# **AI PROJECT REPORT (INT 404)**

**TWITTER SENTIMENT ANALYZER**

GROUP 5:

* Debashis Panda (11801275)
* Diyali Shivhare (11801277)
* Abhinandan Chabri (11801320)

The aim here is to correctly determine the sentiment behind the daily tweets and perform analysis of the positive and negative tweets. This is not a actual app but this app only focuses on the idea behind the analysis of the tweets and provide the output as positive or negative with a confidence percentage using the sentiment classification feature of NLP (Natural Language Processing). This app is hosted online on Alibaba cloud service.

Try it out at this link <https://149.129.167.205:8501>

**Table of Contents**

* Packages Used
* Dataset Used
* Training of ML Model
* Working of ML model
* Deploying the ML model
* Output of the Model

# Packages Used

The following Libraries were Used to build the project:

* Flair
* Pytorch
* Pandas
* Streamlit
* Twitter Scraper

# Dataset Used

The dataset which was used was a set of tweets. It contains 1,600,000 tweets extracted using the twitter API. The tweets have been annotated (0 = negative, 4 = positive) and they can be used to detect sentiment.

It is in CSV format. It contains six fields: target, ids, date, flag user, text. The dataset was divided into three sets – Test data, Training data, and dev data. You can find the dataset here at Kaggle - <https://www.kaggle.com/kazanova/sentiment140>

# Training of ML Model

The model was trained on 1.6 million tweets at 5 epochs. The model here is not so accurate due to RAM issues because it needs a lot of Ram to process the large amount of data so I have trained at a less no of Epochs.

# Working of ML Model

The Library used to train the model is Flair, a state of art NLP library. This framework builds directly on [PyTorch](https://pytorch.org/), making it easy to train models and experiment with new approaches using Flair embeddings and classes.

* The data was first pre-processed to extract the correct format which is used to display the data.
* Then the data is divided into three parts – train data, dev data and test data.
* The data is loaded in the corpus format.
* The text classifier model and model trainer is initialised and the hyper-parameters are set.
* Then the model starts training and a model file is generated in the path specified.
* The the model initialised at first can be used to predict the sentences
* I have used another Library known as Streamlit which is a framework for building the custom ML tools.
* The streamlit is used here to build the frontend which directly deploys the app in a localhost which we can run on any browser.
* There is also another library known as Twitter Scraper which crawls for specified tweets and displays all the positive and negative tweets and the ratio for the positive and negative tweets.

# Deploying the ML Model

The model was deployed on Alibaba Cloud Services using their Elastic Compute Service.

The Specifications of the instance is:

Service name: Elastic Computer Service

vCPU: 2

RAM: 2GB

SSD: 40 GB Ultra Disk

OS: Ubuntu

# Output of the Model

# 

