🚀 Welcome to the Emotion Detection Challenge on Twitter!

You're about to dive into a hackathon where your mission is to detect emotions in tweets. Let's get started!

🌟 The Challenge

Imagine sifting through the endless stream of tweets and figuring out the prevailing emotion. In this challenge, you won't just focus on classic positive or negative sentiment. Instead, you'll tackle the more intricate task of identifying **four core emotions**:

- 😠 Anger
- 😃 Joy
- 😢 Sadness
- 😊 Optimism

Your goal? Assign the most dominant emotion to each tweet. Sounds fun, right? Let's see how you handle the nuances of human feelings, all packed into 280 characters!

Important Reminders

1. Remember to upload your **best solution** (CSV file + Zip file with code and model) in your team's folder named **"Solutions"**.
2. Don't forget to also upload your **CV** in your team's folder named **"CV"**.

Your CV will be shared with **WeHunt**, which may offer you exciting job opportunities. Upload your CV **even if you are not at the top of the leaderboard**, as it will increase your chances of being noticed and considered for career opportunities.

🕛 The Timeline

You've got **just few hours** to bring your best solution to life. Think fast, code smart, and remember: your model should shine in detecting those emotions.

🏆 How You'll Be Judged

We're using the **Macro-Averaged F-1 Score** to evaluate your model. Why? Because it's perfect for handling this multi-class classification task and balancing out any imbalances in the data. So, make sure your model can handle all emotions equally well!

📊 Dataset Overview

You'll be working with a dataset of tweets, each labeled with one of the four emotions. Here's what you'll get:

- **Train_Text.txt:** A collection of labeled tweets in English, complete with the typical quirks of Twitter—hashtags, mentions, and more. It's up to you to decide how to handle these idiosyncrasies.
- **Test_Text.txt:** Similar to the Train Set, but unlabeled. Your task is to predict the emotions expressed in these tweets.
- **Val_Text.txt:** The Validation Set in text format.
- **Val_Labels.txt:** The corresponding labels for the Validation Set, will allow you to evaluate your model's performance before submission.
- **Simulated Solution.csv:** A sample CSV file that shows the expected format of your solution, guiding you on how to structure your final submission.
- **Mapping.txt:** Contains essential information about the label-to-emotion mapping.
- **Example_Prediction_Submission.csv:** An example submission file to give you clarity on how your predictions should be organized.

Make sure to submit the following:

1. **Predictions**: Your prediction results as a `.csv` file, formatted just like the Simulated Solution.
2. **Code and Model**: Zip your well-commented Jupyter Notebook with the code used to train your model and generate the predictions.

We love clean and organized code, so don't forget to include clear explanations and comments to show your thought process!

🔐 The Rules

- Use **any Machine Learning model** you want, but **no pre-trained models** are allowed. We want to see what you can build from scratch!
- Your solution should be **your own**. Don't rely on any ready-made services or cloud-based models.
- Submit a `.csv` file with your predictions, and remember to keep the test data in the same order as provided.

🛠️ What You Need to Submit

1. **Predictions**: A `.csv` file with each tweet from the test set labeled with its corresponding emotion.
2. **Code**: A zipped file containing your Jupyter Notebook with the corresponding code, all the magic behind your solution. We're looking for smart, well-explained code—so impress us with your insights!

📈 Evaluation Criteria

Your model will be judged based on its **macro-averaged F-1 score**, making sure it performs well across all emotion categories.

⚠️ Important Note

The **final leaderboard** will be **temporary** until we have fully verified the results. So, stay tuned for the official rankings once the results have been reviewed!
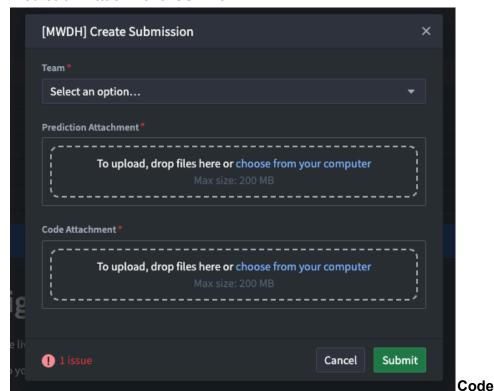
📥 File Upload for Automatic Leaderboard Calculation

This is an example of where you will need to upload your files for the **automatic leaderboard calculation**. Follow the guidelines carefully to ensure your submission is processed smoothly and accurately. Stay tuned for further instructions on the upload platform.

- **Prediction Attachment:** CSV file

- **Code Attachment:** Zip file with code and model

🛠️ Additional Support:

- **#Server Discord ODP**: https://discord.gg/9PnBQhAs38
- **#Server HCAI2024**: https://discord.gg/9PnBQhAs38
- **#Support_Ticket**: https://discord.gg/aTTmDJA3Kg

When you join **Discord**, we will need to assign you the role of **Player** so you can access the servers mentioned above. If we don't do it right away, please send us a direct message so we can assign the role, allowing you to see all the servers.

When you connect to **Foundry**, let us know (if we don't see it immediately) so we can connect you to the correct folder.

⚙️ Resources

You are free to use your own resources or leverage **Foundry** by using the **Jupyter Notebook Space**, an app available within the Foundry platform. This gives you flexibility in how you approach the challenge, so take advantage of whichever setup works best for you!