ML Pipeline Preparation

July 14, 2020

1 ML Pipeline Preparation

Follow the instructions below to help you create your ML pipeline. ### 1. Import libraries and load data from database. - Import Python libraries - Load dataset from database with `read_sql_table` - Define feature and target variables X and Y

```
In [1]: # import libraries
        import nltk
        nltk.download('punkt')
        nltk.download('wordnet')
        nltk.download('averaged_perceptron_tagger')

        import plotly

        import numpy as np
        import pandas as pd
        pd.set_option('display.max_columns', 500)

        import sys
        import os
        import re
        from sqlalchemy import create_engine
        import pickle

        from scipy.stats import gmean
        from sklearn.pipeline import Pipeline, FeatureUnion
        from sklearn.model_selection import train_test_split, GridSearchCV
        from sklearn.metrics import classification_report, confusion_matrix, fbeta_score, make_s
        from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier, AdaBoos
        from sklearn.feature_extraction.text import TfidfTransformer, CountVectorizer
        from sklearn.multioutput import MultiOutputClassifier
        from sklearn.base import BaseEstimator,TransformerMixin
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
```

In [2]:
```
# load data from database
database_filepath = "./disaster_response_db.db"
engine = create_engine('sqlite:///' + database_filepath)
table_name = os.path.basename(database_filepath).replace(".db","") + "_table"
df = pd.read_sql_table(table_name,engine)
```

In [3]:
```
# 'child_alone' = Unary column = all Zero values
# 'related' should be 0,1 but contains 2
df.describe()
```

Out[3]:

	id	related	request	offer	aid_related
count	26216.00000	26216.000000	26216.000000	26216.000000	26216.000000
mean	15224.82133	0.773650	0.170659	0.004501	0.414251
std	8826.88914	0.435276	0.376218	0.066940	0.492602
min	2.00000	0.000000	0.000000	0.000000	0.000000
25%	7446.75000	1.000000	0.000000	0.000000	0.000000
50%	15662.50000	1.000000	0.000000	0.000000	0.000000
75%	22924.25000	1.000000	0.000000	0.000000	1.000000
max	30265.00000	2.000000	1.000000	1.000000	1.000000

	medical_help	medical_products	search_and_rescue	security
count	26216.000000	26216.000000	26216.000000	26216.000000
mean	0.079493	0.050084	0.027617	0.017966
std	0.270513	0.218122	0.163875	0.132831
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000

	military	child_alone	water	food	shelter
count	26216.000000	26216.0	26216.000000	26216.000000	26216.000000
mean	0.032804	0.0	0.063778	0.111497	0.088267
std	0.178128	0.0	0.244361	0.314752	0.283688
min	0.000000	0.0	0.000000	0.000000	0.000000
25%	0.000000	0.0	0.000000	0.000000	0.000000
50%	0.000000	0.0	0.000000	0.000000	0.000000
75%	0.000000	0.0	0.000000	0.000000	0.000000
max	1.000000	0.0	1.000000	1.000000	1.000000

	clothing	money	missing_people	refugees	death
count	26216.000000	26216.000000	26216.000000	26216.000000	26216.000000
mean	0.015449	0.023039	0.011367	0.033377	0.045545

```
std         0.123331      0.150031         0.106011      0.179621      0.208500
min         0.000000      0.000000         0.000000      0.000000      0.000000
25%         0.000000      0.000000         0.000000      0.000000      0.000000
50%         0.000000      0.000000         0.000000      0.000000      0.000000
75%         0.000000      0.000000         0.000000      0.000000      0.000000
max         1.000000      1.000000         1.000000      1.000000      1.000000

          other_aid  infrastructure_related     transport     buildings  \
count  26216.000000            26216.000000  26216.000000  26216.000000
mean       0.131446                0.065037      0.045812      0.050847
std        0.337894                0.246595      0.209081      0.219689
min        0.000000                0.000000      0.000000      0.000000
25%        0.000000                0.000000      0.000000      0.000000
50%        0.000000                0.000000      0.000000      0.000000
75%        0.000000                0.000000      0.000000      0.000000
max        1.000000                1.000000      1.000000      1.000000

        electricity         tools     hospitals         shops   aid_centers  \
count  26216.000000  26216.000000  26216.000000  26216.000000  26216.000000
mean       0.020293      0.006065      0.010795      0.004577      0.011787
std        0.141003      0.077643      0.103338      0.067502      0.107927
min        0.000000      0.000000      0.000000      0.000000      0.000000
25%        0.000000      0.000000      0.000000      0.000000      0.000000
50%        0.000000      0.000000      0.000000      0.000000      0.000000
75%        0.000000      0.000000      0.000000      0.000000      0.000000
max        1.000000      1.000000      1.000000      1.000000      1.000000

       other_infrastructure  weather_related        floods         storm  \
count          26216.000000     26216.000000  26216.000000  26216.000000
mean               0.043904         0.278341      0.082202      0.093187
std                0.204887         0.448191      0.274677      0.290700
min                0.000000         0.000000      0.000000      0.000000
25%                0.000000         0.000000      0.000000      0.000000
50%                0.000000         0.000000      0.000000      0.000000
75%                0.000000         1.000000      0.000000      0.000000
max                1.000000         1.000000      1.000000      1.000000

               fire    earthquake          cold  other_weather  direct_report
count  26216.000000  26216.000000  26216.000000   26216.000000   26216.000000
mean       0.010757      0.093645      0.020217       0.052487       0.193584
std        0.103158      0.291340      0.140743       0.223011       0.395114
min        0.000000      0.000000      0.000000       0.000000       0.000000
25%        0.000000      0.000000      0.000000       0.000000       0.000000
50%        0.000000      0.000000      0.000000       0.000000       0.000000
75%        0.000000      0.000000      0.000000       0.000000       0.000000
max        1.000000      1.000000      1.000000       1.000000       1.000000
```

