# Topic

### **1. Understand the Task**

* **Goal**: Predict the occurrence of specific sub-events (e.g., goals, yellow cards) in tweet streams during football games.
* key words are *'full time', 'goal', 'half time', 'kick off','owngoal', 'penalty', 'red card', 'yellow card' ，*  *'other' : injure(d)*
* **Dataset**:
  + Tweets are grouped into 1-minute intervals.
  + Each tweet includes fields like Tweet, EventType (binary labels: 0 or 1), etc.
* **Evaluation Metric**: Accuracy.

### **2. Team Task Allocation**

**Data Preprocessing**: Clean and prepare the dataset.

1.remove useless words

2.remove symbols

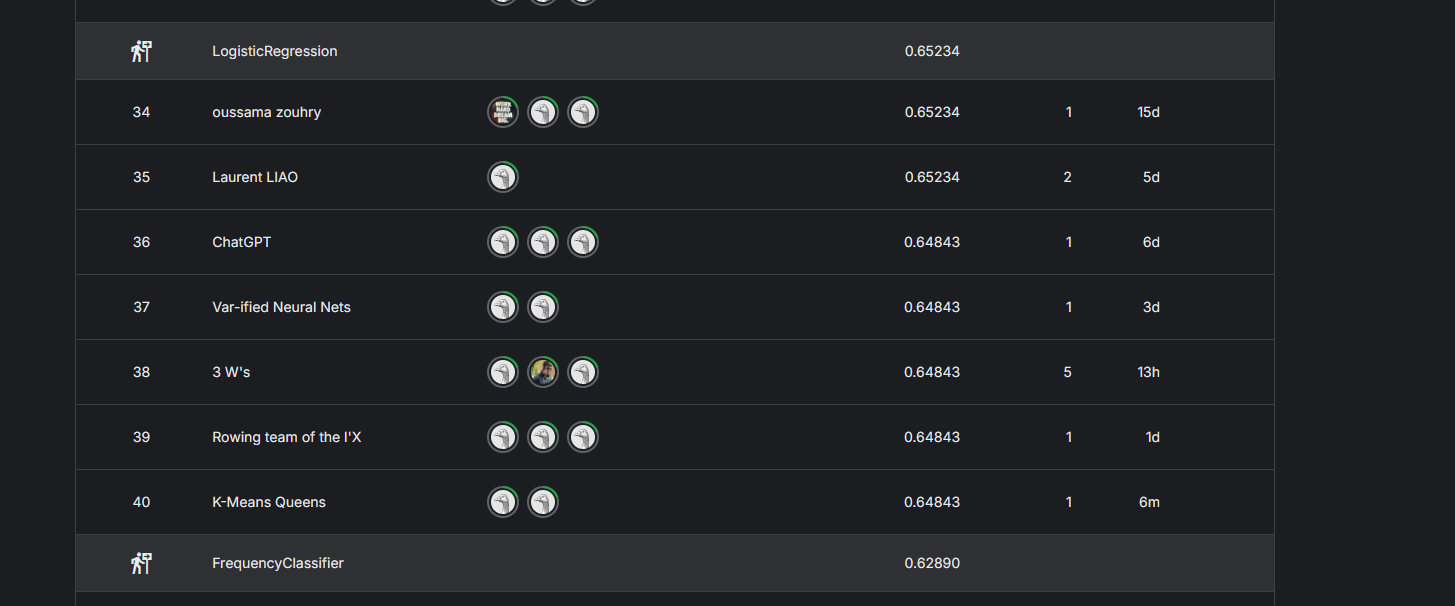
3.remove urls

4. make time stamps associate with time and matchID and date

5. …….

6……

**Model Development**: Explore and train classification models.

try out LogisticRegression 。 FrequencyClassifier from tool

**Result Analysis**: Evaluate model performance and refine the approach.

….

**Documentation & Presentation**: Summarize results and prepare deliverables.

….

# Implementation Steps

#### **Step 1: Data Preprocessing**

1. **Load Data**:
   * Use the baseline.py script to understand how to read and process data.
   * Extract tweets and labels from train\_tweets/\*.json.
2. **Clean Text Data**:
   * Remove stopwords, punctuation, and other noise. like basline
   * Use libraries like NLTK or spaCy for tokenization and stemming/lemmatization.
3. **Feature Engineering**:
   * **Word Embeddings**: Use pre trained embeddings like GloVe or Word2Vec.
   * **Text Features**: Extract n-grams, keyword counts, or TF-IDF vectors.
   * **Temporal Features**: Leverage timestamp data to derive time-based patterns.

#### **Step 2: Model Development**

1. **Choose Models**:
   * **Traditional Models**: Logistic Regression, Support Vector Machines (SVM).
   * **Deep Learning Models**: RNNs, LSTMs, or Transformers (e.g., BERT).
2. **Extend baseline.py**:
   * Use scikit-learn for feature engineering and baseline improvements.
   * Alternatively, implement fine-tuning with Hugging Face Transformers for models like BERT.
3. **Hyperparameter Tuning**:
   * Use tools like GridSearchCV or RandomizedSearchCV for optimization.

#### **Step 3: Model Evaluation**

1. **Validation Set**:(optional, i think)
   * Split 20% of the training data for validation.
   * Use cross-validation to avoid overfitting.
2. **Evaluation Metrics**:
   * Focus on accuracy but also review confusion matrices and F1 scores.
3. **Error Analysis**:
   * Analyze misclassified samples to identify improvement areas.

#### **Step 4: Result Submission**

Format your predictions according to dummy\_predictions.csv. and feed it to the website to get a score !

# Extensions and Optimization

1. **Incorporate Context**:
   * Analyze tweets from adjacent time periods for contextual information.
2. **Multi-task Learning**:
   * If relevant, consider extending the model to predict multiple event types simultaneously.
3. **Ensemble Models**:
   * Combine predictions from multiple models for improved accuracy.
4. **Sentiment Analysis**:
   * Extract sentiment features from tweets to distinguish different types of sub-events.

# Team Responsibilities

1. Handle data cleaning and feature extraction.

2.Develop machine learning models based on the baseline.

3. Implement deep learning models.

4. Analyze model performance and prepare the final submission.

# Immediate Priorities

1. Understand and execute the code in baseline.py to load and process data correctly.
2. Perform an exploratory data analysis (EDA) to understand tweet and label distributions.
3. Build a simple logistic regression or SVM model as a baseline to establish a workflow.