 # DisasterResponseProject

Author: Angelina Espinoza-Limon

Project description:
This project presents the anaysis for the project Disaster Response. This analysis is performed over a set of messages that were captured with the Figure Eight system.

The analysis concludes if a message belongs to some of the following categories: 'related', 'request', 'offer', 'aid_related', 'medical_help', 'medical_products', 'search_and_rescue', 'security', 'military', 'child_alone', 'water', 'food', 'shelter', 'clothing', 'money', 'missing_people', 'refugees', 'death', 'other_aid', 'infrastructure_related', 'transport', 'buildings', 'electricity', 'tools', 'hospitals', 'shops', 'aid_centers', 'other_infrastructure', 'weather_related', 'floods', 'storm', 'fire', 'earthquake', 'cold', 'other_weather', 'direct_report'.

Each message might have several categories, then the problem requires a multi-class, multi-label and multi-output classifier. Therefore, the problem is addressed by coding a grid search to find out the best hyper-parameters for the ExtraTrees and RandomForest classifiers, which are recomended for these kind of multi-class, multi-output classification.

Thereafter, it is firstly cleaned the data, then the messages tokens are extracted and its corresponding TF-IDF matrix is obtained. Then, it is defined a gridsearch and the hyper-parameters options for the ExtraTrees and Random Forest classifiers are set in the grid. This new model is then able to be fitted to predict the message categories over test or unknown data. This analysis includes the bias-variance trade-off to evaluate the model precision, recall and the f1-score.

Data Cleaning:

The data is loaded from the messages.csv and categories.csv files to the messages and categories dataframes. Some pre-processed was done to merge the two dataframes, and for eliminating duplicates. The resulted dataframe has been saved into the DisasterResponse.db with SQLite.

Modeling Process:

From this dataset, it was eliminating from the messages the stopwords, and it was used lemmatization for getting the tokens.

It was defined a pipeline for making a feature union between a new feature (the messages lengh, by using a text lengh extractor) and the messages vectors (by using the CountVectorizer and TfidfTransformer methods) for getting the term frequency (TF-IDF matrix).

The model was defined with a GridSearch to find the optimal hyperparameters for the ExtraTreesClassifier and the RandomForestClassifier classifiers. The model is tested on the testing data and the prediction is compared to the true label, in order to obtain the precision, recall and F1 metrics for each category of the predictions vector.

Model Predictions:

Please, be aware that this model is able to categorize text messages into the following classes (considering the potential bias that is previously commented due to the imbalance dataset used for training the model):

'related', 'request', 'offer', 'aid_related', 'medical_help', 'medical_products', 'search_and_rescue', 'security', 'military', 'child_alone', 'water', 'food', 'shelter', 'clothing', 'money', 'missing_people', 'refugees', 'death', 'other_aid', 'infrastructure_related', 'transport', 'buildings', 'electricity', 'tools', 'hospitals', 'shops', 'aid_centers', 'other_infrastructure', 'weather_related', 'floods', 'storm', 'fire', 'earthquake', 'cold', 'other_weather', 'direct_report'.

Imbalance of the dataset:

This dataset is imbalanced (ie some labels like water have few examples). Thus, the results of the training model are biased, meaning that some messages might not be properly classified in all the categories. This would affect the precision in the validation step, since some messages which contain keywords that are not present in the current dataset (for instance fire) might not be classified with a proper precision. The recall is also affected since the recall figures might not be as expected for those keywords with few presence or at all, in the dataset.

Projects files:

Data files:

- disaster_categories.csv: File containing the categories of the messages.
- disaster_messages.csv: File containing the messages in plain text.
- DisasterResponse.db: Database created with SQLite, containing the messages and categories.

Python files:

- process_data.py
  This file contains the script for the Python code to load, clean and create the DisasterResponse database

- train_classifier.py
  This file contains the script for the Python code to load from the database the messages (already cleaned), to train a model and to

save the model into a pickle file (BestModelAndGridSearch.pkl) ready to be used.

Model files:

- classifier.plk: This file contains the deployed model already training with the train_classifier.py code.

Web App files:

- master.html: This file contains the code to publish the webpages, including the data visualizations, these are two different grahps in the home page for showing the messages count per category and the message genres distribution per category.

The Navigation menu is on top, from which it is possible to go to the window for entering the message to be classified ("Disaser Response Project" option), the link to Udacity ("Made with Udabity" option) and the contact page ("Contact" option).

- go.html: This files extends code from the master.html with more java script code.

- run.py: Contains script code with Python, for obtaining the data from SQLite, to upload the model, and the visualizations code. It also contains the code for linking the webpages with the Python code.

