ETL Pipeline Preparation

April 16, 2019

1 ETL Pipeline Preparation

Follow the instructions below to help you create your ETL pipeline. ### 1. Import libraries and load datasets. - Import Python libraries - Load `messages.csv` into a dataframe and inspect the first few lines. - Load `categories.csv` into a dataframe and inspect the first few lines.

In [1]:
```
# import libraries
import pandas as pd
from sqlalchemy import create_engine

%matplotlib inline
```

In [2]:
```
# load messages dataset
messages = pd.read_csv('messages.csv')
messages.head()
```

Out[2]:

	id	message	original	genre
0	2	Weather update - a cold front from Cuba that c...	Un front froid se retrouve sur Cuba ce matin. ...	direct
1	7	Is the Hurricane over or is it not over	Cyclone nan fini osinon li pa fini	direct
2	8	Looking for someone but no name	Patnm, di Maryani relem pou li banm nouvel li ...	direct
3	9	UN reports Leogane 80-90 destroyed. Only Hospi...	UN reports Leogane 80-90 destroyed. Only Hospi...	direct
4	12	says: west side of Haiti, rest of the country ...	facade ouest d Haiti et le reste du pays aujou...	direct

In [3]:
```
messages.shape
```

Out[3]:
```
(26248, 4)
```

In [7]:
```
# load categories dataset
categories_raw = pd.read_csv('categories.csv')
categories_raw.head()
```

```
Out[7]:    id                                          categories
        0   2  related-1;request-0;offer-0;aid_related-0;medi...
        1   7  related-1;request-0;offer-0;aid_related-1;medi...
        2   8  related-1;request-0;offer-0;aid_related-0;medi...
        3   9  related-1;request-1;offer-0;aid_related-1;medi...
        4  12  related-1;request-0;offer-0;aid_related-0;medi...
```

1.0.1 2. Merge datasets.

- Merge the messages and categories datasets using the common id
- Assign this combined dataset to `df`, which will be cleaned in the following steps

```
In [8]: # merge datasets
        df = pd.merge(messages, categories_raw, on='id')
        df.head()
```

```
Out[8]:    id                                            message  \
        0   2  Weather update - a cold front from Cuba that c...
        1   7            Is the Hurricane over or is it not over
        2   8                    Looking for someone but no name
        3   9  UN reports Leogane 80-90 destroyed. Only Hospi...
        4  12  says: west side of Haiti, rest of the country ...

                                              original   genre  \
        0  Un front froid se retrouve sur Cuba ce matin. ...  direct
        1                Cyclone nan fini osinon li pa fini  direct
        2  Patnm, di Maryani relem pou li banm nouvel li ...  direct
        3  UN reports Leogane 80-90 destroyed. Only Hospi...  direct
        4  facade ouest d Haiti et le reste du pays aujou...  direct

                                             categories
        0  related-1;request-0;offer-0;aid_related-0;medi...
        1  related-1;request-0;offer-0;aid_related-1;medi...
        2  related-1;request-0;offer-0;aid_related-0;medi...
        3  related-1;request-1;offer-0;aid_related-1;medi...
        4  related-1;request-0;offer-0;aid_related-0;medi...
```

1.0.2 3. Split `categories` into separate category columns.

- Split the values in the `categories` column on the `;` character so that each value becomes a separate column. You'll find this method very helpful! Make sure to set `expand=True`.
- Use the first row of categories dataframe to create column names for the categories data.
- Rename columns of `categories` with new column names.

```
In [9]: # create a dataframe of the 36 individual category columns
        categories = categories_raw.categories.str.split(';', expand=True)
        categories.head()
```

Out[9]:

	0	1	2	3	4
0	related-1	request-0	offer-0	aid_related-0	medical_help-0
1	related-1	request-0	offer-0	aid_related-1	medical_help-0
2	related-1	request-0	offer-0	aid_related-0	medical_help-0
3	related-1	request-1	offer-0	aid_related-1	medical_help-0
4	related-1	request-0	offer-0	aid_related-0	medical_help-0

