

# Disaster Response Pipeline

REVIEW

CODE REVIEW

HISTORY

## Meets Specifications

Very good work overall, congratulations. You should be proud of your work in this project.  
Congratulations and stay safe

## Github & Code Quality

All project code is stored in a GitHub repository and a link to the repository has been provided for reviewers.  
The student made at least 3 commits to this repository.

Excellent all project files are with in the Github repo and the student has made atleast 3 commits

Actions Projects Wiki Security Insights

master 1 branch 0 tags

Go to file Add file Code

AkshayJaitly Updated README.md de115f3 23 hours ago 9 commits

File	Commit Message	Time
app	Added app changes for Flask UI with plotly dashboard and catgorical	23 hours ago
data	Messages and categories and SQLite db added	23 hours ago
gif	Added Readme.md, images and gif	23 hours ago
images	renamed png file	23 hours ago
models	Added ML training model which uses sklearn and gridsearchcv to train ...	23 hours ago
.gitignore	Initial commit	yesterday
LICENSE	Initial commit	yesterday
README.md	Updated README.md	23 hours ago

About Disaster Response pipe 8 data

Readme MIT License

Releases No releases published

Languages Python 81.4% HTML

README.md

## Disaster Response Pipeline Project

The README file includes a summary of the project, how to run the Python scripts and web app, and an explanation of the files in the repository. Comments are used effectively and each function has a docstring.

A good summary of the project is provided in your work. Furthermore, the information to run the Python scripts and web app are clearly contained in your report. Moreover, each function has a docstring. Nice layout! 🙌  
(Already passed by the previous reviewer)

Scripts have an intuitive, easy-to-follow structure with code separated into logical functions. Naming for variables and functions follows the PEP8 style guidelines.

Good code formatting. Your implementation logic is good and well organized. Also, naming for variables and functions follows the PEP8 style guidelines. Awesome work!  
(Already passed by the previous reviewer)

The ETL script, `process_data.py`, runs in the terminal without errors. The script takes the file paths of the two datasets and database, cleans the datasets, and stores the clean data into a SQLite database in the specified database file path.

The ETL script runs in the terminal without errors. That's neat!  
(Already passed by the previous reviewer)

The script successfully follows steps to clean the dataset. It merges the messages and categories datasets, splits the categories column into separate, clearly named columns, converts values to binary, and drops duplicates.

Good work! The script successfully follows steps to clean the dataset. The goal of this rubric is satisfied, you should be proud of yourself. 🙌  
(Already passed by the previous reviewer)

## Machine Learning

The machine learning script, `train_classifier.py`, runs in the terminal without errors. The script takes the database file path and model file path, creates and trains a classifier, and stores the classifier into a pickle file to the specified model file path.

The machine learning script runs without errors and creates and trains a classifier.  
(Already passed by the previous reviewer)

The script uses a custom tokenize function using nltk to case normalize, lemmatize, and tokenize text. This function is used in the machine learning pipeline to vectorize and then apply TF-IDF to the text.

The script perfectly uses a custom tokenize function and set it to the tokenizer parameter in the CountVectorizer step in the pipeline. 😊

(Already passed by the previous reviewer)

The script builds a pipeline that processes text and then performs multi-output classification on the 36 categories in the dataset. GridSearchCV is used to find the best parameters for the model.

You have perfectly used GridSearchCV to find the best parameters for the model. Well done!  
(Already passed by the previous reviewer)

The TF-IDF pipeline is only trained with the training data. The f1 score, precision and recall for the test set is outputted for each category.

The TF-IDF pipeline is only trained with the training data.  
The f1 score, precision, and recall for the test set is outputted for each category.  
(Already passed by the previous reviewer)

## Deployment

The web app, run.py, runs in the terminal without errors. The main page includes at least two visualizations using data from the SQLite database.

The web app runs in the terminal without errors. Great work here!  
You did very well by adding the visualizations, nice work in run.py.  
(Already passed by the previous reviewer)

When a user inputs a message into the app, the app returns classification results for all 36 categories.

When a user inputs a message into the app, the app returns classification results for all 36 categories. 🎉  
(Already passed by the previous reviewer)

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

Rate this review