Visão geral do projeto

Preparar e analisar Dados

Introdução

Dados do mundo real raramente vem limpos. Colete dados de uma série de fontes em uma variedade de formatos, avalie sua qualidade e arrumação e, então, limpe-os. Isso é chamado de data wrangling. Você irá documentar seus esforços de wrangling em um notebook Jupyter, além de exibi-los por meio de análises e visualizações usando Python e/ou SQL.

O conjunto de dados no qual você irá fazer o wrangling (e analisar e visualizar) é o arquivo de tweets do usuário do Twitter @dog_rates, também conhecido como WeRateDogs. WeRateDogs é uma conta no Twitter que classifica os cães das pessoas com um comentário bem humorado sobre o cão. Ele foi iniciado em 2015 pelo estudante universitário Matt Nelson e recebeu cobertura da mídia internacional. Em outubro de 2017 tinha mais 3,7 milhões de seguidores. Estas classificações têm quase sempre um denominador de 10. Mas e os numeradores? Quase sempre maior que 10. 11/10, 12/10, 13/10, etc. Por quê? Porque "são bons cães, Brent."
WeRateDogs deu à Udacity acesso exclusivo a seu arquivo tweets para este projeto. Este arquivo contém dados básicos de tweets para todos os seus mais de 5000 tweets como eles estavam em 1 de agosto de 2017. Voltaremos a esse assunto em breve.

Image via Boston Magazine

Qual é o software de que eu preciso?

Você precisa ser capaz de trabalhar em um notebook Jupyter em seu computador. Revisite nossos tutoriais sobre notebook Jupyter e Anaconda que apareceram anteriormente no programa Nanodegree para instruções de instalação.

Os seguintes pacotes (ou seja, bibliotecas) são necessários para concluir este projeto:

- pandas
- numpy
- requests
- tweepy
- json

Você pode instalar esses pacotes via conda ou pip. Revisite nosso tutorial sobre Anaconda que apareceu anteriormente no programa Nanodegree para instruções de instalação de pacotes.

Você também precisa ser capaz de criar documentos escritos com imagens que possam ser exportados como arquivos PDF. Exemplo: Google Docs, que é gratuito.
Um editor de texto, como Sublime (que é gratuito), será útil, mas não é necessário.

Motivação do projeto

Meta

Sua meta: fazer wrangling dos dados de tweets de WeRateDoga para que você possa criar análises e visualizações interessantes e confiáveis. Sim, WeRateDogs deu à Udacity acesso exclusivo a seu arquivo de tweets, mas ele contém informações muito básicas. Coleta, avaliação e limpeza adicionais são necessárias para análises e visualizações que mereçam uma reação de "*Uau!*"

