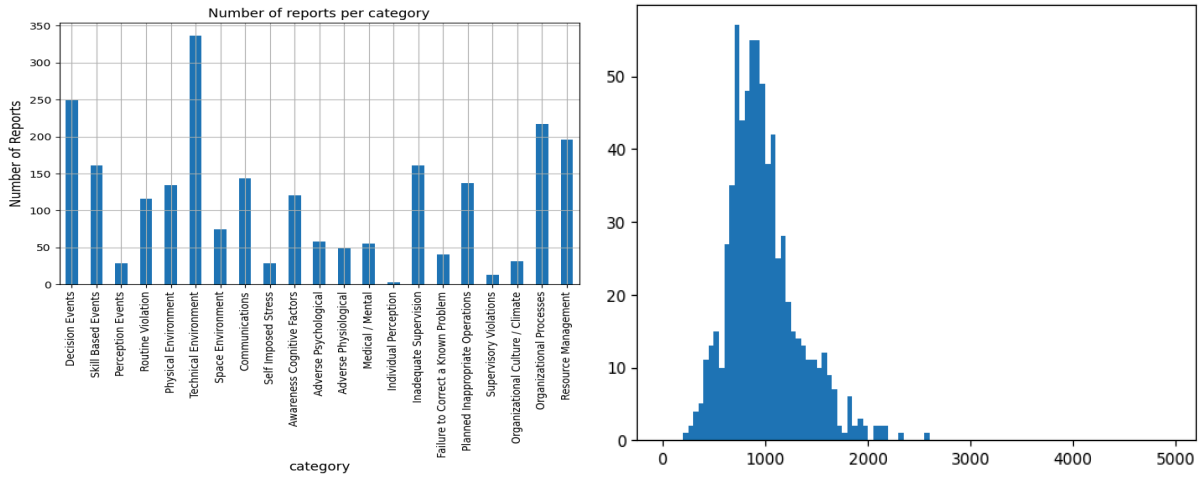
MARITIME ACCIDENT ANALYSIS USING AI

This project is part of my research internship at the **ARIES Research Group**, Universidad De Nebrija, under the guidance of **Prof. Christian Velasco Gallego** and **Dr. Beatriz Navas De Maya**. I worked on the topic "Maritime Accident Analysis using AI," where we used the **HFACS** framework developed by **NASA** to classify and analyze accident reports. The project had two stages. In the first stage, I prepared a dataset from accident reports released by the governments of the **US** and **UK**. I manually labeled **700** reports based on the abstract section, which contains details such as the accident summary, safety issues, location, date, and title.

In the initial stages of the project, we developed an **RAG** model for classifying the reports. However, due to its poor performance, we decided not to proceed with this approach. We then shifted our focus to traditional machine learning-based approaches. Initially, we applied various preprocessing techniques to clean the data, including:

1. Combining both summary and safety issues by concatenating them.
2. Removing newline characters and converting text to lowercase.
3. Removing labels with less than 10% of the total data points.
4. Applying additional preprocessing techniques using NLTK.



Word count and number of reports per category

We used the **TF-IDF Vectorizer** to convert the text data into numerical representation and applied five different machine learning models, such as LinearSVC, Decision Tree, and Logistic Regression. We also explored deep learning-based approaches but found that basic machine learning models performed better. This is likely due to the small dataset and low feature-to-data-point ratio. We had 11 different labels (after removing minority labels) and 700 data points. The model performance was assessed by varying different models and their respective parameters, using 0.15 as the test size ratio.

Given the limited data, we attempted to upsample it. For our multi-label classification problem, we used **MLSMOTE** (SMOTE for multi-label datasets) as part of preprocessing. However, this did not improve our model's performance. Due to data imbalance, the recall value was low for labels with fewer data points and higher for the majority labels. We couldn't undersample the data because of the limited data points available. Below are links to learn more about the MLSMOTE approach:

- Handling Data Imbalance in Multi-label Classification (MLSMOTE) | by Niteshsukhwani | TheCyPhy | Medium
- SMOTE for multilabel classification - Stack Overflow
- GitHub - theopsall/multiSmote: A multi-label approach of the SMOTE algorithm

After classification, we moved on to keyword analysis, applying the following three techniques:

1. **Wordcloud**: This library was used to identify the top keywords based on their frequency in each label. You can check out the Wordcloud folder to see the visualizations generated.
2. **TextRank**: This technique was utilized to retrieve the top keywords based on the structure and context of the incidents.
3. **KeyBERT:** This approach uses cosine similarity and BERT embeddings to retrieve the top sub-documents (keywords) that are similar to the overall document. We applied two approaches:
    - **Sentence Transformers**: I selected the following five publicly available sentence transformer models: All-mpnet-base-v2, Multi-qa-mpnet-base-dot-v1, All-distilroberta-v1, All-MiniLM-L12-v2, and Multi-qa-distilbert-cos-v1.
    - **Flair**: Flair allows us to choose any publicly available embedding model. I used the popular BERT base uncased, RoBERTa base, and DistilBERT base uncased models.

I applied keyword analysis methodologies to both the safety issues and the overall dataset, retrieving the top **10** keywords for each label according to the KeyBERT model. In this approach, we identified the primary keywords for each report and recorded their occurrences. We then grouped the keywords for each label and identified those that appeared most frequently. Of the three approaches, I found that the KeyBERT approach provided better keywords compared to the other two approaches.