign	monitor	woman	screen	phone	standing	food	light	shirt	mirror	map	refrigerator	photo	girl	red	device	building	person
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	
14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
15	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
17	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1
18	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	1
19	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1
20	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
21	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1
22	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
23	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
24	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0
25	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
28	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	1	1	0	0
29	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0

Figura 5: Captura de tela dos dados estruturados exibidos em uma planilha.

A variável resposta foi a quantidade de curtidas binarizada. Dois grupos foram definidos de acordo com a mediana das curtidas de todas as imagens, sendo 0 a resposta para "quantidade de curtidas abaixo da mediana" e 1 para "quantidade de curtidas acima ou igual à mediana".

5 Resultados

No primeiro passo do projeto, o programa extraiu 1439 imagens disponíveis no perfil @ajudeaacd, disponíveis na época da escrita deste relatório, com histórico de postagens de 06/12/2010 a 10/01/2018. Outras informações também foram extraídas, por exemplo, data e hora de publicação, quantidades de curtidas e de comentários de cada imagem. Estas informações foram salvas nos próprios nomes dos arquivos (veja Figura 6).

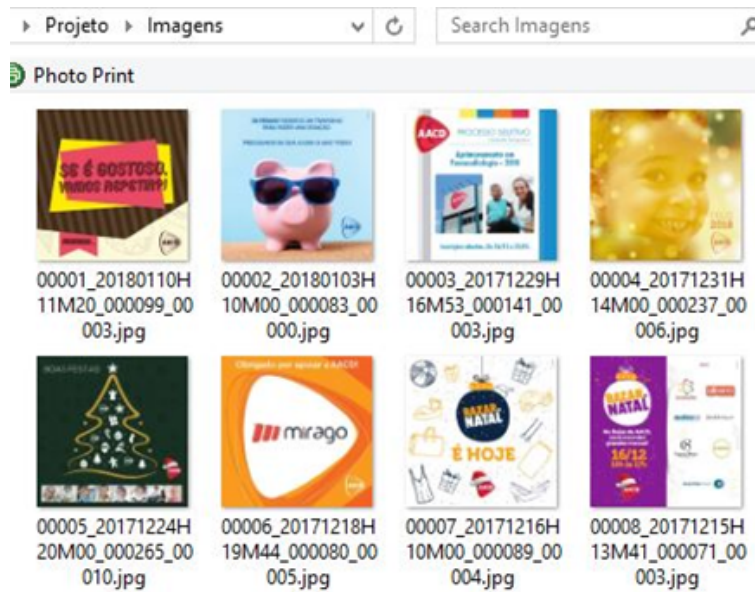


Figura 6: Arquivos das imagens extraídas, nomeados de acordo com o índice, data e hora de publicação, número de curtidas e número de comentários de cada imagem.

No segundo passo do projeto, para cada imagem, o programa extraiu as *tags* que descrevem as informações e os componentes visuais da imagem. Essas informações foram salvas com o mesmo nome do arquivo da imagem (com a extensão `txt`) para facilitar a organização dos dados para o próximo passo do projeto (Figura 7).

Na Figura 8, temos um exemplo do conteúdo de um arquivo `txt` com as *tags* obtidas para descrever o conteúdo de uma dada imagem. Neste ponto, não nos preocupamos com o formato dos dados. Note que o formato atual permite que um programa em Python construa um dicionário para facilitar a estruturação dos dados no próximo passo do projeto.

Após a estruturação dos dados, foi possível fazer uma análise exploratória [17]. A Figura 3 mostra um exemplo com dois boxplots, um para a quantidade de vezes que a *tag* aparece (1) e outro para a quantidade de vezes que a *tag* não aparece (0). Por meio dos boxplots, foi possível identificar informações importantes, como quartis, percentis e média. A ideia foi examinar a distribuição dos dados e escolher as características discriminantes para a classificação.