In [4]:
```
# 'related' = 2 = 188 rows = assume to be anomaly and remove all
```

```
df.groupby("related").count()
```

Out[4]:
```
            id  message  original  genre  request  offer  aid_related  \
related
0         6122     6122      3395   6122     6122   6122         6122
1        19906    19906      6643  19906    19906  19906        19906
2          188      188       132    188      188    188          188

         medical_help  medical_products  search_and_rescue  security  \
related
0                6122              6122               6122      6122
1               19906             19906              19906     19906
2                 188               188                188       188

         military  child_alone  water   food  shelter  clothing  money  \
related
0            6122         6122   6122   6122     6122      6122   6122
1           19906        19906  19906  19906    19906     19906  19906
2             188          188    188    188      188       188    188

         missing_people  refugees  death  other_aid  infrastructure_related  \
related
0                  6122      6122   6122       6122                    6122
1                 19906     19906  19906      19906                   19906
2                   188       188    188        188                     188

         transport  buildings  electricity  tools  hospitals  shops  \
related
0             6122       6122         6122   6122       6122   6122
1            19906      19906        19906  19906      19906  19906
2              188        188          188    188        188    188

         aid_centers  other_infrastructure  weather_related  floods  storm  \
related
0               6122                  6122             6122    6122   6122
1              19906                 19906            19906   19906  19906
2                188                   188              188     188    188

          fire  earthquake   cold  other_weather  direct_report
related
0         6122        6122   6122           6122           6122
1        19906       19906  19906          19906          19906
2          188         188    188            188            188
```

In [5]:
```
# Dropped all 188 rows
df = df[df.related != 2]
df.groupby("related").count()
```

Out[5]:
```
            id  message  original  genre  request  offer  aid_related  \
```

```
related
0          6122     6122      3395   6122     6122   6122          6122
1         19906    19906      6643  19906    19906  19906         19906

         medical_help  medical_products  search_and_rescue  security  \
related
0                6122              6122               6122      6122
1               19906             19906              19906     19906

         military  child_alone  water   food  shelter  clothing  money  \
related
0            6122         6122   6122   6122     6122      6122   6122
1           19906        19906  19906  19906    19906     19906  19906

         missing_people  refugees  death  other_aid  infrastructure_related  \
related
0                  6122      6122   6122       6122                    6122
1                 19906     19906  19906      19906                   19906

         transport  buildings  electricity  tools  hospitals  shops  \
related
0             6122       6122         6122   6122       6122   6122
1            19906      19906        19906  19906      19906  19906

         aid_centers  other_infrastructure  weather_related  floods  storm  \
related
0               6122                  6122             6122    6122   6122
1              19906                 19906            19906   19906  19906

          fire  earthquake   cold  other_weather  direct_report
related
0         6122        6122   6122           6122           6122
1        19906       19906  19906          19906          19906
```

In [6]:
```python
# Extract X and y variables from the data for the modelling
X = df['message']
y = df.iloc[:,4:]
```