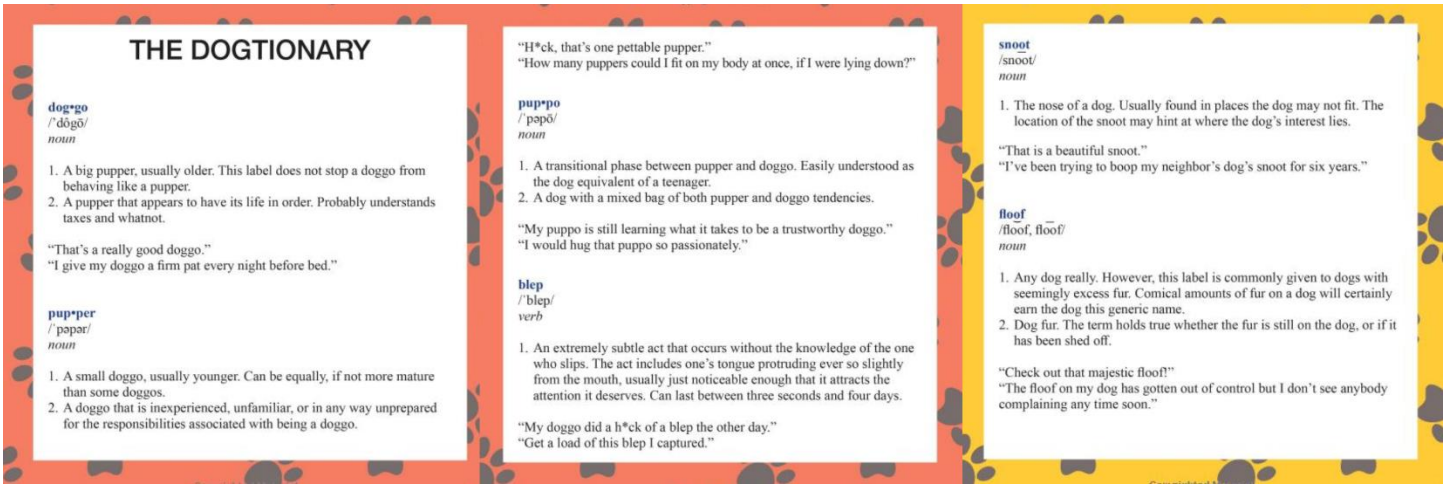
Contexto

Este arquivo contém dados básicos de tweets para os mais de 5000 de seus tweets, mas não tudo. Uma coluna o arquivo contém com certeza: cada texto de tweet, o que eu usei para extrair classificação, nome e "estágio" do cachorro (ou seja, doggo, floofer, pupper e puppo).

text	rating_ numerator	rating_ denominator	name	doggo	floofer	pupper	puppo
This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76dJU	13	10	Phineas	None	None	None	None
This is Tilly. She's just checking pup on you. Hopes you're doing ok. If not, she's available for pats, snugs, boops, the whole bit. 13/10	13	10	Tilly	None	None	None	None
This is Archie. He is a rare Norwegian Pouncing Corgo. Lives in the tall grass. You never know when one may strike. 12/10 https://t.co	12	10	Archie	None	None	None	None
This is Darla. She commenced a snooze mid meal. 13/10 happens to the best of us https://t.co/tD36da7qLQ	13	10	Darla	None	None	None	None
This is Franklin. He would like you to stop calling him "cute." He is a very fierce shark and should be respected as such. 12/10 #BarkW	12	10	Franklin	None	None	None	None
Here we have a majestic great white breaching off South Africa's coast. Absolutely h*ckin breathtaking. 13/10 (IG: tucker_marlo) #Bar	13	10	None	None	None	None	None
Meet Jax. He enjoys ice cream so much he gets nervous around it. 13/10 help Jax enjoy more things by clicking below https://t.co/Zr4hWfAs1H https://t.co/tVJBRMnhxl	13	10	Jax	None	None	None	None
When you watch your owner call another dog a good boy but then they turn back to you and say you're a great boy. 13/10 https://t.co/	13	10	None	None	None	None	None
This is Zoey. She doesn't want to be one of the scary sharks. Just wants to be a snuggly pettable boatpet. 13/10 #BarkWeek https://t.c	13	10	Zoey	None	None	None	None
This is Cassie. She is a college pup. Studying international doggo communication and stick theory. 14/10 so elegant much sophisticat	14	10	Cassie	doggo	None	None	None
This is Koda. He is a South Australian deckshark. Deceptively deadly. Frighteningly majestic. 13/10 would risk a petting #BarkWeek ht	13	10	Koda	None	None	None	None
This is Bruno. He is a service shark. Only gets out of the water to assist you. 13/10 terrifyingly good boy https://t.co/u1XPQMl29g	13	10	Bruno	None	None	None	None
Here's a puppo that seems to be on the fence about something haha no but seriously someone help her. 13/10 https://t.co/BxvuXk0UC	13	10	None	None	None	None	puppo
This is Ted. He does his best. Sometimes that's not enough. But it's ok. 12/10 would assist https://t.co/f8dEDcrKSR	12	10	Ted	None	None	None	None
This is Stuart. He's sporting his favorite fanny pack. Secretly filled with bones only. 13/10 puppared puppo #BarkWeek https://t.co/y70	13	10	Stuart	None	None	None	puppo

Os dados extraídos do texto de cada tweet

Extraí esses dados por meio de programação, mas não fiz um trabalho muito bom. As avaliações provavelmente não estão todas corretas. O mesmo acontece para os nomes e estágios de cachorros. Você precisará avaliar e limpar essas colunas se você quiser usá-las para análise e visualização.