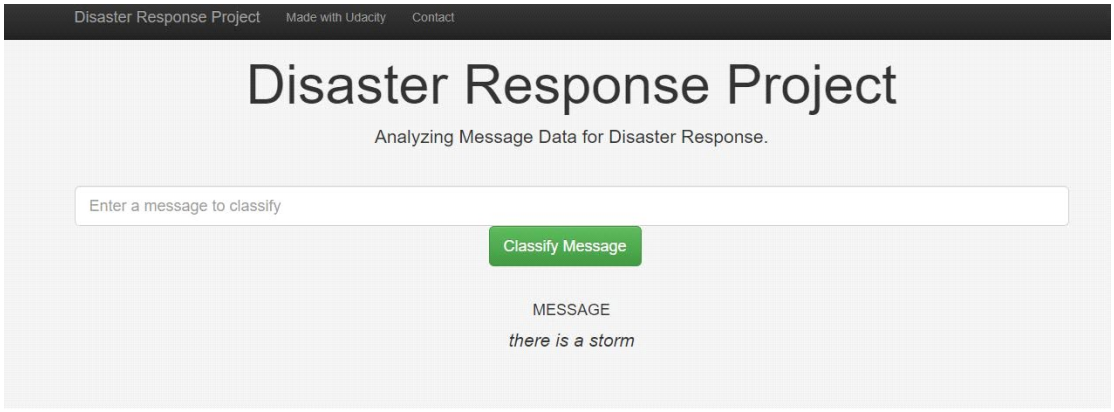
App requirements:
- This app was programmed with Python 3.6.3
- You should have as OS: Ubuntu 16.04.7 LTS

Installation instructions:
1. Download the zip folder of the project in your local folder
2. Unzip the folder and get the folders: app, data, models
3. Run the following command in the app's directory to run the web app:

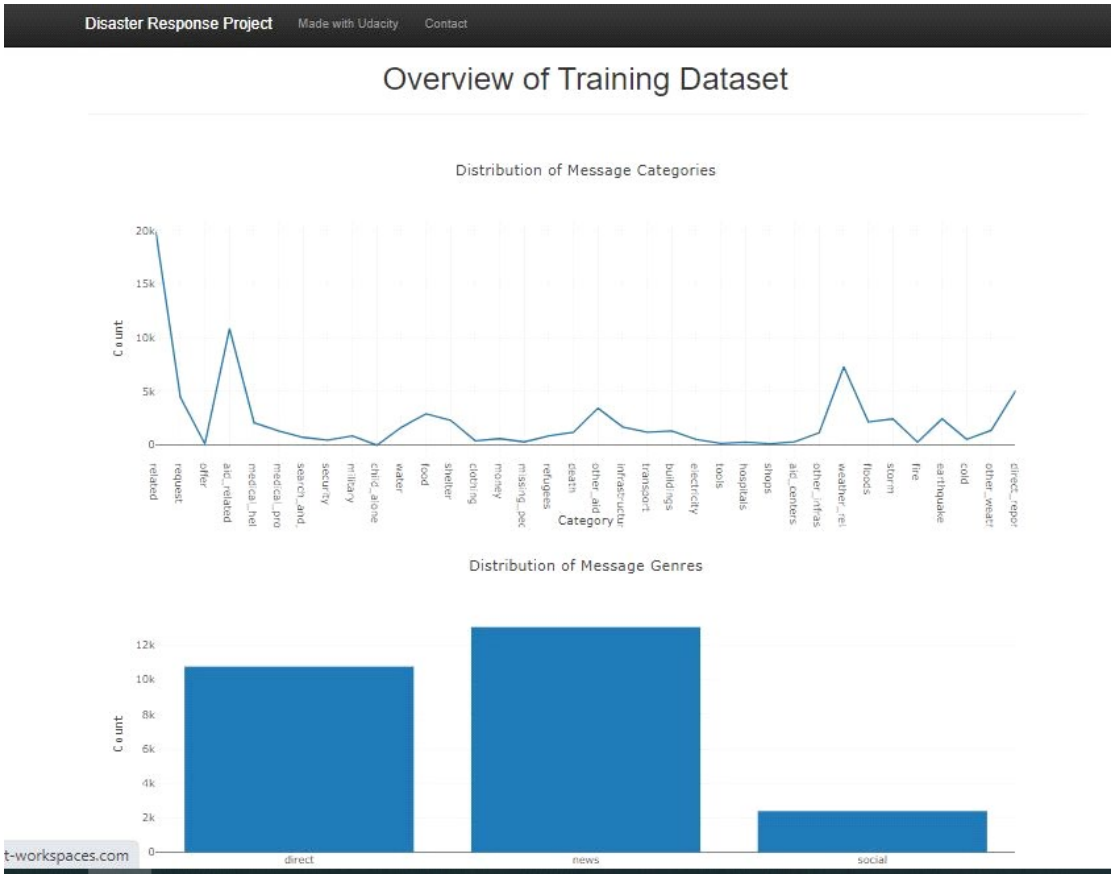
    python run.py

4. Go to http://0.0.0.0:3001/

Usage of the app:
1. In the web site home, you will have the following image:



and some graphs of the training data used for this model:



- Introduce a message text and the model will give a classification result, as the following:

Acknowledgements:
This course has been financed by Enterprise Ireland and the Marie Sklodowska-Curie actions, through the project: Career-FIT MF20180009.