Wrangle Report - Projeto de Data Wrangling

para #WeRateDogs *tweets*

1. Reunindo os dados

Nesta fase, o mais desafiador foi reunir os dados através do *tweepy*, a API do Twitter. Após consultar o Stack Overflow, a própria documentação da API e algumas tentativas, foi possível obter e salvar todos os dados.

2. Investigando os dados

Esta fase foi bem complicada. A princípio, iniciei a fase de avaliação com 8 problemas de qualidade, que ao longo do tempo evoluiu para 14. Investiguei dados duplicados, tipos de dados inválidos, valores nulos, colunas com excessiva informação nula, conteúdo dos campos (alterei, em algum momento, para mostrar todo o conteúdo dos campos, porque haviam campos com muita informação) e agrupei valores de algumas colunas que aparentavam não estar devidamente preenchidas (por exemplo, coluna com todos os campos preenchidos, mas com campos preenchidos com *None*).

Um problema identificado logo no final do projeto foi classificação inválida, onde alguns *rating_numerators* e *rating_denominators* não estavam preenchidos com a classificação correta. Neste caso, foi necessário avaliar cada caso e corrigir conforme os erros eram encontrados, através da avaliação de *outliers*. Numeradores variavam, principalmente, em torno de 7 a 14 e denominadores eram, em sua maioria, 10. Qualquer valor fora desta variação era analisado.

3. Avaliando

Com a investigação concluída, foi necessário avaliar os problemas identificados. Ao total, foram 14 problemas de qualidades e 2 de arrumação.

Em arrumação, notei que uma variável, a que classifica a "fase" do cachorro, estava separada em 4 colunas (*doggo*, *floofer*, *pupper* e *puppo*). O outro problema era a necessidade de juntar as informações de quantidade de favoritos e *retweets* de um determinado *tweet* com o próprio *dataframe* de *tweets* (afinal, estas informações compõe um *tweet*).

4. Limpando

A fase de limpeza foi, também, desafiadora. Por várias vezes consultei o Stack Overflow para corrigir os 14 problemas de qualidade identificados. Alguns casos mais desafiadores foram: remover as *URLs* repetidas da *expanded_urls*, criar a função *dog_prediction_only* e usá-la no método *apply* do *dataframe* (principalmente porque neste caso realizei o retorno de duas colunas). Usar *loc* foi um pouco desafiador, porque havia me esquecido de como usá-lo, sendo necessário relembrar.

Em arrumação, criar uma nova coluna *dog_phase* contendo as fases do cachorro (*doggo*, *floofer*, *pupper* or *puppo*) foi muito desafiadora, já que não tinha prática em usar métodos *apply*, *map* ou *applymap*.

5. Visualização

O mais complicado nesta fase foi determinar "o que" visualizar. Resolvi aproveitar as predições e entender o comportamento dos seguidores e do próprio #WeRateDogs. Selecionei as 5 raças com maior quantidade de *tweets* para análise.

Para criar os gráficos, foi necessário realizar uma boa pesquisa no Python Graph Gallery. Ao final, obtive resultados satisfatórios e bem interessantes, como raças com maior quantidade de favoritos, *retweets* e classificação.