Projeto Tags		Search Tags
Photo Print		
Name	Date modified	
00001_20180110H11M20_000099_00003.txt	16/01/2018 6:03 p....	
00002_20180103H10M00_000083_00000.txt	16/01/2018 6:04 p....	
00003_20171229H16M53_000141_00003.txt	16/01/2018 6:04 p....	
00004_20171231H14M00_000237_00006.txt	16/01/2018 6:04 p....	
00005_20171224H20M00_000265_00010.txt	16/01/2018 6:04 p....	
00006_20171218H19M44_000080_00005.txt	16/01/2018 6:04 p....	
00007_20171216H10M00_000089_00004.txt	16/01/2018 6:04 p....	
00008_20171215H13M41_000071_00003.txt	16/01/2018 6:04 p....	

Figura 7: Arquivos das tags, um arquivo *txt* para cada arquivo de imagem.

```

"description": {
  "captions": [
    {
      "confidence": 0.8800966122564563,
      "text": "a group of people posing for a picture"
    }
  ],
  "tags": [
    "person",
    "sport",
    "building",
    "group",
    "man",
    "sitting",
    "people",
    "ball",
    "young",
    "posing",
    "player",
    "bunch",
    "..."
  ]
}

```

Figura 8: Exemplo de conteúdo de um arquivo *txt* com as tags.

Em geral, as classificações apresentaram resultados preliminares interessantes, mas longe do ideal. De acordo com nossos experimentos preliminares, o acerto foi em torno de 65%. O melhor deles foi o SVM, que atingiu 70% de acerto.

Este resultado pode ser devido à ausência de contexto entre as *tags*. As *tags* foram consideradas independentes entre si, o que não ocorre na realidade, pois o conteúdo semântico de uma imagem depende fundamentalmente da presença de combinações específicas de diferentes componentes ou *tags*.

Em resumo, as classificações apresentaram resultados que reforçam que a presunção de independência entre as *tags* gera uma perda de informação de contexto da imagem, empobrecendo a performance de classificação. Ou seja, a presença de uma *tag* sozinha pode ser insuficiente para gerar uma grande quantidade de curtidas, mas uma combinação delas pode motivar e sensibilizar mais uma pessoa devido ao contexto ou à situação ilustrada pela imagem.

6 Cronograma

Atividade	01	02	03	04	05	06	07	08	09	10
Estudo bibliográfico e de ferramentas	X	X	X	X	X	X	X	X		
Extração de dados de redes sociais	X	X	X	X						
Classificação de imagens para obter <i>tags</i>		X	X	X	X					
Experimentos com aprendizado de máquina						X	X	X	X	
Análise de resultados e escrita				X	X	X		X	X	X

Figura 9: Cronograma extraído do projeto submetido para obtenção da bolsa PDPD.

A Figura 9 exibe o cronograma do projeto, que resume as atividades propostas e realizadas ao longo dos dez meses de duração da bolsa. O projeto seguiu o cronograma original. As tarefas de extração de dados de redes sociais e de classificação de imagens para obter as *tags* já foram completadas. Desta forma, todas as atividades propostas para serem executadas até o mês 5, destacadas na Figura 9, foram completadas na primeira etapa do projeto.

As tarefas a partir do mês 6 foram executadas na segunda etapa. Na segunda etapa, o foco foi estruturar os dados da extração das imagens e realizar uma análise exploratória dos dados, segundo [17]. Finalmente, experimentou-se diferentes algoritmos de classificação, usando recursos da ferramenta `scikit-learn` [3].

7 Discussão

Para extrair as imagens e as *tags*, algumas dificuldades foram enfrentadas e superadas ao longo do projeto. Uma dificuldade enfrentada no primeiro passo do projeto foi com relação à quantidade limite de imagens que a biblioteca [7] possibilitava extrair por vez. Pelos experimentos, a configuração padrão da biblioteca permitia acessar somente as 100 imagens mais recentes. Ao longo dos estudos, o aluno descobriu uma maneira de extrair as 500 imagens mais recentes. Posteriormente, descobriu que era necessário “virar a página” para acessar o próximo grupo de imagens. Desta forma, foi possível obter a totalidade das imagens disponíveis no perfil do *Facebook*.

