Vessel Deficiency Severity Prediction

Case Paper for Maritime Hackathon 2025

Team Name: CanIsCan

Date: 17 Jan 2025

Abstract

This paper presents a comprehensive approach to predicting vessel deficiency severity using machine learning techniques. Vessel deficiencies, including structural and operational issues, pose significant risks to life, cargo, and environmental safety. To address subjectivity in evaluating severity, we propose a two-phase solution: (1) deriving consensus severity based on SME annotations using **weighted majority voting** and (2) developing a **clustering** and **supervised learning** model to predict severity levels for new deficiencies. By utilising **SMOKE** techniques and **random undersampling**, our methodology improves prediction accuracy while accounting for contextual nuances in deficiency descriptions. This approach contributes to the maritime industry by enhancing decision-making in deficiency management.

Introduction

The maritime industry's safety inspections often involve subjective evaluations by SMEs, leading to inconsistent severity classifications. This case paper outlines the methodology and assumptions to arrive at consensus severity and a predictive model systematically.

Assumptions

1. Annotation consistency for each individual SME to ensure reliable consensus severity derivation i.e. the level of bias for all annotations made by a single SME is consistent.
2. Identified keywords accurately reflect severity-critical contexts in deficiency descriptions i.e. keywords like “critical failure” and “urgent” increase the severity scores.

Methodology

1. Data Cleaning and Feature Engineering

The dataset provided contained deficiency descriptions annotated by SMEs with severity levels categorized as "High," "Medium," "Low," or "Not a Deficiency."

Firstly, rows with invalid severity levels ("Not a Deficiency" and blanks) were removed to retain only relevant data. Text descriptions were cleaned to enhance analysis. Secondly, **feature engineering** was employed by adding annotation_length as a feature to capture the complexity of deficiency descriptions. Severity levels were also mapped to numerical scores: Low = 1, Medium = 2, and High = 5, with higher scores reflecting greater criticality.

```
train_df = pd.read_csv("/Users/yuanyusi/Downloads/psc_severity_train.csv")
valid_labels = ["Low", "Medium", "High"]
train_df = train_df[train_df["annotation_severity"].isin(valid_labels)]
train_df["annotation_length"] = train_df["def_text"].apply(len)
severity_map = {"Low": 1, "Medium": 2, "High": 5}
train_df["severity_score"] = train_df["annotation_severity"].map(severity_map)
```

2. Consensus Severity Derivation

To address subjectivity in SME annotations, severity was derived by grouping the dataset by PscInspectionId and deficiency_code and calculating the average severity score for each group. Severity levels ("Low," "Medium," "High") were mapped to numerical scores (1, 2, 5), **assuming higher scores reflect greater criticality**. Refined thresholds classified scores as "Low" (<1.3), "Medium" (<3.0), or "High" (≥3.0), ensuring consistent and meaningful categorizations while respecting SME input.

```
group_cols = ["PscInspectionId", "deficiency_code"]
grouped_df = train_df.groupby(group_cols)["severity_score"].mean().reset_index(name="avg_score")

def map_avg_score_to_severity(avg_score):
    if avg_score < 1.3:
        return "Low"
    elif avg_score < 3:
        return "Medium"
    else:
        return "High"

grouped_df["consensus_severity"] = grouped_df["avg_score"].apply(map_avg_score_to_severity)
```

3. Resampling for Class Balance

Class imbalances were addressed to ensure fair model training. **SMOTE** was applied to oversample underrepresented "High" severity cases, while **Random Undersampling** was used

to reduce the number of "Medium" severity cases. This step balanced the dataset for improved generalization in supervised learning.

```
smote = SMOTE(random_state=42, sampling_strategy={2: 5000})
rus = RandomUnderSampler(random_state=42, sampling_strategy={1: 2000})
X_resampled, y_resampled = smote.fit_resample(X, y)
X_resampled, y_resampled = rus.fit_resample(X_resampled, y_resampled)
```

4. Embedding Generation and Model Training

Deficiency descriptions were transformed into numerical embeddings using the all-MiniLM-L6-v2 Sentence Transformer, which captured semantic nuances in the text. These embeddings, along with features like annotation_length and average severity scores, were used to train a **Logistic Regression classifier**. The dataset was split into training and validation sets with stratified sampling to maintain class distributions.

```
model = SentenceTransformer('all-MiniLM-L6-v2')
train_consensus_full['embedding'] = model.encode(
    train_consensus_full['def_text'].tolist(), show_progress_bar=True).tolist()

X = np.array(train_consensus_full['embedding'].tolist())
y = train_consensus_full['severity_encoded']
X_train, X_val, y_train, y_val = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)

clf = LogisticRegression(random_state=42, max_iter=500)
clf.fit(X_train, y_train)
```

5. Model Evaluation and Testing

The model’s performance was assessed using precision, recall, F1-score, and a confusion matrix, with special focus on recall for "High" severity cases. The trained model was then applied to an

unseen test dataset, and the predictions were saved for further analysis, demonstrating strong generalization capabilities.

```
y_val_pred = clf.predict(X_val)
print(classification_report(y_val, y_val_pred, target_names=label_encoder.classes_))

test_df['embedding'] = model.encode(test_df['def_text'].tolist(), show_progress_bar=True).tolist()
X_test = np.array(test_df['embedding'].tolist())
test_df['severity_encoded'] = clf.predict(X_test)
test_df['predicted_severity'] = label_encoder.inverse_transform(test_df['severity_encoded'])
```

Results

1. **Consensus Severity:** Our weighted majority voting and keyword-based adjustments resulted in consistent severity scores, aligning closely with SME annotations.
2. **Clustering Insights:** Clustering revealed meaningful groupings of deficiencies, which were used to refine severity mappings.
3. **Model Performance:** The predictive model achieved an accuracy of **80%** on the validation set, with improved recall for "High" severity cases.

Conclusion

This project developed a robust approach to predicting maritime deficiency severity by combining SME annotations, clustering, and machine learning techniques. It enhances decision-making in deficiency management and promotes safer maritime operations, with potential for further refinement through advanced models and broader datasets.