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 sqlite3
import pandas as pd
import numpy as np
import sqlite3
from sqlalchemy import create_engine
```

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]:

```
# load categories dataset
categories = pd.read_csv ('categories.csv')
categories.head()
```

Out[3]:

	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...

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 [4]:

```
# merge datasets
df = messages.merge (categories, left_on = 'id', right_on = 'id', how = 'inner', valida
te = 'many_to_many')

df.head()
```

Out[4]:

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

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 (https://pandas.pydata.org/pandas-docs/version/0.23/generated/pandas.Series.str.split.html)](https://pandas.pydata.org/pandas-docs/version/0.23/generated/pandas.Series.str.split.html) 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 [5]:

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

Out[5]:

	0	1	2	3	4	5	6	
0	related-1	request-0	offer-0	aid_related-0	medical_help-0	medical_products-0	search_and_rescue-0	se
1	related-1	request-0	offer-0	aid_related-1	medical_help-0	medical_products-0	search_and_rescue-0	se
2	related-1	request-0	offer-0	aid_related-0	medical_help-0	medical_products-0	search_and_rescue-0	se
3	related-1	request-1	offer-0	aid_related-1	medical_help-0	medical_products-1	search_and_rescue-0	se
4	related-1	request-0	offer-0	aid_related-0	medical_help-0	medical_products-0	search_and_rescue-0	se

5 rows × 36 columns

In [6]:

```
# select the first row of the categories dataframe
row = categories.iloc [0]

# 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.apply (lambda x: x.rstrip ('- 0 1'))
print(category_colnames)
```

```
0                    related
1                    request
2                      offer
3                aid_related
4               medical_help
5           medical_products
6          search_and_rescue
7                   security
8                   military
9                child_alone
10                    water
11                     food
12                  shelter
13                 clothing
14                    money
15            missing_people
16                  refugees
17                     death
18                other_aid
19    infrastructure_related
20                transport
21                buildings
22              electricity
23                    tools
24                hospitals
25                    shops
26              aid_centers
27      other_infrastructure
28           weather_related
29                    floods
30                     storm
31                      fire
32                earthquake
33                      cold
34             other_weather
35             direct_report
Name: 0, dtype: object
```

In [7]:

```
# rename the columns of `categories`
categories.columns = category_colnames
categories.head()
```

Out[7]:

	related	request	offer	aid_related	medical_help	medical_products	search_and_rescue	s
0	related-1	request-0	offer-0	aid_related-0	medical_help-0	medical_products-0	search_and_rescue-0	s
1	related-1	request-0	offer-0	aid_related-1	medical_help-0	medical_products-0	search_and_rescue-0	s
2	related-1	request-0	offer-0	aid_related-0	medical_help-0	medical_products-0	search_and_rescue-0	s
3	related-1	request-1	offer-0	aid_related-1	medical_help-0	medical_products-1	search_and_rescue-0	s
4	related-1	request-0	offer-0	aid_related-0	medical_help-0	medical_products-0	search_and_rescue-0	s

5 rows × 36 columns

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 (https://pandas.pydata.org/pandas-docs/stable/text.html#indexing-with-str)](https://pandas.pydata.org/pandas-docs/stable/text.html#indexing-with-str), 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 [8]:

```
for column in categories:
    # set each value to be the last character of the string
    categories[column] = categories[column].str [-1]

    # convert column from string to numeric
    categories[column] = pd.to_numeric(categories[column], errors = 'coerce')
categories.head()
```

Out[8]:

	related	request	offer	aid_related	medical_help	medical_products	search_and_rescue	se
0	1	0	0	0	0	0	0	
1	1	0	0	1	0	0	0	
2	1	0	0	0	0	0	0	
3	1	1	0	1	0	1	0	
4	1	0	0	0	0	0	0	

5 rows × 36 columns

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 [9]:

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

df.head()
```

Out[9]:

	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 [10]:

