## Dataset Description

The data for this project comes from the Global Shark Attack File (GSAF), a publicly available dataset that tracks shark attacks worldwide.  It contains 4,052 documented cases, each representing a distinct encounter between a shark and a human. The dataset spans more than a century, with entries ranging from the early 1900s to recent years, and includes a wide geographic scope, covering over 50 countries.

Each case includes detailed information about the circumstances of the attack, such as the date, location, activity of the victim, and the type of incident (for example, unprovoked, provoked, or related to sea disasters). Where available, demographic details like the victim’s age and sex are included, along with whether the attack was fatal or not. In many cases, the species of shark involved is also reported, though a significant number of entries contain missing or unknown values for this attribute.

This dataset offers a unique blend of biological, geographic, and behavioral variables, enabling valuable analysis of shark attack trends over time, location, and human activity. However, inconsistencies and gaps in fields such as age, shark species, and victim activity required significant data cleaning and preprocessing before analysis.

#### Dataset Composition:

* **4,052 observations** (individual shark attack cases)
* **Over 20 original features**, including:
* Temporal variables: Date, Year, Time
* Demographic variables: Sex, Age
* Behavioral context: Activity, Type (Unprovoked, Provoked, etc.)
* Geographical variables: Country, Area, Location
* Outcome variable: Fatal (Y/N)
* Biological info: Species of shark (when known)

#### Project Goal

This project aims to create a predictive model that assesses the probability of a shark attack being fatal, using contextual, demographic, and environmental data from the dataset. Although fatal shark attacks are uncommon, they present significant public safety risks. By analyzing historical attack patterns—such as victim activity, age, gender, time of year, and encounter type—I wanted to determine which factors most strongly contribute to fatal outcomes. This can provide insights for marine safety researchers and policymakers, as well as support the development of preventive strategies, risk awareness campaigns, and potentially even real-time risk assessment tools for beachgoers, lifeguards, and water sports professionals.

## ***1. Dataset Exploration***

#### 1.1 Importing libraries

I started the project flow by importing the necessary libraries used for data analysis and machine learning. These include pandas, used for data manipulation, numpy, used for numerical computations, matplotlib and seaborn – for data visualization.

A close up of a text

AI-generated content may be incorrect.

#### 1.2 Loading the dataset and displaying info

Next step was to load the dataset using pandas:

A close up of a text

AI-generated content may be incorrect.

To gain an initial understanding of the dataset, I used the .info() and .head() functions from the pandas library. The head() function was used to display first 10 rows of the dataset, and info() function provided a summary of the dataset structure:

A screenshot of a computer

AI-generated content may be incorrect.

#### 1.3 Displaying statistics for all columns and counting values in the target column

To better understand the structure and quality of the dataset, I used describe(include='all') to print descriptive statistics for all columns, and value\_counts() to explore the distribution of the target column, **'Fatal (Y/N)'**.

*Summary statistics:*

A screenshot of a computer

AI-generated content may be incorrect.

*A screenshot of a computer

AI-generated content may be incorrect.*

This revealed significant data quality issues, such as inconsistent formatting in categorical fields and a high number of missing values. Out of 25723 entries, over 19000 entries in the target column were missing, and some of the rest included invalid values (for example, ‘2017’). This data exploration pointed out the need for extensive data cleaning and preprocessing.

#### 1.4 Plotting barplot for top 10 most common values

To better understand the distribution of key categorical features in the dataset, I visualized the top 10 most frequent values for several columns using barplots. This was done using a simple loop that applied sns.barplot() to the following features:



These variables were chosen due to their relevance to the nature of shark attack incidents. The visualizations revealed that non-fatal incidents ('N') are more common, surfing and swimming are the most frequent activities during attacks, and most incidents occur in the USA and Australia.

These plots provided a clearer view of dominant patterns in the dataset and helped with feature selection in later stages of analysis.

A graph of a number of values

AI-generated content may be incorrect.

A graph of a number of people

AI-generated content may be incorrect.

#### 1.5 Correlation heatmap for numeric features

To examine the relationships between numerical variables, I generated a correlation heatmap using Seaborn's heatmap() function. The correlation matrix was calculated by first selecting numeric columns from the dataset with select\_dtypes():

A screen shot of a computer code

AI-generated content may be incorrect.

The heatmap revealed that there is a moderate positive correlation (0.39) between the Year and the original order column, which is expected considering that the dataset follows a chronological order. No other strong correlations were present, suggesting limited multicollinearity among the available numeric features.

A red and blue squares

AI-generated content may be incorrect.

## ***2. Data Preprocessing***

#### 2.1 Checking for missing values and duplicate rows

To prepare the dataset for analysis, I performed essential data cleaning steps. I began by dropping rows containing only missing values and removing irrelevant or duplicate columns such as additional case number entries and metadata. After cleaning, I verified the presence of missing values in all columns and found that some variables—like Age, Species, and Time—had a considerable number of null entries, while others had only a few. I also checked for and confirmed the absence of duplicate rows. At this stage, the dataset was reduced to 6,304 entries, providing a cleaner starting point for further handling of missing values and preprocessing.

A screenshot of a computer

AI-generated content may be incorrect.

#### 2.2 Handling missing values

To handle missing values, I began by standardizing key categorical columns—converting entries to uniform uppercase or lowercase and removing extra spaces for consistency. For the target column *Fatal (Y/N)*, ambiguous entries were excluded, and values were converted to a binary flag (1 for fatal, 0 for non-fatal). The *Sex* column was filtered to retain only valid entries (M or F). Critical features like *Activity, Sex, Country, Case Number*, and *Year* were mandatory, so any rows missing these were removed. The Age column was cleaned by extracting numeric values and filling missing entries with the median age. Less crucial fields such as *Type* and *Area* had missing values replaced with "unknown," while irrelevant columns like *Name, Time*, and *Location* were removed entirely. Although the Species column remained partially incomplete, I intentionally retained its missing values, as this feature could still provide valuable insights or patterns during future analysis or model interpretation stages (for example, using SHAP). After these steps, the dataset was reduced to 5,228 rows and 13 well-structured columns.