Outra dificuldade enfrentada nos dois passos do projeto foi com relação ao limite de tempo no uso das bibliotecas. Pelos experimentos, a biblioteca Facebook Graph API [7] requer um *token* de acesso que permite até 2 horas de funcionamento do programa. Mas para a extração de *tags*, a biblioteca Computer Vision API da Microsoft Azure [2] permite classificar apenas 20 imagens por minuto.

Note que, a princípio, poderíamos dividir a extração dos dados em dois programas diferentes, um para extrair as imagens e outro para extrair as *tags*. Porém, até o momento, não conseguimos fazer isto pois a biblioteca [2] funciona através de chamadas aos métodos via `http`, e os parâmetros de entrada podem ser especificados na forma de `urls`. No nosso caso, para extrair as *tags*, precisamos passar as imagens como parâmetros via `urls`.

Neste projeto, optamos extrair as imagens e as *tags* em um único programa. Em particular, a extração das *tags* é feita imediatamente após a extração de cada imagem, usando o mesmo `url` para a localização da imagem nos dois primeiros passos do projeto. Desta forma, foi necessário criar um atraso adequado para o programa extrair apenas 20 imagens por minuto, mas não longo o suficiente para ultrapassar 2 horas. Este problema foi resolvido adotando-se um atraso de 2 segundos por imagem. O resultado parcial do projeto foi obter 1439 imagens, gravadas em arquivos `jpg`, e 1439 arquivos `txt` com as *tags* correspondentes às imagens.

Note que um desafio do próximo passo foi estruturar os dados coletados para montar um conjunto rotulado, com um vetor de características e um vetor de respostas para cada amostra. Na próxima fase do projeto, foi importante responder as seguintes perguntas:

1. 1) Quais indicadores seriam interessantes para identificar um engajamento?

- Neste trabalho, foram usadas as quantidades de curtidas de cada imagem, classificadas como “muitas” ou “poucas” curtidas.

2. 2) Quais características seriam interessantes para resolver o problema de classificação em questão?

- Foram usadas variáveis indicando a presença ou a ausência de *tags* em cada imagem para classificar a quantidade de curtidas.

3. 3) Quais seriam as possíveis respostas para o problema de classificação?

- Além da quantidade de curtidas, poderia-se explorar a quantidade de comentários, os sentimentos expressos nesses comentários, os componentes de texto da postagem, etc.

Apesar do foco ser em classificação de imagens, a princípio seria possível ampliar o projeto para analisar sentimentos expressos nos textos das postagens, além do conteúdo de vídeos, criando-se assim novas possibilidades de extensão para futuros trabalhos de iniciação científica. Desta forma, a falta de contexto e a falta de precisão na extração automática das *tags* poderia ser contornada analisando-se mais informações presentes nas postagens para melhorar a performance dos classificadores de indicadores de engajamento.

Referências

- [1] <https://app.buzzmonitor.com.br/>.
- [2] <https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/>.
- [3] <http://scikit-learn.org/stable/index.html/>.
- [4] <http://scikit-learn.org/stable/modules/svm.html#classification>.
- [5] <https://cloud.google.com/products/machine-learning/>.
- [6] <https://codecanyon.net/item/facebook-seeker-data-collector/18727854>.
- [7] <https://developers.facebook.com/docs/graph-api/>.
- [8] <https://docs.microsoft.com/pt-br/azure/cognitive-services/computer-vision/quickstarts/python>.
- [9] <https://github.com/explore>.
- [10] <https://towardsdatascience.com/how-to-use-facebook-graph-api-and-extract-data-using-python-1839e19d6999>.
- [11] <https://www.dinamize.com.br/>.
- [12] <https://www.facebook.com/ajudeaacd/>.
- [13] <https://www.ibm.com/watson/services/visual-recognition/>.
- [14] <https://www.scup.com/pt/>.
- [15] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2014.
- [16] A. Barabási. *Linked: How Everything Is Connected To Everything Else And What It Means For Business, Science, And Everyday Life*. Plume, 2003.
- [17] P.A. Morettin and W. de Oliveira Bussab. *Estatística básica*. Saraiva, 2012.
- [18] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.