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

Out[10]:

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

5 rows × 40 columns

6. Remove duplicates.

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

In [11]:

```
# check number of duplicates
df.isnull().mean()
```

Out[11]:

```
id                          0.000000
message                     0.000000
original                    0.611688
genre                       0.000000
related                     0.000000
request                     0.000000
offer                       0.000000
aid_related                 0.000000
medical_help                0.000000
medical_products            0.000000
search_and_rescue           0.000000
security                    0.000000
military                    0.000000
child_alone                 0.000000
water                       0.000000
food                        0.000000
shelter                     0.000000
clothing                    0.000000
money                       0.000000
missing_people              0.000000
refugees                    0.000000
death                       0.000000
other_aid                   0.000000
infrastructure_related      0.000000
transport                   0.000000
buildings                   0.000000
electricity                 0.000000
tools                       0.000000
hospitals                   0.000000
shops                       0.000000
aid_centers                 0.000000
other_infrastructure        0.000000
weather_related             0.000000
floods                      0.000000
storm                       0.000000
fire                        0.000000
earthquake                  0.000000
cold                        0.000000
other_weather               0.000000
direct_report               0.000000
dtype: float64
```

In [12]:

```
# drop duplicates
df2 = df.drop_duplicates (subset = ['message'])
df2.shape
```

Out[12]:

```
(26177, 40)
```

In [13]:

```
# check number of duplicates
display(df2.nunique ())
df2.shape
```

```
id                        26177
message                   26177
original                   9630
genre                         3
related                       3
request                       2
offer                         2
aid_related                   2
medical_help                  2
medical_products              2
search_and_rescue             2
security                      2
military                      2
child_alone                   1
water                         2
food                          2
shelter                       2
clothing                      2
money                         2
missing_people                2
refugees                      2
death                         2
other_aid                     2
infrastructure_related        2
transport                     2
buildings                     2
electricity                   2
tools                         2
hospitals                     2
shops                         2
aid_centers                   2
other_infrastructure          2
weather_related               2
floods                        2
storm                         2
fire                          2
earthquake                    2
cold                          2
other_weather                 2
direct_report                 2
dtype: int64
```

Out[13]:

```
(26177, 40)
```

We observed that genre, related and child_alone columns have 3,3,1 categories respectively. analysis should be done more

In [14]:

```
df2['related'].unique()
```

Out[14]:

```
array([1, 0, 2])
```

In [15]:

```
df2['child_alone'].unique()
```

Out[15]:

```
array([0])
```

In [16]:

```
df2['genre'].unique()
```

Out[16]:

```
array(['direct', 'social', 'news'], dtype=object)
```

In [17]:

```
df2 [df2 ['related'] ==2].describe ()
```

Out[17]:

	id	related	request	offer	aid_related	medical_help	medical_products	sea
count	188.000000	188.0	188.0	188.0	188.0	188.0	188.0	
mean	11703.340426	2.0	0.0	0.0	0.0	0.0	0.0	
std	5479.507080	0.0	0.0	0.0	0.0	0.0	0.0	
min	146.000000	2.0	0.0	0.0	0.0	0.0	0.0	
25%	8956.000000	2.0	0.0	0.0	0.0	0.0	0.0	
50%	13770.000000	2.0	0.0	0.0	0.0	0.0	0.0	
75%	14376.750000	2.0	0.0	0.0	0.0	0.0	0.0	
max	29126.000000	2.0	0.0	0.0	0.0	0.0	0.0	