	5	6	7	8
0	medical_products-0	search_and_rescue-0	security-0	military-0
1	medical_products-0	search_and_rescue-0	security-0	military-0
2	medical_products-0	search_and_rescue-0	security-0	military-0
3	medical_products-1	search_and_rescue-0	security-0	military-0
4	medical_products-0	search_and_rescue-0	security-0	military-0

	9	...	26	27
0	child_alone-0	...	aid_centers-0	other_infrastructure-0
1	child_alone-0	...	aid_centers-0	other_infrastructure-0
2	child_alone-0	...	aid_centers-0	other_infrastructure-0
3	child_alone-0	...	aid_centers-0	other_infrastructure-0
4	child_alone-0	...	aid_centers-0	other_infrastructure-0

	28	29	30	31	32	33
0	weather_related-0	floods-0	storm-0	fire-0	earthquake-0	cold-0
1	weather_related-1	floods-0	storm-1	fire-0	earthquake-0	cold-0
2	weather_related-0	floods-0	storm-0	fire-0	earthquake-0	cold-0
3	weather_related-0	floods-0	storm-0	fire-0	earthquake-0	cold-0
4	weather_related-0	floods-0	storm-0	fire-0	earthquake-0	cold-0

	34	35
0	other_weather-0	direct_report-0
1	other_weather-0	direct_report-0
2	other_weather-0	direct_report-0
3	other_weather-0	direct_report-0
4	other_weather-0	direct_report-0

[5 rows x 36 columns]

In [10]:
```
# select the first row of the categories dataframe
row = categories[:1]

# use this row to extract a list of new column names for categories.
# one way is to apply a lambda function that takes everything
# up to the second to last character of each string with slicing
category_colnames = row.applymap(lambda s: s[:-2]).iloc[0, :].tolist()
print(category_colnames)
```

```
['related', 'request', 'offer', 'aid_related', 'medical_help', 'medical_products', 'search_and_r
```

```
In [11]: # rename the columns of `categories`
         categories.columns = category_colnames
         categories.head()
```

Out[11]:

	related	request	offer	aid_related	medical_help	medical_products	search_and_rescue	security	military	child_alone	...	aid_centers	other_infrastructure	weather_related	floods	storm	fire	earthquake	cold	other_weather	direct_report
0	related-1	request-0	offer-0	aid_related-0	medical_help-0	medical_products-0	search_and_rescue-0	security-0	military-0	child_alone-0	...	aid_centers-0	other_infrastructure-0	weather_related-0	floods-0	storm-0	fire-0	earthquake-0	cold-0	other_weather-0	direct_report-0
1	related-1	request-0	offer-0	aid_related-1	medical_help-0	medical_products-0	search_and_rescue-0	security-0	military-0	child_alone-0	...	aid_centers-0	other_infrastructure-0	weather_related-1	floods-0	storm-1	fire-0	earthquake-0	cold-0	other_weather-0	direct_report-0
2	related-1	request-0	offer-0	aid_related-0	medical_help-0	medical_products-0	search_and_rescue-0	security-0	military-0	child_alone-0	...	aid_centers-0	other_infrastructure-0	weather_related-0	floods-0	storm-0	fire-0	earthquake-0	cold-0	other_weather-0	direct_report-0
3	related-1	request-1	offer-0	aid_related-1	medical_help-0	medical_products-1	search_and_rescue-0	security-0	military-0	child_alone-0	...	aid_centers-0	other_infrastructure-0	weather_related-0	floods-0	storm-0	fire-0	earthquake-0	cold-0	other_weather-0	direct_report-0
4	related-1	request-0	offer-0	aid_related-0	medical_help-0	medical_products-0	search_and_rescue-0	security-0	military-0	child_alone-0	...	aid_centers-0	other_infrastructure-0	weather_related-0	floods-0	storm-0	fire-0	earthquake-0	cold-0	other_weather-0	direct_report-0

[5 rows x 36 columns]

1.0.3 4. Convert category values to just numbers 0 or 1.

