Project Details

Below are additional details about each component and tips to get you started.

Data Pipelines: Jupyter Notebooks

We've provided Jupyter notebooks in Project Workspaces with instructions to get you started with both data pipelines. The Jupyter notebook is not required for submission, but highly recommended to complete before getting started on the Python script.

Project Workspace - ETL

The first part of your data pipeline is the Extract, Transform, and Load process. Here, you will read the dataset, clean the data, and then store it in a SQLite database. We expect you to do the data cleaning with pandas. To load the data into an SQLite database, you can use the pandas dataframe `.to_sql()` method, which you can use with an SQLAlchemy engine.

Feel free to do some exploratory data analysis in order to figure out how you want to clean the data set. Though you do not need to submit this exploratory data analysis as part of your project, you'll need to include your cleaning code in the final ETL script, `process_data.py`.

Project Workspace - Machine Learning Pipeline

For the machine learning portion, you will split the data into a training set and a test set. Then, you will create a machine learning pipeline that uses NLTK, as well as scikit-learn's Pipeline and GridSearchCV to output a final model that uses the `message` column to predict classifications for 36 categories (multi-output classification). Finally, you will export your model to a pickle file. After completing the notebook, you'll need to include your final machine learning code in `train_classifier.py`.

Data Pipelines: Python Scripts

After you complete the notebooks for the ETL and machine learning pipeline, you'll need to transfer your work into Python scripts, `process_data.py` and `train_classifier.py`. If someone in the future comes with a revised or new dataset of messages, they should be able to easily create a new model just by running your code. These Python scripts should be able to run with additional arguments specifying the files used for the data and model.

Example:

```
python process_data.py disaster_messages.csv disaster_categories.csv DisasterResponse.db

python train_classifier.py ../data/DisasterResponse.db classifier.pkl
```