1.0.1 2. Write a tokenization function to process your text data

In [7]:
```python
def tokenize(text,url_place_holder_string="urlplaceholder"):

    # Replace urls with url placeholder string
    url_regex = 'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-f

    # Extract urls from the provided text
    detected_urls = re.findall(url_regex, text)
```

```
        # Replace url with url placeholder string
        for detected_url in detected_urls:
            text = text.replace(detected_url, url_place_holder_string)

        # Extract the word tokens from the provided text
        tokens = nltk.word_tokenize(text)

        #Lemmanitizer to remove inflectional and derivationally related forms of a word
        lemmatizer = nltk.WordNetLemmatizer()

        # List of clean tokens
        clean_tokens = [lemmatizer.lemmatize(w).lower().strip() for w in tokens]

        return clean_tokens
```

```
In [8]: # Custom transformer to extract the starting verb of sentence
        class StartingVerbExtractor(BaseEstimator, TransformerMixin):
            """
            Starting Verb Extractor class

            Extract first verb of sentence to convert it into new feature for the ML classifier
            """

            def starting_verb(self, text):
                sentence_list = nltk.sent_tokenize(text)
                for sentence in sentence_list:
                    pos_tags = nltk.pos_tag(tokenize(sentence))
                    first_word, first_tag = pos_tags[0]
                    if first_tag in ['VB', 'VBP'] or first_word == 'RT':
                        return True
                return False

            def fit(self, X, y=None):
                return self

            def transform(self, X):
                X_tagged = pd.Series(X).apply(self.starting_verb)
                return pd.DataFrame(X_tagged)
```

1.0.2 3. Build a machine learning pipeline

This machine pipeline should take in the `message` column as input and output classification results on the other 36 categories in the dataset. You may find the MultiOutputClassifier helpful for predicting multiple target variables.

```
In [9]: pipeline1 = Pipeline([
                ('features', FeatureUnion([
```

```
            ('text_pipeline', Pipeline([
                ('count_vectorizer', CountVectorizer(tokenizer=tokenize)),
                ('tfidf_transformer', TfidfTransformer())
            ]))
        ])),

        ('classifier', MultiOutputClassifier(AdaBoostClassifier()))
    ])

pipeline2 = Pipeline([
        ('features', FeatureUnion([

            ('text_pipeline', Pipeline([
                ('count_vectorizer', CountVectorizer(tokenizer=tokenize)),
                ('tfidf_transformer', TfidfTransformer())
            ])),

            ('starting_verb_transformer', StartingVerbExtractor())
        ])),

        ('classifier', MultiOutputClassifier(AdaBoostClassifier()))
    ])
```

1.0.3 4. Train pipeline

- Split data into train and test sets
- Train pipeline

In [10]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y)
pipeline_fitted = pipeline1.fit(X_train, y_train)
```

1.0.4 5. Test your model

Report the f1 score, precision and recall for each output category of the dataset. You can do this by iterating through the columns and calling sklearn's `classification_report` on each.

In [11]:
```
y_prediction_train = pipeline_fitted.predict(X_train)
y_prediction_test = pipeline_fitted.predict(X_test)

# Print classification report on test data
print(classification_report(y_test.values, y_prediction_test, target_names=y.columns.va
```

```
              precision    recall  f1-score   support

     related       0.83      0.94      0.88      4982
     request       0.75      0.53      0.62      1099
       offer       0.00      0.00      0.00        34
 aid_related       0.77      0.61      0.68      2688
medical_help       0.59      0.27      0.37       516
```

```
      medical_products       0.73      0.36      0.48       329
     search_and_rescue       0.50      0.17      0.25       172
              security       0.21      0.04      0.07       120
              military       0.58      0.36      0.44       219
           child_alone       0.00      0.00      0.00         0
                 water       0.76      0.67      0.71       381
                  food       0.80      0.68      0.73       715
               shelter       0.78      0.55      0.65       594
              clothing       0.77      0.46      0.58       102
                 money       0.55      0.36      0.44       163
        missing_people       0.67      0.25      0.36        64
              refugees       0.55      0.22      0.32       224
                 death       0.71      0.46      0.56       297
             other_aid       0.51      0.17      0.26       848
infrastructure_related       0.46      0.07      0.12       455
             transport       0.57      0.17      0.27       288
             buildings       0.64      0.39      0.48       308
           electricity       0.45      0.19      0.26       135
                 tools       0.00      0.00      0.00        43
             hospitals       0.33      0.07      0.12        86
                 shops       0.00      0.00      0.00        28
           aid_centers       0.32      0.07      0.11        86
  other_infrastructure       0.33      0.09      0.14       293
       weather_related       0.85      0.66      0.74      1805
                floods       0.88      0.58      0.70       567
                 storm       0.74      0.54      0.62       597
                  fire       0.53      0.32      0.40        60
            earthquake       0.89      0.78      0.83       602
                  cold       0.77      0.32      0.45       136
         other_weather       0.44      0.15      0.22       327
         direct_report       0.70      0.48      0.57      1233

           avg / total       0.73      0.59      0.63     20596


/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWa

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.

