Esforços do processo de *data wrangling*

O esforços descritos neste documento consistem em observações do processo de *data wrangling* do arquivo de Tweets do perfil @weratedogs no Twitter e uma base de predição das raças dos cães por meio das fotos enviadas por usuários da rede. O processo de *data wrangling* pode ser dividido em três etapas, coleta, avaliação e limpeza.

A tarefa envolveu três conjuntos de dados, sendo eles:

- **Arquivo de Tweets do perfil @weratedogs:** Um conjunto do Arquivo de Tweets do perfil @weratedogs
- **Predição de imagens:** A predição de imagens consiste em um conjunto de dados com o resultado da análise e predição da raça dos cães das fotos anexadas aos tweets enviados ao perfil.
- **Twitter API:** Como o Arquivo de Tweets não apresentava todas as informações necessárias para o desenvolvimento do projeto foram efetuadas requisições à API do Twitter com o objetivo de resgatar estes dados ausentes.

COLETA

O processo de coleta dos dados se deu por download para os conjuntos de dados Arquivo de Tweets e Predição de imagens, sendo o segundo acessado programaticamente.
A coleta das informações ausentes no Arquivo de Tweets foram requisitadas à API do Twitter de forma automatizada consultando a lista de IDs dos Tweets no Arquivo de Tweets.
Para essa tarefa foi utilizada a biblioteca Tweepy e foi necessário criar um App na plataforma do micro blog.

AVALIAÇÃO

Arquivo de Tweets

Acerca da qualidade de dados, o dataframe do Arquivo de Tweets estava incompleto, pois apresentou 2356 registros dos 5000 anunciados. Em sua estrutura, duas colunas não apresentavam valor a unidade de observação, sendo na coluna ‘source’ URLs com referência ao download do App do Twitter nas plataformas móveis, e na coluna ‘expanded_url’ a URL para o Tweet na plataforma. Registros nas colunas ‘nome’ e também na classificação do cão, com valor literal ‘None’, o que deveria ser representado por vazio (np.nan).

Sobre a estrutura dos dados do Arquivo de Tweets, a classificação do cão está distribuída em quatro colunas, ‘doggo’, ‘floofer’, ‘pupper’ e ‘puppo’, contrariando a regra de boas práticas que salienta que cada variável deve formar uma coluna. Colunas como ‘timestamp’, ‘retweet_timestamp’, no formato de *string* e em *float64* as colunas ‘in_reply_to_status_id’, ‘in_reply_to_user_id’, ‘retweeted_status_user_id’.

Image prediction

No que tange a qualidade dos dados do dataframe de Predições, foram encontrados registros em duplicidade para o a mesma URL de imagem, além de apresentar variáveis nos nomes das colunas.

Twitter API

Os dados complementares carregados na API do Twitter apresentou Tweets referenciando a mesma imagem e em sua estrutura colunas com nomes diferentes aos dataframes já mencionados, almejando organização e melhor fluidez da análise. Colunas como ‘created_at’e ‘in_reply_to_status_id’ e ‘in_reply_to_user_id’ como ‘string’, quando seus dados representam informações, respectivamente, de data e hora, e id, números inteiros.

LIMPEZA

Arquivo de Tweets

Sobre a completude do Arquivo de Tweets, nada pôde ser feito, haja vista que a API do Twitter restringe o histórico de pesquisa de Tweets e não é possível alcançar o período aferido pelo enunciado do projeto.

Como descrito anteriormente na Avaliação deste *dataframe*, as colunas ‘source’ e ‘expanded_urls’ foram descartadas por não agregar valor a unidade de observação.

Os registros encontrados com o valor literal ‘None’ na coluna ‘name’ foram substituídos por np.nan e assim auxiliar na utilização e identificação de valores não preenchidos, o mesmo para as colunas de classificação do animal. A classificação dos animais estava representada no nome das colunas e visando utilizar apenas uma variável para este objetivo, foi criada a coluna ‘category’ e os valores atribuídos a ela.

O objetivo da análise em questão visa avaliar somente tweets originais, ou seja, que não sejam retweets. Para identificar os tweets provenientes de outros tweets os registros com valores nas colunas ‘retweeted_status_id’ ou ‘retweeted_status_user_id’ podem ser descartados e posteriormente, também, as colunas.

Image prediction

Os registros das predições foram formatados nos padrões de uma variável por coluna. Nesse sentido, foi criada uma coluna para identificar a tentativa de predição (‘prediction_try’) e outras colunas para os valores mesclados nas colunas, ‘prediction’ para o valor da predição, ‘confidence’ para o grau de confiança da predição, e por último, se a predição afirma ser um cão ou não na imagem com a coluna ‘is_a_dog’.

Twitter API

Visando facilitar a manipulação e combinação de registros entre *dataframes*, os nomes das colunas foram padronizados, renomeando as colunas ‘id’ e ‘media_url’ para ‘tweet_id’ e ‘jpg_url’ respectivamente.

Antes de eliminar a duplicidade de tweets referenciando a mesma imagem, foram filtrados e descartados os registros que apontavam para Tweets identificados como retweets no *dataframe* do Arquivo de Tweets. Após esse passo, foram removidas as duplicidades de tweets apontando para uma mesma imagem.

DATAFRAME FINAL

Após a conclusão da limpeza dos *dataframes* citados acima, foi criado um principal para análise e será salvo como ‘twitter_archive_master.csv’. Este *dataframe* consiste no acréscimo das colunas ‘retweet_count’ e ‘favorite_count’ do *dataframe* ‘df_tweets_api’ ao Arquivo de Tweets (‘df_arch_clean’), e também a coluna ‘prediction_breed’ do *dataframe* Image prediction que forem marcados como cães, coluna ‘is_a_dog’. Assim, no *dataframe* final constará as seguintes colunas: tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, text, rating_numerator, rating_denominator, name, category, retweet_count, favorite_count e prediction_breed.