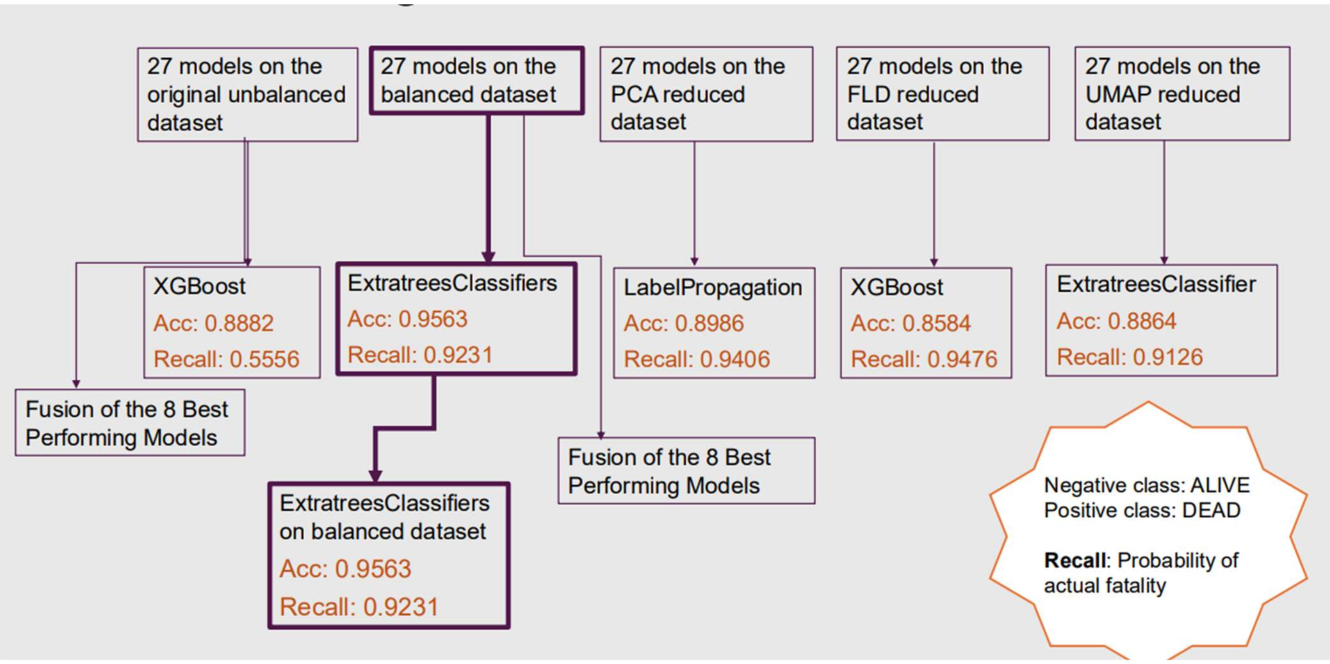
METHODS


27 models on the original unbalanced dataset
27 models on the balanced dataset
27 models on the PCA reduced dataset
27 models on the FLD reduced dataset
27 models on the UMAP reduced dataset
XGBoost
Acc: 0.8882
Recall: 0.5556
ExtratreesClassifiers
Acc: 0.9563
Recall: 0.9231
LabelPropagation
Acc: 0.8986
Recall: 0.9406
XGBoost
Acc: 0.8584
Recall: 0.9476
ExtratreesClassifier
Acc: 0.8864
Recall: 0.9126
Fusion of the 8 Best Performing Models
Fusion of the 8 Best Performing Models
ExtratreesClassifiers on balanced dataset
Acc: 0.9563
Recall: 0.9231
Negative class: ALIVE
Positive class: DEAD
Recall: Probability of actual fatality

After preprocessing our dataset, we had 5 versions of the dataset. The first is the preprocessed version of the original dataset, balanced version of the dataset with minority class oversampled using SMOTE, PCA reduced version, UMAP reduced version and FLD reduced version. We trained 27 classification algorithms on each version of the dataset and optimized the best performing algorithm in each case. We also fused the best 8 classifiers trained on the balanced dataset, leaving us with 6 candidate model to choose from. At the end, we selected ExtratreeClassifier trained on the original version of the dataset as our best option.

Brief Description

Extremely Randomized Tree Classifier (The ExtratreesClassifier) is an ensemble learning algorithm designed for supervised classification problems. It belongs to the family of decision-tree-based ensemble methods and operates similarly to Random Forests but with key differences in tree construction. During training, it builds multiple decision trees, combining their predictions to make a final decision. Unlike Random Forests, which select the best split for a feature based on criteria like Gini impurity or entropy, ExtraTrees introduces additional randomness by splitting at random thresholds for selected features. By introducing randomness, ExtraTrees reduces the variance of the model while retaining a similar level of bias as Random Forests. The algorithm is described mathematically as follows:
For a randomly chosen feature f and a random threshold t, we compute the split S as

S= $\{x|x[f] \leq t\} \cup \{x|x[f] > t\}$

where $x|x[f]$ is the value of a feature f for a sample x. We randomly choose t from the feature's value range in the data subset. Now, for T trees and a sample x, the final class y is predicted as:

$$y = argmax_c(\frac{1}{T}\sum_{i=1}^{T} P_i(y = c|x))$$

New Technique Used

In the state-of-the-art paper, 4 classifications were trained on just the original unbalanced dataset, but in our project, we created 5 different versions of our dataset after preprocessing, fit 27 classification algorithms on each of the 5 versions of the dataset, and optimized the best in each case through hyperparameter tuning. We also did a fusion of some of the classifiers before arriving at the optimal result. The state-of-the-art paper trained its model on an unbalanced dataset. We addressed this class imbalance by oversampling the minority class using SMOTE. With this, we were able to significantly outperform the SOTA paper.