/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1137: UndefinedMetricWa

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

In [12]:
```
# Print classification report on training data
print('\n',classification_report(y_train.values, y_prediction_train, target_names=y.col
```

	precision	recall	f1-score	support
related	0.83	0.95	0.88	14924
request	0.80	0.56	0.66	3375
offer	0.40	0.10	0.15	84
aid_related	0.77	0.61	0.68	8172
medical_help	0.66	0.29	0.40	1568
medical_products	0.70	0.35	0.47	984
search_and_rescue	0.64	0.19	0.29	552
security	0.47	0.08	0.13	351
military	0.66	0.38	0.48	641
child_alone	0.00	0.00	0.00	0
water	0.78	0.66	0.72	1291
food	0.82	0.69	0.75	2208
shelter	0.80	0.57	0.67	1720
clothing	0.74	0.46	0.57	303
money	0.65	0.34	0.44	441
missing_people	0.75	0.19	0.31	234
refugees	0.67	0.27	0.39	651
death	0.78	0.47	0.59	897
other_aid	0.58	0.18	0.28	2598
infrastructure_related	0.50	0.09	0.15	1250
transport	0.77	0.27	0.40	913
buildings	0.71	0.42	0.53	1025
electricity	0.72	0.31	0.43	397
tools	0.62	0.09	0.15	116
hospitals	0.44	0.11	0.18	197
shops	0.53	0.09	0.15	92
aid_centers	0.43	0.09	0.14	223
other_infrastructure	0.43	0.11	0.18	858
weather_related	0.87	0.65	0.75	5492
floods	0.88	0.55	0.68	1588
storm	0.76	0.56	0.65	1846
fire	0.74	0.33	0.45	222
earthquake	0.90	0.78	0.83	1853
cold	0.76	0.34	0.47	394
other_weather	0.57	0.19	0.28	1049
direct_report	0.76	0.50	0.60	3842
avg / total	0.77	0.60	0.65	62351

```
/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWa

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1137: UndefinedMetricWa

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

1.0.5 6. Improve your model

Use grid search to find better parameters.

```
In [13]: # pipeline1.get_params().keys()
         parameters_grid = {'classifier__estimator__learning_rate': [0.01, 0.02, 0.05],
                       'classifier__estimator__n_estimators': [10, 20, 40]}

         #parameters_grid = {'classifier__estimator__n_estimators': [10, 20, 40]}

         cv = GridSearchCV(pipeline1, param_grid=parameters_grid, scoring='f1_micro', n_jobs=-1)
         cv.fit(X_train, y_train)
```

```
Out[13]: GridSearchCV(cv=None, error_score='raise',
                estimator=Pipeline(memory=None,
              steps=[('features', FeatureUnion(n_jobs=1,
                transformer_list=[('text_pipeline', Pipeline(memory=None,
              steps=[('count_vectorizer', CountVectorizer(analyzer='word', binary=False, decode_
                 dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
                 lowercase...mator=None,
                   learning_rate=1.0, n_estimators=50, random_state=None),
                    n_jobs=1))]),
                fit_params=None, iid=True, n_jobs=-1,
                param_grid={'classifier__estimator__learning_rate': [0.01, 0.02, 0.05], 'classif
                pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
                scoring='f1_micro', verbose=0)
```

```
In [14]: # Get the prediction values from the grid search cross validator
         y_prediction_test = cv.predict(X_test)
         y_prediction_train = cv.predict(X_train)
```

1.0.6 7. Test your model

Show the accuracy, precision, and recall of the tuned model.

Since this project focuses on code quality, process, and pipelines, there is no minimum performance metric needed to pass. However, make sure to fine tune your models for accuracy, precision and recall to make your project stand out - especially for your portfolio!