O Dogtionary explica os diversos estágios do cão: doggo, pupper, puppo e floof(er) (pelo livro #WeRateDogs na Amazon)

De volta ao básico desse conjunto de dados: a contagem de retweets e favoritos são duas colunas omissas notáveis. Felizmente, esses dados adicionais podem ser coletados da API do Twitter, mas, só porque você tem acesso a IDs no arquivo de tweets (sem essas identificações, o público só tem acesso aos últimos ~3000 tweets pela API do Twitter). Adivinha só? Você vai reunir esses dados adicionais.

Mais uma coisa legal: usando uma rede neural que consegue classificar as raças de cachorros*, executei cada imagem no arquivo do Twitter de WeRateDogs. Os resultados: uma tabela cheia de previsões de imagens ao lado de cada ID de tweet, URL de imagem e o número de imagem de corresponde à previsão mais confiante (1 a 4, já que tweets podem ter quatro imagens).

tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
892177421306343426	https://pbs.twimg.cor	1	Chihuahua	0.323581	TRUE	Pekinese	0.0906465	TRUE	papillon	0.0689569	TRUE
891815181378084864	https://pbs.twimg.cor	1	Chihuahua	0.716012	TRUE	malamute	0.078253	TRUE	kelpie	0.0313789	TRUE
891689557279858688	https://pbs.twimg.cor	1	paper_towel	0.170278	FALSE	Labrador_retriever	0.168086	TRUE	spatula	0.0408359	FALSE
891327558926688256	https://pbs.twimg.cor	2	basset	0.555712	TRUE	English_springer	0.22577	TRUE	German_short-haired_pointer	0.175219	TRUE
891087950875897856	https://pbs.twimg.cor	1	Chesapeake_Bay_retriever	0.425595	TRUE	Irish_terrier	0.116317	TRUE	Indian_elephant	0.0769022	FALSE
890971913173991426	https://pbs.twimg.cor	1	Appenzeller	0.341703	TRUE	Border_collie	0.199287	TRUE	ice_lolly	0.193548	FALSE
890729181411237888	https://pbs.twimg.cor	2	Pomeranian	0.566142	TRUE	Eskimo_dog	0.178406	TRUE	Pembroke	0.0765069	TRUE
890609185150312448	https://pbs.twimg.cor	1	Irish_terrier	0.487574	TRUE	Irish_setter	0.193054	TRUE	Chesapeake_Bay_retriever	0.118184	TRUE
890240255349198849	https://pbs.twimg.cor	1	Pembroke	0.511319	TRUE	Cardigan	0.451038	TRUE	Chihuahua	0.0292482	TRUE
890006608113172480	https://pbs.twimg.cor	1	Samoyed	0.957979	TRUE	Pomeranian	0.0138835	TRUE	chow	0.00816748	TRUE
889880896479866881	https://pbs.twimg.cor	1	French_bulldog	0.377417	TRUE	Labrador_retriever	0.151317	TRUE	muzzle	0.0829811	FALSE
889665388333682689	https://pbs.twimg.cor	1	Pembroke	0.966327	TRUE	Cardigan	0.0273557	TRUE	basenji	0.00463323	TRUE
889638837579907072	https://pbs.twimg.cor	1	French_bulldog	0.99165	TRUE	boxer	0.00212864	TRUE	Staffordshire_bullterrier	0.00149818	TRUE
889531135344209921	https://pbs.twimg.cor	1	golden_retriever	0.953442	TRUE	Labrador_retriever	0.0138341	TRUE	redbone	0.00795775	TRUE

Dados de previsão de imagem de tweets

Assim, para a última linha dessa tabela:

- tweet_id é a última parte da URL após "*status/*"
  → https://twitter.com/dog_rates/status/889531135344209921
- p1 é a previsão número 1 do algoritmo para a imagem no tweet → **golden retriever**
- p1_conf é o quão confiante o algoritmo está nessa previsão número 1 → **95%**
- p1_dog é se a previsão número 1 é uma raça de cachorro → **TRUE**
- p2 é a segunda previsão mais provável do algoritmo → **Labrador retriever**
- p2_conf é o quão confiante esse algoritmo está nessa previsão número 2 → **1%**
- p2_dog é se a previsão número 2 é uma raça de cachorro → **TRUE**
- etc.
  E a previsão número um para a imagem naquele tweet estava correta:

WeRateDogs™ (author)

@dog_rates

Following

This is Stuart. He's sporting his favorite fanny pack. Secretly filled with bones only. 13/10 puppared puppo #BarkWeek

1:02 PM - 24 Jul 2017

Então tá tudo certo e muito bom. Mas todos esses dados adicionais, depois de coletados, precisarão ser avaliados e limpos. É aqui que você entra.

