**By Shreyas nanaware**

Multi-Class Text classification

**Introduction**

Problem Statement

Given a list of statements about different topics, create a machine learning model that is capable of classifying a new statement into a topic.

Following are the list of topics:

* Animals: Statements talking about animals
* Compliment: Mostly positive, encouraging statements
* Education: Centers around education and their policies, schooling, etc.
* Health: News about health
* Heavy Emotion: Statements that convey mostly negative or strong emotion, usually anger
* Joke: Statements that are part of a joke
* Love: Romantic statements, experiences, or about love itself
* Politics: Everything going on in governments worldwide, mostly U.S.
* Religion: Discussion about all kinds of religion
* Science: Various statements concerning discoveries or research in science
* Self: Statements where the speaker is talking about themselves

E.g.

Statement : “The eternal mystique of Goldman Sachs”

Topic: “politics”

Model used for this classification problem is *BERT* (Bidirectional Encoder Representation from Transformers), it is the state of the art language model for NLP ( Natural language processing ) tasks.

**Design**

**Implementation**

**Folder Structure**

* Multi-Class Text Classification
  + Contains *main.py.*
* Predictions
  + Contains *evaluate\_model.py.*
    - Model
      * Contains trained model
* Rest Api
  + Contains *RestApi.py*
* Training
  + Containing .pynb files containing the training phase for all ML, NN and BERT models.

Implementation

1. Main.py

This is the main python program file which runs the flask framework to get the classification result.

1. Evaluate\_model.py

This file performs the following :

* + Load pre-trained model
  + Pre-process data in a format accepted by the model.
  + Model used: BERT
  + Perform prediction and return the result

1. RestApi.py

This is a helper class to run the flask framework that runs the get method to display the result in the json format on the server.

Guide

Steps to run the program:

* **Installation**: before running the main file, install all the dependencies using the *requirements.txt* file.

Code: ‘*python install -r requirements.txt’*

* Run the main file.

Code: ‘python main.py <Enter the statement in JSON format>’

e.g. python main.py '{"statement":" The eternal mystique of Goldman Sachs “}'

* Flask framework Initialized:

After application has started running on the given port, open the URL in the browser.

Copy and paste the URL in the browser and add the parameter ‘/topic’ in the URL.

e.g. ‘http://0.0.0.0:4444/topic’

* Result will be displayed in the JSON format.