```
In [15]: # Print classification report on test data
         print(classification_report(y_test.values, y_prediction_test, target_names=y.columns.va

                         precision    recall  f1-score   support
```

```
               related       0.77      1.00      0.87      4982
               request       0.58      0.38      0.46      1099
                 offer       0.00      0.00      0.00        34
           aid_related       0.81      0.19      0.31      2688
          medical_help       0.63      0.13      0.21       516
      medical_products       0.72      0.13      0.22       329
     search_and_rescue       0.60      0.19      0.29       172
              security       0.00      0.00      0.00       120
              military       0.51      0.15      0.23       219
           child_alone       0.00      0.00      0.00         0
                 water       0.56      0.82      0.67       381
                  food       0.77      0.68      0.72       715
               shelter       0.86      0.31      0.46       594
              clothing       0.74      0.33      0.46       102
                 money       0.50      0.12      0.20       163
        missing_people       0.67      0.31      0.43        64
              refugees       0.50      0.01      0.03       224
                 death       0.78      0.17      0.28       297
             other_aid       0.00      0.00      0.00       848
infrastructure_related       0.00      0.00      0.00       455
             transport       0.53      0.19      0.28       288
             buildings       0.00      0.00      0.00       308
           electricity       0.00      0.00      0.00       135
                 tools       0.00      0.00      0.00        43
             hospitals       0.00      0.00      0.00        86
                 shops       0.00      0.00      0.00        28
           aid_centers       0.00      0.00      0.00        86
  other_infrastructure       0.00      0.00      0.00       293
       weather_related       0.91      0.22      0.35      1805
                floods       0.94      0.36      0.52       567
                 storm       0.77      0.28      0.41       597
                  fire       0.44      0.40      0.42        60
            earthquake       0.90      0.65      0.75       602
                  cold       0.82      0.26      0.40       136
         other_weather       0.59      0.11      0.19       327
         direct_report       0.60      0.39      0.47      1233

           avg / total       0.66      0.44      0.47     20596
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWa

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.

/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1137: UndefinedMetricWa

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

```
In [16]: # Print classification report on training data
         print('\n',classification_report(y_train.values, y_prediction_train, target_names=y.col
```

```
                        precision    recall  f1-score   support

               related       0.76      1.00      0.87     14924
               request       0.62      0.38      0.47      3375
                 offer       1.00      0.02      0.05        84
           aid_related       0.80      0.19      0.31      8172
          medical_help       0.64      0.11      0.19      1568
      medical_products       0.73      0.13      0.21       984
     search_and_rescue       0.65      0.19      0.29       552
              security       0.00      0.00      0.00       351
              military       0.52      0.14      0.22       641
           child_alone       0.00      0.00      0.00         0
                 water       0.58      0.85      0.69      1291
                  food       0.78      0.68      0.72      2208
               shelter       0.84      0.29      0.43      1720
              clothing       0.75      0.31      0.44       303
                 money       0.57      0.17      0.27       441
        missing_people       0.68      0.24      0.36       234
              refugees       0.83      0.02      0.03       651
                 death       0.74      0.16      0.27       897
             other_aid       0.00      0.00      0.00      2598
infrastructure_related       0.00      0.00      0.00      1250
             transport       0.59      0.26      0.37       913
             buildings       0.00      0.00      0.00      1025
           electricity       0.00      0.00      0.00       397
                 tools       0.00      0.00      0.00       116
             hospitals       0.00      0.00      0.00       197
                 shops       0.00      0.00      0.00        92
           aid_centers       0.00      0.00      0.00       223
  other_infrastructure       0.00      0.00      0.00       858
       weather_related       0.92      0.22      0.36      5492
                floods       0.92      0.33      0.48      1588
                 storm       0.72      0.25      0.38      1846
                  fire       0.53      0.41      0.46       222
            earthquake       0.90      0.64      0.75      1853
                  cold       0.64      0.20      0.31       394
         other_weather       0.57      0.12      0.21      1049
         direct_report       0.64      0.39      0.48      3842

           avg / total       0.67      0.44      0.47     62351
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWa

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.

/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1137: UndefinedMetricWa

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

1.0.7 8. Try improving your model further. Here are a few ideas:

- try other machine learning algorithms
- add other features besides the TF-IDF

```
In [17]: #Use pipeline2 which includes StartingVerbEstimator
         X_train, X_test, y_train, y_test = train_test_split(X, y)
         pipeline_fitted = pipeline2.fit(X_train, y_train)

         y_prediction_train = pipeline_fitted.predict(X_train)
         y_prediction_test = pipeline_fitted.predict(X_test)

         # Print classification report on test data
         print(classification_report(y_test.values, y_prediction_test, target_names=y.columns.va
```