- Iterate through the category columns in df to keep only the last character of each string (the 1 or 0). For example, `related-0` becomes `0`, `related-1` becomes `1`. Convert the string to a numeric value.
- You can perform normal string actions on Pandas Series, like indexing, by including `.str` after the Series. You may need to first convert the Series to be of type string, which you can

do with `astype(str)`.

```
In [12]: categories = categories.applymap(lambda s: int(s[-1]))

In [13]: for column in categories:
             # set each value to be the last character of the string
             categories[column] = categories[column].astype(str).str[-1]

             # convert column from string to numeric
             categories[column] = categories[column].astype(int)
         categories.head()
```

Out[13]:

	related	request	offer	aid_related	medical_help	medical_products	\
0	1	0	0	0	0	0	
1	1	0	0	1	0	0	
2	1	0	0	0	0	0	
3	1	1	0	1	0	1	
4	1	0	0	0	0	0	

	search_and_rescue	security	military	child_alone	...	\
0	0	0	0	0	...	
1	0	0	0	0	...	
2	0	0	0	0	...	
3	0	0	0	0	...	
4	0	0	0	0	...	

	aid_centers	other_infrastructure	weather_related	floods	storm	fire	\
0	0	0	0	0	0	0	
1	0	0	1	0	1	0	
2	0	0	0	0	0	0	
3	0	0	0	0	0	0	
4	0	0	0	0	0	0	

	earthquake	cold	other_weather	direct_report
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

[5 rows x 36 columns]

```
In [14]: categories[categories.related==2]
```

Out[14]:

	related	request	offer	aid_related	medical_help	medical_products	\
117	2	0	0	0	0	0	
219	2	0	0	0	0	0	
305	2	0	0	0	0	0	
460	2	0	0	0	0	0	

576	2	0	0	0	0	0
655	2	0	0	0	0	0
656	2	0	0	0	0	0
883	2	0	0	0	0	0
897	2	0	0	0	0	0
925	2	0	0	0	0	0
931	2	0	0	0	0	0
933	2	0	0	0	0	0
1228	2	0	0	0	0	0
1250	2	0	0	0	0	0
1311	2	0	0	0	0	0
1403	2	0	0	0	0	0
1498	2	0	0	0	0	0
1686	2	0	0	0	0	0
1780	2	0	0	0	0	0
2344	2	0	0	0	0	0
2469	2	0	0	0	0	0
2534	2	0	0	0	0	0
2550	2	0	0	0	0	0
3019	2	0	0	0	0	0
3124	2	0	0	0	0	0
3360	2	0	0	0	0	0
3614	2	0	0	0	0	0
4587	2	0	0	0	0	0
4629	2	0	0	0	0	0
4630	2	0	0	0	0	0
...	...	...	...	...	...	...
12318	2	0	0	0	0	0
12322	2	0	0	0	0	0
12323	2	0	0	0	0	0
12324	2	0	0	0	0	0
12333	2	0	0	0	0	0
12342	2	0	0	0	0	0
12344	2	0	0	0	0	0
12351	2	0	0	0	0	0
12353	2	0	0	0	0	0
12405	2	0	0	0	0	0
12611	2	0	0	0	0	0
12684	2	0	0	0	0	0
13648	2	0	0	0	0	0
14848	2	0	0	0	0	0
14937	2	0	0	0	0	0
15515	2	0	0	0	0	0
15837	2	0	0	0	0	0
15883	2	0	0	0	0	0
15950	2	0	0	0	0	0
16821	2	0	0	0	0	0
17424	2	0	0	0	0	0

