Twitter Disaster Detection



In our final project of the general track, we are on the mission of saving lives.

Tweeting in case of fire might put you into an unnecessary danger, but it hardly prevents people from doing it. Though we can’t change human nature, we can at least put it to good use. Disaster tweets may inform others about an impending danger and save lives.

However, if you decide to build an app which makes your phone vibrate whenever someone tweets ‘fire’ in your vicinity, you are going to be bombarded with false alerts. Yes, people do say ‘fire’ while talking about [anything](https://www.urbandictionary.com/define.php?term=Fire) but the actual fire (and normally much better things), and same rule applies to many other words, which may otherwise indicate a life-threatening disaster. It’s clear that looking for keywords only won’t cut it - context is key. We would need an NLP algorithm, capable of identifying disaster tweets and only ring in case of emergency.

## The Data

You will work with a [collection](https://drive.google.com/file/d/1BDthbCVuzoen327AVyhZnP-3aI2QZXZO/view?usp=sharing) of over 10,000 tweets. Some of them talk about real disasters, and some talk about something entirely different, while using similar wording (‘fire’, ‘blast’, ‘ablaze’ etc). The dataset contains following columns:

* Tweet ID
* The text of a tweet
* A keyword from that tweet (if known)
* The location the tweet was sent from (if known)
* Target (1 for disaster, 0 for false alert)

## The Task

Your task is to build a model which is capable of correctly classifying tweets as either disaster tweets or false alerts

* Although this is primarily a NLP task, feel free to use all the methods you consider appropriate
* Since tweets are labeled, you have a chance to test the performance of your model
* You may use additional data

## Deliverables

* A public github repository (see [Using Git](#_pcnhnj32fvac)), containing
  + At least one Jupyter Notebook with solutions and explanations of your approach
  + Your presentation slides
  + A Readme.md that explains what the different notebooks/branches (if any) contain in your repository
* Feel free to include visualizations
* 15 minutes presentation per group

## Using Git

As this is somewhat of a dry-run for the certification project, you will use a github repository to collaborate on this project. For differently complex git workflows, consider [this resource](https://zepel.io/blog/5-git-workflows-to-improve-development/) to get a better idea of how you can utilize branches for this project. Make sure that each member of your team commits (and pushes) their code (in any branch) to the github repository *on a daily basis*.

Previously, you collaborated on google colab notebooks because you could easily share your code with the sending of a link. However, google colab has several drawbacks which make it unsustainable to work with the colab technology in your later career.

* It has poor versioning support
* You are at the mercy of google’s resource limitations
* Telekom is not relying on the google tech-stack

So, sooner or later you will work without colab anyway. Thus, you should use [Jupyter Notebook or Jupyter Lab](https://jupyter.readthedocs.io/en/latest/install.html) in a local environment on your machine, and share the code you’ve been working on, by pushing and pulling from/to the git repository (instead of sending files through slack/email/etc.).