Pontos principais

Pontos-chave para ter mente quando ao fazer data wrangling para esse projeto:

- Só queremos classificações originais (não retweets) que têm imagens.
- Avaliar e limpar totalmente todo o banco de dados requer um esforço excepcional para que apenas um subconjunto de seus problemas (oito problemas de qualidade e dois problemas de arrumação) precisem ser avaliados e limpos.
- A limpeza inclui a fusão de acordo com as regras de dados arrumados para facilitar a análise e visualização.
  **Curiosidade: criar esta rede neural é um dos projetos do programa Nanodegree de Inteligência Artificial na Udacity.*

Detalhes do projeto

Detalhes do projeto

- Data wrangling, que consiste em:
- Coletar dados
- Avaliar dados
- Limpar dados
- Armazenar, analisar e visualizar seus dados wrangled
- Elaborar relatórios sobre 1) seus esforços de data wrangling 2) suas análises e visualizações de dados

Coletando dados para esse projeto

Colete cada um dos três pedaços de dados conforme descritos abaixo em um notebook Jupyter intitulado `wrangle_act.ipynb`:

1. O arquivo WeRateDogs. Estou dando este arquivo a você, então o imagine como um arquivo já em mãos. Baixe este arquivo manualmente clicando no link a seguir: `twitter_archive_enhanced.csv`
2. As previsões de imagens em tweets, isto é, qual raça de cachorro ou objeto inanimado está presente em cada tweet Este arquivo (`image_predictions.tsv`) está hospedado nos servidores da Udacity e devem ser baixados programaticamente usando a seguinte URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv
3. A contagem de retweets e favoritos (curtida) de cada tweet, no mínimo, e quaisquer dados adicionais que você achar interessantes. Faça uma consulta na API do Twitter (usando as ID de tweets no arquivo do Twitter de WeRateDogs) para o conjunto de dados completo de cada tweet usando a biblioteca Tweepy do Python e armazene esses dados em um arquivo chamado `tweet_json.txt`, onde os dados JSON armazenados de cada tweet devem ser escritos em sua própria linha. Então, leia este .txt linha por linha em um dataframe do Pandas com (no mínimo) ID de tweet, contagem de retweets e contagem de favoritos. *Obs.: não inclua suas chaves de API do Twitter e tokens de acesso no envio de seu projeto.*

Avaliando dados para esse projeto

Após coletar cada um dos pedaços de dados acima, os avalie visualmente e por meio de programação para problemas de qualidade e arrumação. Detecte e documente ao menos **oito (8) problemas de qualidade** e **dois (2) problemas de arrumação** em seu notebook Jupyter. Para atender às especificações, os problemas que satisfazem a motivação do projeto (ver página anterior) devem ser avaliados.

Limpando dados para esse projeto

Limpe cada um dos problemas que você documentou enquanto avaliava. O resultado deve ser um dataframe master do Pandas (ou dataframes, caso seja apropriado) de alta qualidade e arrumado. De novo, os problemas que satisfazem a motivação do projeto devem ser limpos.

Armazenando, analisando e visualizando dados para este projeto

Armazene o(s) dataframe(s) limpo(s) em um arquivo CSV, com o principal deles intitulado `twitter_archive_master.csv`. Se adicionais existirem, os nomeie de forma apropriada.

Além disso, você pode armazenar os dados limpos em um banco de dados SQLite (que também deve ser enviado, caso você o faça).
Analise e visualize seus dados wrangled em seu notebook Jupyter. Pelo menos **três (3) insights e uma (1) visualização** devem ser produzidos.