18275	2	0	0	0	0	0
18538	2	0	0	0	0	0
19739	2	0	0	0	0	0
20078	2	0	0	0	0	0
20351	2	0	0	0	0	0
20522	2	0	0	0	0	0
22355	2	0	0	0	0	0
23411	2	0	0	0	0	0
25247	2	0	0	0	0	0

	search_and_rescue	security	military	child_alone	...	\
117	0	0	0	0	...	
219	0	0	0	0	...	
305	0	0	0	0	...	
460	0	0	0	0	...	
576	0	0	0	0	...	
655	0	0	0	0	...	
656	0	0	0	0	...	
883	0	0	0	0	...	
897	0	0	0	0	...	
925	0	0	0	0	...	
931	0	0	0	0	...	
933	0	0	0	0	...	
1228	0	0	0	0	...	
1250	0	0	0	0	...	
1311	0	0	0	0	...	
1403	0	0	0	0	...	
1498	0	0	0	0	...	
1686	0	0	0	0	...	
1780	0	0	0	0	...	
2344	0	0	0	0	...	
2469	0	0	0	0	...	
2534	0	0	0	0	...	
2550	0	0	0	0	...	
3019	0	0	0	0	...	
3124	0	0	0	0	...	
3360	0	0	0	0	...	
3614	0	0	0	0	...	
4587	0	0	0	0	...	
4629	0	0	0	0	...	
4630	0	0	0	0	...	
...	...	...	...	...	...	
12318	0	0	0	0	...	
12322	0	0	0	0	...	
12323	0	0	0	0	...	
12324	0	0	0	0	...	
12333	0	0	0	0	...	
12342	0	0	0	0	...	

12344	0	0	0	0	...
12351	0	0	0	0	...
12353	0	0	0	0	...
12405	0	0	0	0	...
12611	0	0	0	0	...
12684	0	0	0	0	...
13648	0	0	0	0	...
14848	0	0	0	0	...
14937	0	0	0	0	...
15515	0	0	0	0	...
15837	0	0	0	0	...
15883	0	0	0	0	...
15950	0	0	0	0	...
16821	0	0	0	0	...
17424	0	0	0	0	...
18275	0	0	0	0	...
18538	0	0	0	0	...
19739	0	0	0	0	...
20078	0	0	0	0	...
20351	0	0	0	0	...
20522	0	0	0	0	...
22355	0	0	0	0	...
23411	0	0	0	0	...
25247	0	0	0	0	...

	aid_centers	other_infrastructure	weather_related	floods	storm	\
117	0	0	0	0	0	
219	0	0	0	0	0	
305	0	0	0	0	0	
460	0	0	0	0	0	
576	0	0	0	0	0	
655	0	0	0	0	0	
656	0	0	0	0	0	
883	0	0	0	0	0	
897	0	0	0	0	0	
925	0	0	0	0	0	
931	0	0	0	0	0	
933	0	0	0	0	0	
1228	0	0	0	0	0	
1250	0	0	0	0	0	
1311	0	0	0	0	0	
1403	0	0	0	0	0	
1498	0	0	0	0	0	
1686	0	0	0	0	0	
1780	0	0	0	0	0	
2344	0	0	0	0	0	
2469	0	0	0	0	0	
2534	0	0	0	0	0	

2550	0	0	0	0	0
3019	0	0	0	0	0
3124	0	0	0	0	0
3360	0	0	0	0	0
3614	0	0	0	0	0
4587	0	0	0	0	0
4629	0	0	0	0	0
4630	0	0	0	0	0
...	...	...	...	...	...
12318	0	0	0	0	0
12322	0	0	0	0	0
12323	0	0	0	0	0
12324	0	0	0	0	0
12333	0	0	0	0	0
12342	0	0	0	0	0
12344	0	0	0	0	0
12351	0	0	0	0	0
12353	0	0	0	0	0
12405	0	0	0	0	0
12611	0	0	0	0	0
12684	0	0	0	0	0
13648	0	0	0	0	0
14848	0	0	0	0	0
14937	0	0	0	0	0
15515	0	0	0	0	0
15837	0	0	0	0	0
15883	0	0	0	0	0
15950	0	0	0	0	0
16821	0	0	0	0	0
17424	0	0	0	0	0
18275	0	0	0	0	0
18538	0	0	0	0	0
19739	0	0	0	0	0
20078	0	0	0	0	0
20351	0	0	0	0	0
20522	0	0	0	0	0
22355	0	0	0	0	0
23411	0	0	0	0	0
25247	0	0	0	0	0

	fire	earthquake	cold	other_weather	direct_report
117	0	0	0	0	0
219	0	0	0	0	0
305	0	0	0	0	0
460	0	0	0	0	0
576	0	0	0	0	0
655	0	0	0	0	0
656	0	0	0	0	0

