RELATÓRIO DATA WRANGLING

Data Wrangling está baseado em três pilares: a coleta de dados, a avaliação de possíveis “problemas” presentes nos dados coletados e consequentemente a limpeza dos dados para posteriormente uma ação de análise revelar o que tais dados podem nos dizer (esta última não faz parte do *Data Wrangling*).

Para este projeto a conta do *Twitter WeRateDogs* (conta que classifica os cães com comentários bem humorados) cedeu os dados de *tweets* dos anos 2015, 2016 e 2017 (não necessariamente o ano completo).

Coleta

Neste projeto foi coletado dados (*DataFrames*) de três fontes diferentes:

- Um arquivo dado pela *Udacity* (*twitter-archive-enhanced.csv*), que bastava baixá-lo manualmente e salvo-lo na pasta do projeto e posteriormente, no *Notebook Jupyter*, com a ajuda da biblioteca *Pandas* através da função **.read_csv( )* ler e carregar o arquivo;
- Uma pasta que contém um arquivo que estava presente nos servidores do *Udacity* (*image_predictions/image-predictions.tsv*), na qual era necessário baixá-lo programaticamente e em seguida através da função **.read_csv(* , sep='\t')* ler e carregar o arquivo;
- E para finalizar era necessário através de *API* do *Twitter* (através da biblioteca do *Python*, *Tweepy*), realizar o download de informações de *tweets* presentes no arquivo *twitter-archive-enhanced.csv* (coluna *tweet_id*) e através da biblioteca *JSON* carregar e salvar (*tweet_json.txt*) tal arquivo como **.txt.*

Avaliação

Após a etapa de coleta de dados, a etapa da avaliação é uma etapa de suma importância, pois é nela que será avalizado os *DataFrames* para localizar possíveis “problemas” presentes que podem ocasionar análises futuras imprecisas ou até mesmo errôneas. É nesta etapa que colocamos todas as observações a serem feitas na etapa da limpeza.

O arquivo *twitter-archive-enhanced.csv* obteve a maior quantidade de modificação a ser feita na parte da limpeza, totalizando 10 problemas (6 problemas de qualidade e 4 de arrumação).

Para o arquivo *image-predictions.tsv* 3 problemas de qualidade e 1 de arrumação.

E para o *tweet_json.txt* 3 problemas de qualidade e 3 de arrumação.

Para essa avaliação foi utilizado comandos como:

- *.head( ):* mostra, se não especificado entre os parentes, 5 linhas do *DataFrame*;
- *.info( ):* mostra informações de tipo de dados presentes na coluna e quantidade de dados não vazios (*null*) por coluna;
- *.describe( ):* mostra um quadro estatístico dos dados (*int* e *float*), como média, máximo, mínimo, mediana, entre outros.
- Entre outros...

Limpeza

Após a avaliação ser realizado, é a hora da limpeza.

Primeiramente (por segurança), devemos realizar um cópia dos *DataFrames* originais, através da função *.copy( ).*

Posteriormente, definir o que será feito em cada passo (com a ajuda do resumo realizado na avaliação) e realizar a codificação para que aquela alteração seja realizada no *DataFrame* de segurança (ou de limpeza, *Dataframe_clean*). Após a codificação devemos reavaliar se a alteração foi realizada com sucesso; para isso devemos utilizar as funções mencionadas na tópica avaliação.

Observação: sempre que preciso, a etapa de avaliação deve ser reconsultada.

Nesta etapa foram criadas funções *'def'* para realizar a limpeza de forma mais eficiente. E uma das operações que estava presente em ambos os *DataFrame* foi a alteração do tipo de dado, pois os dados da coluna *tweet_id* estavam como *int*, porém com se trata de uma dado que não será realizado operações matemáticas, ele deve ser uma string (*str*).

Em suma, para esse projeto a etapa de coleta de dados através do *API* foi a etapa que deu mais trabalho, por se tratar de uma biblioteca nova e procedimentos novos. Na etapa de avaliação, a distinção de problemas de qualidade e arrumação é a parte que há mais confusão. Na etapa de limpeza, a documentação do *Python* ajuda muito para saber como tal função deve ser aplicada.