Rating Dogs – Fase de Wrangling

Por Nadsoon Brito Gondim

Iniciamos o wrangling com a coleta dos dados. O arquivo “twitter-archive-enhanced” foi baixado da internet manualmente, de modo que foi necessário apenas carrega-lo dentro do ambiente de trabalho, no dataframe “twitter_archive”. Por sua vez, o download do “arquivo image-predictions.tsv” foi feito programaticamente. Depois de baixá-lo, carregamos a tabela no dataframe “image_predictions”. Ambos os arquivos foram fornecidos pela Udacity.

Feito isso, acessando a API Tweepy, armazenamos os dados 'json' de cada tweet em um arquivo de texto 'tweet_json.txt'. Contudo, alguns tweets retornaram mensagem de erro, de modo que os dados não puderam ser carregados. A partir do arquivo “txt”, construímos o dataframe “df_tweepy”.

Com todos os dados já coletados, começamos a avaliá-los para tentar encontrar algum problema de qualidade ou de estrutura. Nesse momento, encontramos, ao todo, onze problemas de qualidade e quatro de estrutura. Passamos então à fase de limpeza.

O primeiro passo da limpeza foi criar uma cópia para cada dataframe com que estamos trabalhando.

Dos problemas de qualidade encontrados, com relação a três deles, nada pudemos fazer para solucioná-los, pois diziam respeito à ausência de dados, aos quais não temos acesso. Outros problemas de qualidade estavam relacionados a utilização de tipos de variável inadequados para representar os dados. Para solucioná-los, apenas modificamos os tipos de variável.

Uma outra situação problemática era a de variáveis com valores que não representam efetivamente o conteúdo que deveriam conter. Foi necessário, então, substituir esses valores pelo valor nulo.

Houve um caso em que a avaliação dada ao cachorro na postagem tinha denominador igual a zero, impossibilitando efetuar a divisão do numerador pelo

denominador. Infelizmente, para corrigir o problema, foi preciso excluir da tabela “twitter_archive” a linha correspondente a esse tweet.

Além desses citados, houve outros problemas de qualidade, mas foram também corrigidos.

Quanto aos problemas de estrutura, o primeiro deles foi a existência, na tabela “twitter_archive”, de uma coluna para o numerador e outra para o denominador, sendo que o correto seria existir uma única coluna contendo o resultado da divisão dos valores daquelas. Para resolver a situação, criamos uma nova coluna, chamada “rating”, que tem como valor o resultado da dita divisão, e deletamos as duas colunas originais (a do numerador e a do denominador).

Também havia o problema da multiplicidade de colunas para representar os estágios de vida dos cachorros. Resolvemos, então, criar uma coluna nova que contenha, em si, todas as informações, e apagamos as quatro colunas originais problemáticas.

Outro problema de estrutura encontrado foi que os dados que coletamos da API Tweepy não deveriam formar uma tabela a parte, mas sim deveriam fazer parte da tabela “twitter_archive”. Tivemos então que mesclar essas duas tabelas, de modo que toda a informação fique reunida.

Ainda quanto à estrutura, tínhamos que solucionar o problema de que a ordem das predições do algoritmo (1ª, 2ª e 3ª predição) não estava representada em uma coluna própria, estava espalhada por diferentes colunas. Foi preciso criar uma nova coluna só para a variável, além de outros procedimentos adicionais.

Completados todos esses passos, foi detectado um novo problema de estrutura: a presença de valores totalmente fora do esperado na variável “rating”. Para concertar isso, apagamos as linhas com os valores problemáticos.

Com isso, foram resolvidos todos os problemas de qualidade e de estrutura e os dados com que estamos trabalhando ficaram prontos para serem explorados. Salvamos, então, as tabelas “twitter_archive_clean” e

“image_predictions_clean” em novos arquivos “csv”. Encerramos, assim, a fase de wrangling.