883	0	0	0	0	0
897	0	0	0	0	0
925	0	0	0	0	0
931	0	0	0	0	0
933	0	0	0	0	0
1228	0	0	0	0	0
1250	0	0	0	0	0
1311	0	0	0	0	0
1403	0	0	0	0	0
1498	0	0	0	0	0
1686	0	0	0	0	0
1780	0	0	0	0	0
2344	0	0	0	0	0
2469	0	0	0	0	0
2534	0	0	0	0	0
2550	0	0	0	0	0
3019	0	0	0	0	0
3124	0	0	0	0	0
3360	0	0	0	0	0
3614	0	0	0	0	0
4587	0	0	0	0	0
4629	0	0	0	0	0
4630	0	0	0	0	0
...	...	...	...	...	...
12318	0	0	0	0	0
12322	0	0	0	0	0
12323	0	0	0	0	0
12324	0	0	0	0	0
12333	0	0	0	0	0
12342	0	0	0	0	0
12344	0	0	0	0	0
12351	0	0	0	0	0
12353	0	0	0	0	0
12405	0	0	0	0	0
12611	0	0	0	0	0
12684	0	0	0	0	0
13648	0	0	0	0	0
14848	0	0	0	0	0
14937	0	0	0	0	0
15515	0	0	0	0	0
15837	0	0	0	0	0
15883	0	0	0	0	0
15950	0	0	0	0	0
16821	0	0	0	0	0
17424	0	0	0	0	0
18275	0	0	0	0	0
18538	0	0	0	0	0
19739	0	0	0	0	0

```
20078     0           0     0              0              0
20351     0           0     0              0              0
20522     0           0     0              0              0
22355     0           0     0              0              0
23411     0           0     0              0              0
25247     0           0     0              0              0

[193 rows x 36 columns]
```

1.0.4 5. Replace `categories` column in `df` with new category columns.

- Drop the categories column from the df dataframe since it is no longer needed.
- Concatenate df and categories data frames.

```
In [15]: # drop the original categories column from `df`
         df.drop('categories', axis=1, inplace=True)
         df.head()
```

```
Out[15]:    id                                            message  \
         0   2  Weather update - a cold front from Cuba that c...
         1   7            Is the Hurricane over or is it not over
         2   8                     Looking for someone but no name
         3   9  UN reports Leogane 80-90 destroyed. Only Hospi...
         4  12  says: west side of Haiti, rest of the country ...

                                              original   genre
         0  Un front froid se retrouve sur Cuba ce matin. ...  direct
         1                Cyclone nan fini osinon li pa fini  direct
         2  Patnm, di Maryani relem pou li banm nouvel li ...  direct
         3  UN reports Leogane 80-90 destroyed. Only Hospi...  direct
         4  facade ouest d Haiti et le reste du pays aujou...  direct
```

```
In [16]: # concatenate the original dataframe with the new `categories` dataframe
         df = pd.concat([df, categories], axis=1)
         df.head()
```

```
Out[16]:    id                                            message  \
         0   2  Weather update - a cold front from Cuba that c...
         1   7            Is the Hurricane over or is it not over
         2   8                     Looking for someone but no name
         3   9  UN reports Leogane 80-90 destroyed. Only Hospi...
         4  12  says: west side of Haiti, rest of the country ...

                                              original   genre  related  \
         0  Un front froid se retrouve sur Cuba ce matin. ...  direct      1.0
         1                Cyclone nan fini osinon li pa fini  direct      1.0
         2  Patnm, di Maryani relem pou li banm nouvel li ...  direct      1.0
         3  UN reports Leogane 80-90 destroyed. Only Hospi...  direct      1.0
         4  facade ouest d Haiti et le reste du pays aujou...  direct      1.0
```

```
   request  offer  aid_related  medical_help  medical_products      ...        \
0      0.0    0.0          0.0           0.0               0.0      ...
1      0.0    0.0          1.0           0.0               0.0      ...
2      0.0    0.0          0.0           0.0               0.0      ...
3      1.0    0.0          1.0           0.0               1.0      ...
4      0.0    0.0          0.0           0.0               0.0      ...

   aid_centers  other_infrastructure  weather_related  floods  storm  fire  \
0          0.0                   0.0              0.0     0.0    0.0   0.0
1          0.0                   0.0              1.0     0.0    1.0   0.0
2          0.0                   0.0              0.0     0.0    0.0   0.0
3          0.0                   0.0              0.0     0.0    0.0   0.0
4          0.0                   0.0              0.0     0.0    0.0   0.0

   earthquake  cold  other_weather  direct_report
0         0.0   0.0            0.0            0.0
1         0.0   0.0            0.0            0.0
2         0.0   0.0            0.0            0.0
3         0.0   0.0            0.0            0.0
4         0.0   0.0            0.0            0.0

[5 rows x 40 columns]
```

