**Project Components:**

There are three components of this project:

**1. ETL Pipeline**

File \_data/process\_data.py\_ contains data cleaning pipeline that:

- Loads the `messages` and `categories` dataset

- Merges the two datasets

- Cleans the data

- Stores it in a \*\*SQLite database\*\*

**2. ML Pipeline**

File \_models/train\_classifier.py\_ contains machine learning pipeline that:

- Loads data from the \*\*SQLite database\*\*

- Splits the data into training and testing sets

- Builds a text processing and machine learning pipeline

- Trains and tunes a model using GridSearchCV

- Outputs result on the test set

- Exports the final model as a pickle file

**3. Flask Web App**

File \_app/run\_app.py\_

Running this command will start the web app where users can enter their query, i.e., a request message sent during a natural disaster.

What the app will do is that it will classify the text message into categories so that appropriate relief agency can be reached out for help.

**Files:**

**app:**

run.py - FLASK FILE THAT RUNS APP

static - favicon.ico - FAVICON FOR THE WEB APP

templates - go.html - CLASSIFICATION RESULT PAGE OF WEB APP; master.html - MAIN PAGE OF WEB APP

**data:**

DisasterResponse.db - DATABASE TO SAVE CLEANED DATA TO

disaster\_categories.csv - DATA TO PROCESS

disaster\_messages.csv - DATA TO PROCESS

process\_data.py - PERFORMS ETL PROCESS

**screenshots:**

- images of various plots, main web app look, result and command prompt run output

**models:**

train\_classifier.py - PERFORMS CLASSIFICATION TASK

classifier.pkl - PICKLE FILE

**Software Requirements:**

Flask==1.0.2

nltk==3.4

numpy==1.15.4

pandas==0.22.0

plotly==3.4.2

scikit-learn==0.20.1

SQLAlchemy==1.2.14