```
                        precision    recall  f1-score   support

               related       0.82      0.95      0.88      4938
               request       0.77      0.53      0.63      1119
                 offer       0.14      0.04      0.06        24
           aid_related       0.76      0.61      0.68      2682
          medical_help       0.54      0.25      0.35       502
      medical_products       0.60      0.33      0.43       311
     search_and_rescue       0.69      0.21      0.32       181
              security       0.29      0.06      0.09       109
              military       0.66      0.32      0.43       209
           child_alone       0.00      0.00      0.00         0
                 water       0.76      0.64      0.69       415
                  food       0.80      0.69      0.74       731
               shelter       0.77      0.53      0.63       576
              clothing       0.73      0.45      0.56        97
                 money       0.59      0.29      0.39       140
        missing_people       0.36      0.11      0.17        73
              refugees       0.63      0.24      0.35       219
                 death       0.75      0.43      0.55       291
             other_aid       0.54      0.14      0.22       855
infrastructure_related       0.42      0.10      0.16       427
             transport       0.60      0.19      0.29       293
```

```
            buildings       0.69      0.40      0.50       344
          electricity       0.55      0.21      0.30       150
                tools       0.50      0.03      0.06        34
            hospitals       0.33      0.07      0.11        74
                shops       0.17      0.04      0.06        26
          aid_centers       0.23      0.10      0.14        72
 other_infrastructure       0.34      0.08      0.13       294
      weather_related       0.87      0.64      0.74      1818
               floods       0.86      0.55      0.67       535
                storm       0.75      0.49      0.59       612
                 fire       0.76      0.33      0.46        80
           earthquake       0.89      0.75      0.82       627
                 cold       0.70      0.30      0.42       125
        other_weather       0.42      0.13      0.20       337
        direct_report       0.73      0.49      0.59      1273

          avg / total       0.74      0.58      0.63     20593
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWa

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.

/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1137: UndefinedMetricWa

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

In [18]:
```python
# Print classification report on training data
print('\n',classification_report(y_train.values, y_prediction_train, target_names=y.col
```

```
                      precision    recall  f1-score   support

              related       0.82      0.95      0.88     14968
              request       0.80      0.55      0.65      3355
                offer       0.40      0.09      0.14        94
          aid_related       0.77      0.61      0.68      8178
         medical_help       0.66      0.30      0.42      1582
     medical_products       0.71      0.34      0.46      1002
    search_and_rescue       0.66      0.22      0.33       543
             security       0.45      0.07      0.12       362
             military       0.66      0.38      0.48       651
          child_alone       0.00      0.00      0.00         0
                water       0.78      0.68      0.73      1257
                 food       0.81      0.70      0.75      2192
```

```
               shelter       0.81      0.57      0.66      1738
              clothing       0.78      0.48      0.59       308
                 money       0.62      0.31      0.41       464
        missing_people       0.78      0.22      0.34       225
              refugees       0.66      0.28      0.39       656
                 death       0.80      0.48      0.60       903
             other_aid       0.56      0.16      0.25      2591
infrastructure_related       0.54      0.11      0.18      1278
             transport       0.75      0.24      0.36       908
             buildings       0.72      0.43      0.53       989
           electricity       0.64      0.25      0.36       382
                 tools       0.21      0.03      0.06       125
             hospitals       0.45      0.15      0.23       209
                 shops       0.44      0.04      0.08        94
           aid_centers       0.49      0.13      0.20       237
  other_infrastructure       0.49      0.11      0.19       857
       weather_related       0.87      0.66      0.75      5479
                floods       0.87      0.58      0.70      1620
                 storm       0.77      0.52      0.62      1831
                  fire       0.62      0.31      0.41       202
            earthquake       0.89      0.78      0.83      1828
                  cold       0.75      0.38      0.51       405
         other_weather       0.59      0.22      0.32      1039
         direct_report       0.76      0.49      0.60      3802

           avg / total       0.76      0.60      0.65     62354


/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWa

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.

/opt/conda/lib/python3.6/site-packages/sklearn/metrics/classification.py:1137: UndefinedMetricWa

Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples.
```

1.0.8 9. Export your model as a pickle file

```
In [19]: m = pickle.dumps('classifier.pkl')
```

1.0.9 10. Use this notebook to complete `train.py`

Use the template file attached in the Resources folder to write a script that runs the steps above to create a database and export a model based on a new dataset specified by the user.

```
In [ ]:
```