1.0.5 6. Remove duplicates.

- Check how many duplicates are in this dataset.
- Drop the duplicates.
- Confirm duplicates were removed.

In [17]:
```
# check number of duplicates
df[df.duplicated(subset='message')].count()
```

Out[17]:
```
id                      209
message                 209
original                 93
genre                   209
related                 209
request                 209
offer                   209
aid_related             209
medical_help            209
medical_products        209
search_and_rescue       209
security                209
military                209
child_alone             209
water                   209
```

```
food                      209
shelter                   209
clothing                  209
money                     209
missing_people            209
refugees                  209
death                     209
other_aid                 209
infrastructure_related    209
transport                 209
buildings                 209
electricity               209
tools                     209
hospitals                 209
shops                     209
aid_centers               209
other_infrastructure      209
weather_related           209
floods                    209
storm                     209
fire                      209
earthquake                209
cold                      209
other_weather             209
direct_report             209
dtype: int64
```

```
In [18]: # drop duplicates
         df.drop_duplicates(subset='message', inplace=True)
```

```
In [19]: # check number of duplicates
         df[df.duplicated(subset='message')].count()
```

```
Out[19]: id                      0
         message                 0
         original                0
         genre                   0
         related                 0
         request                 0
         offer                   0
         aid_related             0
         medical_help            0
         medical_products        0
         search_and_rescue       0
         security                0
         military                0
         child_alone             0
         water                   0
```

```
food                    0
shelter                 0
clothing                0
money                   0
missing_people          0
refugees                0
death                   0
other_aid               0
infrastructure_related  0
transport               0
buildings               0
electricity             0
tools                   0
hospitals               0
shops                   0
aid_centers             0
other_infrastructure    0
weather_related         0
floods                  0
storm                   0
fire                    0
earthquake              0
cold                    0
other_weather           0
direct_report           0
dtype: int64
```

```
In [18]: df.dropna(subset=category_colnames, inplace=True)
```

```
In [22]: # after reading here found that 2 = no, so replace all values of 2 with 0
         df.related.replace(2, 0, inplace=True)
```

1.0.6 7. Save the clean dataset into an sqlite database.

You can do this with pandas `to_sql` method combined with the SQLAlchemy library. Remember to import SQLAlchemy's `create_engine` in the first cell of this notebook to use it below.

```
In [23]: engine = create_engine('sqlite:///DisasterResponse.db')
         df.to_sql('labeled_data_messages', engine, index=False, if_exists='replace')
```

```
In [24]: engine.dispose()
```

1.0.7 8. Use this notebook to complete `etl_pipeline.py`

Use the template file attached in the Resources folder to write a script that runs the steps above to create a database based on new datasets specified by the user. Alternatively, you can complete `etl_pipeline.py` in the classroom on the `Project Workspace IDE` coming later.

```
In [ ]:
```