Fazendo relatórios para este projeto

Crie um relatório de 300-600 palavras chamado `wrangle_report.pdf` que descreva brevemente seus esforços de wrangling. Ele deve ser enquadrado como um documento interno.
Crie um relatório escrito de mais de 250 palavras, chamado `act_report.pdf`, que comunique os insights e exiba a(s) visualização(ões) produzida(s) por seus dados wrangled. Ele deve ser enquadrado como um documento externo.

API do Twitter

Como consultar dados do Twitter

Neste projeto, você usará o Tweepy para consultar o API do Twitter por dados adicionais, além dos dados incluídos no arquivo WeRateDogs do Twitter. Esses dados adicionais incluirão a contagem de quantas vezes cada mensagem teve um retweet e quantas vezes foi marcada como favorita.
Algumas APIs são completamente abertas, como a MediaWiki (acessada através da biblioteca wptools) na Aula 2. Outras requerem autenticação. A API do Twitter é uma que requer que os usuários estejam autenticados para utilizá-la. Isso significa que antes de poder usar o código de consulta da API, você precisa configurar sua própria aplicação do Twitter. Aqui estão os passos para fazer isso na página do Twitter:

- Primeiro, se você ainda não tem uma conta, você precisa criar uma conta do Twitter.
- Em seguida, para criar uma conta de desenvolvedor, siga as direções no Portal do Desenvolvedor do Twitter, na seção “How to Apply” (“Como Solicitar”).
- Você será guiado por uma série de passos, e pedirão que você descreva em suas próprias palavras o que você está construindo. Aqui está uma sugestão de texto que você pode usar: "As a Udacity student, I need to access the Twitter API in order to complete a Data Wrangling student project. In this project, I'll be using Tweepy to query Twitter's API for data included in the WeRateDogs Twitter archive. This data will include retweet count and favorite count. Before I can run my API querying code, I need to set up my own Twitter application. Once I have this set up, I will develop some code to create an API object that I'll use to gather Twitter data. After querying each tweet ID, I will write its JSON data to a tweet_json.txt file with each tweet's JSON data on its own line. I will then read this file, line by line, to create a pandas DataFrame that I will assess and clean. I may post this completed project on my GitHub account, where it will get viewers. Otherwise there will be no other readers or users of my Twitter data or project analysis beyond the Udacity instructors and reviewers."
- Após submeter sua aplicação, você deverá receber um e-mail do Twitter, lhe avisando que eles aprovaram sua conta de desenvolvedor do Twitter. Siga o link no e-mail para começar a criar sua aplicação.
- Se lhe pedirem um nome para sua aplicação, você pode dar qualquer nome que julgar apropriado. Se lhe pedirem uma URL de site, pode ser qualquer coisa em um formato URL. Você pode deixar todas as outras URLs em branco.
- Se lhe pedirem para explicar como sua aplicação será usada, você pode dizer algo como “Eu estou criando essa aplicação para um projeto de Data Wrangling na Udacity, no qual eu preciso consultar e analisar dados do Twitter de WeRateDogs.”
- Você deve então receber uma mensagem de Sucesso, e uma nova página de desenvolvedor será exibida para você, na qual você pode gerenciar sua aplicação.
- Após isso, você pode ir para a aba “Keys and Tokens”, para visualizar ou gerar as chaves de API de Consumidor, e o Access Token e Access Token Secret que você precisará.
(Nota: Se você não conseguir configurar uma aplicação por problemas de verificação, por favor mande um email para o Rick Gaston em rick@udacity.com)

Uma vez que você tiver sua conta do Twitter e seu app do Twitter configurados, o código a seguir, que é fornecido na seção de Getting started da documentação do Tweepy, irá criar um objeto API que você pode usar para acumular dados do Twitter.

```
import tweepy

consumer_key = 'YOUR CONSUMER KEY'
consumer_secret = 'YOUR CONSUMER SECRET'
access_token = 'YOUR ACCESS TOKEN'
access_secret = 'YOUR ACCESS SECRET'
```

```
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth)
```

Dados de tweets são armazenados em formato JSON pelo Twitter. A técnica de pegar dados JSON de tweets usando o ID deles com o Tweepy é descrita nesta resposta do StackOverflow. Note que colocar o parâmetro `tweet_mode` em `'extended'` na chamada `get_status` pode ser útil. (exemplo: `api.get_status(tweet_id, tweet_mode='extended')`)
Note também que há tweets que correspondem a alguns IDs no arquivo que já podem ter sido deletados. Blocos de try-except podem ser úteis nesses casos.

