Relatório de Data Wrangling

Durante a coleta dos dados utilizamos alguns métodos para realizar a coleta completa, para o arquivo baixado, bastava utilizar o método *read_csv* do pandas para importar para um *DataFrame*. Já o segundo arquivo, foi necessário utilizar a biblioteca request com o método get passando a url do arquivo, e então seguia basicamente os mesmos passos do primeiro arquivo, mudando apenas o tipo de separadores do arquivo. Por fim, o mais trabalhoso, definimos as chaves de consumo e acesso para ter acesso a API do *twitter*, e por meio da biblioteca tweepy foi possível obter os dados. No objeto recebido, pela requisição da API, bastava invocar o atributo ._json, que os dados estavam prontos para uso, bastava filtrar quais colunas deveriam ser utilizadas, assim salvar em um arquivo txt, que então poderia ser lido com o método *read_json* do pandas.
Partimos então para a fase de acesso, a primeira tabela buscada foi a que mais precisou de tratamento de qualidade, continha todos os tweets contando retweets e comentários, existiam colunas excedentes que não seriam utilizadas, além da conversão dos tipos de dados, colunas erroneamente preenchidas com "None" onde deveria ser nulo, erros nos campos das avaliações, colunas de difícil leitura (que foram removidas no final quando percebi que não apresentavam dados concisos para análise). Já na segunda tabela (de informações das imagens) tinham poucos ajustes a serem feitos, principalmente na clareza do nome das colunas, tipos de dados e colunas em excesso que não deveriam ser levadas em consideração. A terceira coluna tinha apenas dados inteiros vindos da API, fora passar o *tweet_id* para string, não teve nenhuma outra alteração.
Por fim na limpeza fizemos também segmentada por cada tabela, solucionando cada um dos problemas passados para por fim juntar todas tabelas na tabela final. A limpeza da primeira tabela foi bem trabalhosa, onde fui no caso da correção das notas em cada um dos tweets para verificar a situação do erro e assim poder corrigi-los isoladamente. Uma observação interessante é que eu planejei agregar as colunas *doggo*, *floofer*, *pupper* e *puppo* em apenas uma. Entretanto enquanto eu estava testando, verifiquei que haviam linhas em que existiam dois dos atributos selecionados e tive que voltar ao acesso e trocar a solução para trocar o tipo de dado da tabela para booleano para não perder estes dados (e de fato eu verifiquei as imagens, e as que tinham *doggo* e *pupper* marcados realmente tinham dois cachorros). Para a segunda tabela a questão de limpar mais trabalhosa nesta tabela foi levar em consideração só a previsão de maior precisão em que era relativa a alguma raça de cachorro que acabei fazendo uma solução inversa atribuindo os atributos de p3 para p2, então de p2 para p1, para ter apenas uma coluna concisa. E por fim as três tabelas foram mescladas de tal forma que apenas os tweet_id presente nas três fossem levados em consideração para a composição final.