8 rows × 37 columns

In [18]:

```
df2 [df2 ['related'] ==1].describe ()
```

Out[18]:

	id	related	request	offer	aid_related	medical_help	medic
count	19874.000000	19874.0	19874.000000	19874.000000	19874.000000	19874.000000	1
mean	15691.340495	1.0	0.224615	0.005887	0.545436	0.104659	
std	8794.219871	0.0	0.417339	0.076503	0.497944	0.306122	
min	2.000000	1.0	0.000000	0.000000	0.000000	0.000000	
25%	7896.250000	1.0	0.000000	0.000000	0.000000	0.000000	
50%	16596.500000	1.0	0.000000	0.000000	1.000000	0.000000	
75%	23058.750000	1.0	0.000000	0.000000	1.000000	0.000000	
max	30265.000000	1.0	1.000000	1.000000	1.000000	1.000000	

8 rows × 37 columns

In [19]:

```
df2 [df2 ['related'] ==0].describe()
```

Out[19]:

	id	related	request	offer	aid_related	medical_help	medical_products	se
count	6115.000000	6115.0	6115.0	6115.0	6115.0	6115.0	6115.0	
mean	13823.403271	0.0	0.0	0.0	0.0	0.0	0.0	
std	8844.978443	0.0	0.0	0.0	0.0	0.0	0.0	
min	14.000000	0.0	0.0	0.0	0.0	0.0	0.0	
25%	6751.000000	0.0	0.0	0.0	0.0	0.0	0.0	
50%	10470.000000	0.0	0.0	0.0	0.0	0.0	0.0	
75%	22591.000000	0.0	0.0	0.0	0.0	0.0	0.0	
max	30262.000000	0.0	0.0	0.0	0.0	0.0	0.0	

8 rows × 37 columns

In [20]:

```
df2 [df2 ['related'] ==2].head(n= 5)
```

Out[20]:

	id	message	original	genre	related	request	offer	aid_related	medical
117	146	Dans la zone de Saint Etienne la route de Jacm...	Nan zon st. etine rout jakmel la bloke se mize...	direct	2	0	0	0	
221	263	. .. i with limited means. Certain patients co...	t avec des moyens limites. Certains patients v...	direct	2	0	0	0	
307	373	The internet caf (/cdn-cgi/l/email-protection) that's by the Dal road...	Cyber cafe (/cdn-cgi/l/email-protection) ki chita rout de dal tou pr ...	direct	2	0	0	0	
462	565	Bonsoir, on est a bon repos aprs la compagnie ...	Bonswa nou nan bon repo apri teleko nan wout t...	direct	2	0	0	0	
578	700	URGENT CRECHE ORPHANAGE KAY TOUT TIMOUN CROIX ...	r et Salon Furterer. mwen se yon Cosmtologue. ...	direct	2	0	0	0	

5 rows × 40 columns

In [21]:

```
#dropping 'child_alone'
df3 = df2.drop('child_alone', axis = 1)

#slicing so to get a dataframe that contains only related !=2
df4 = df3[df3['related'] !=2 ]
```

In [22]:

```
df4.nunique()
```

Out[22]:

```
id                        25989
message                   25989
original                   9507
genre                         3
related                       2
request                       2
offer                         2
aid_related                   2
medical_help                  2
medical_products              2
search_and_rescue             2
security                      2
military                      2
water                         2
food                          2
shelter                       2
clothing                      2
money                         2
missing_people                2
refugees                      2
death                         2
other_aid                     2
infrastructure_related        2
transport                     2
buildings                     2
electricity                   2
tools                         2
hospitals                     2
shops                         2
aid_centers                   2
other_infrastructure          2
weather_related               2
floods                        2
storm                         2
fire                          2
earthquake                    2
cold                          2
other_weather                 2
direct_report                 2
dtype: int64
```

In [23]:

```
df4.related.unique()
```

Out[23]:

```
array([1, 0])
```

7. Save the clean dataset into an sqlite database.

You can do this with pandas `to_sql` method (https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.to_sql.html) combined with the SQLAlchemy library. Remember to import SQLAlchemy's `create_engine` in the first cell of this notebook to use it below.

In [24]:

```
engine = create_engine('sqlite:///DisasterResponse.db')
df.to_sql('MessagesCategories', engine, index=False)
```

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 [ ]:

```

```