Não inclua suas chaves API, Secrets, e Tokens no seu envio

Não inclua suas chaves (keys) de API, seus secrets, e tokens no envio de seu projeto. Esta é uma prática padrão para APIs e código público.

Limites do Twitter

O API do Twitter tem um limite de taxa de consulta. Isso é usado para controlar a taxa de tráfego enviado ou recebido pelo servidor. Há uma página de informações de limites do Twitter que explicita isso:
Rate limits are divided into 15 minute intervals | A taxa de limite é dividida em intervalos de 15 minutos.

Para consultar todos os IDs de tweets no arquivo WeRateDogs do Twitter, espere um tempo de execução entre 20 e 30 minutos. Imprimir cada ID de tweet depois de consultá-los e usar um timer é útil por razões de sanidade. Configurar os parâmetros `wait_on_rate_limit` e `wait_on_rate_limit_notify` para `True` na classe `tweepy.api` tam bém é útil.

Escrevendo e lendo JSON do Twitter

Depois de consultar cada ID, você irá escrever os dados JSON para o arquivo requerido `tweet_json.txt` com os dados JSON de cada tweet em sua própria linha separada. Você irá então ler este arquivo, linha por linha, para criar um DataFrame de Pandas que você irá avaliar e limpar. Este artigo sobre Ler e Escrever JSON para um arquivo em Python do Stack Abuse será útil.

Preparar e analisar dados

Project Submission

Neste projeto, você irá coletar, avaliar e limpar dados e então partir para a realização de análises, visualização e/ou construção de modelos.

Antes de enviar o seu projeto:

1. Certifique-se que seu projeto atende as especificações de todos os itens da Rubrica do Projeto. Seu projeto "atende às especificações" somente se todos critérios forem atendidos.
2. Certifique-se que você **não incluiu** suas chaves API, secrets e tokens nos seus arquivos do projeto.
3. *Se você completou seu projeto no **Workspace do Projeto***, certifique-se que os arquivos a seguir estão no seu workspace e então clique em "Enviar Projeto" no canto inferior direito da página do **Workspace do Projeto**:

- `wrangle_act.ipynb`: código para coletar, avaliar, limpar, analisar e visualizar os dados
- `wrangle_report.pdf` ou `wrangle_report.html`: documentação para os passos de data wrangling: coletar, avaliar e limpar
- `act_report.pdf` ou `act_report.html`: documentação das análises e insights sobre os dados finais
- `twitter_archive_enhanced.csv`: arquivo conforme fornecido
- `image_predictions.tsv`: arquivo baixado através do seu programa
- `tweet_json.txt`: arquivo construído via API
- `twitter_archive_master.csv`: dados limpos e combinados
- quaisquer arquivos adicionais (e.g. arquivos para pedaços adicionais de dados ou um arquivo de banco de dados para seu dados limpos armazenados)

4. *Se você completou seu projeto fora da Sala de Aula da Udacity*, comprima os arquivos listados acima em um arquivo ZIP ou armazene em um repositório do GitHub, então clique no botão "Enviar Projeto" dessa página.
   Como dissemos no *ponto 4* acima, você pode enviar seu projeto como um arquivo ZIP ou indicar um repositório no GitHub que contém seus arquivos do projeto. Se você usar o GitHub, saiba que o seu envio será uma cópia do repositório indicado realizada no instante do envio. Recomendamos que você mantenha cada projeto em um repositório separado para evitar qualquer possível confusão: se o revisor receber várias pastas com múltiplos projetos, ele pode se confundir sobre qual é o projeto que deve ser avaliado.
   Pode ser que a revisão do projeto demore até uma semana, mas na maioria dos casos este processo é bem mais rápido. Você receberá um e-mail quando o seu envio tiver sido revisado. Caso você tenha algum problema no envio do seu projeto ou queira checar o status do seu envio, envie uma mensagem para suporte@udacity.com. Por hora, fique à vontade para prosseguir com sua jornada de aprendizagem, seguindo para o próximo módulo do programa.