A screenshot of a computer

AI-generated content may be incorrect.

#### 2.3 Normalization

To normalize the Age feature, standard scaling was applied using StandardScaler from scikit-learn, which transformed the values to have a mean of 0 and a standard deviation of 1.

A screenshot of a computer code

AI-generated content may be incorrect.

This ensures the Age feature is on a comparable scale with other numerical inputs. The comparison plots show that although the overall shape of the distribution remained unchanged, the values were successfully standardized, making the data more suitable for modeling.

A graph of a graph

AI-generated content may be incorrect.

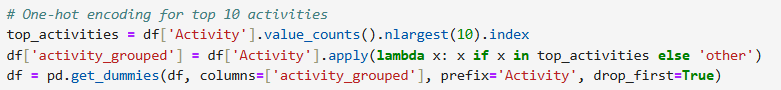
#### 2.4 Categorical encoding

To prepare categorical data for machine learning algorithms, both label encoding and one-hot encoding techniques were applied. The Sex column was binary encoded, mapping 'M' to 0 and 'F' to 1.

A close up of words

AI-generated content may be incorrect.

For the Type and Country columns, one-hot encoding was used to convert each category into a binary feature, excluding the first category to avoid multicollinearity. Also, the Activity column was simplified by grouping less frequent activities under an "other" category, and then one-hot encoded to capture only the most common 10 activities.



This step reduced dimensionality and ensured categorical data was in a machine-readable format, improving the performance and interpretability of models.

#### 2.5 Treating outliers

To address potential distortions caused by extreme values in the Age feature, I applied outlier treatment using the interquartile range (IQR) method. I calculated the first quartile (Q1) and third quartile (Q3) of the Age distribution, then defined any values lying outside the range [Q1 - 1.5×IQR, Q3 + 1.5×IQR] as outliers. These rows were subsequently removed from the dataset.

A screenshot of a computer

AI-generated content may be incorrect.

The distribution plot on the left clearly shows a narrower and more symmetric distribution of age values after removing outliers, centering around 24–25 years. The boxplot on the right further confirms that most data points now lie within the expected range, with only a few moderate outliers remaining on both ends.

A graph of two people

AI-generated content may be incorrect.

#### 2.6 Visualizing distribution after data preprocessing

To gain a deeper understanding of the cleaned dataset, various visualizations were generated for both numerical and categorical features. The distributions of Age, Year, and Age\_scaled revealed a highly concentrated range, with most shark incident victims being in their early 20s.

A graph of distribution of age and distribution of age

AI-generated content may be incorrect.

The encoded variable Sex\_encoded clearly reflected a male-dominant distribution.

A comparison of a graph

AI-generated content may be incorrect.

Among categorical features, activities like **surfing** and **swimming** turned out to be the most common contexts of shark encounters, while **white**, **tiger**, and **bull sharks** were the most frequently involved species.

A graph of different types of objects

AI-generated content may be incorrect.

Additionally, the Fatal outcome distribution showed that most shark attacks were non-fatal, highlighting a class imbalance that could influence modeling later.

A graph with a number of blue squares

AI-generated content may be incorrect.

These visualizations helped confirm the effectiveness of preprocessing and highlighted important patterns in the dataset.

## ***3. Feature Engineering***

#### 3.1 Engineered Features

In the feature engineering phase, several new features were created to enrich the dataset and enhance model performance:

1. **is\_water\_activity**: This binary feature indicates whether the activity involved direct interaction with water (for example, swimming, surfing, diving). It was derived by checking for the presence of common water-related keywords in the Activity column.



1. **season**: To capture potential seasonal patterns in shark attacks, a new categorical feature was created based on the month extracted from the Date column. It was then one-hot encoded into dummy variables like season\_spring, season\_summer, and so on.

A screen shot of a computer code

AI-generated content may be incorrect.

1. **age\_group**: The continuous Age feature was grouped into broader categories: child, teen, adult, and senior. This transformation helps identify age-based risk patterns. These categories were also one-hot encoded for modeling.

A screenshot of a computer program

AI-generated content may be incorrect.

1. **is\_extreme\_sport**: This binary feature flags whether the recorded activity involved extreme sports such as surfing, kiteboarding, wakeboarding, or spearfishing. The idea is to test if high-risk recreational behavior correlates with fatal outcomes.



#### 3.2 Correlation Heatmap

The correlation heatmap reveals that none of the numerical or engineered features show a strong linear relationship with the target variable Fatal. The feature with the strongest correlation to Fatal is is\_extreme\_sport at -0.26, indicating a slight inverse relationship, but this should not be overinterpreted. The highest correlation overall is between is\_water\_activity and is\_extreme\_sport (0.36), which makes sense as many extreme sports take place in water. Year has a weak negative correlation with Fatal (-0.14), possibly reflecting a slight decline in fatal cases over time. As expected, Age and Age\_scaled are perfectly correlated (1.00), so only one should be retained during modeling. Overall, the lack of strong correlations suggests that fatality outcomes may be influenced more by complex interactions or categorical/contextual features rather than individual numeric predictors alone.

A screenshot of a computer screen

AI-generated content may be incorrect.

#### 3.3 Feature Selection

To identify the most relevant predictors for fatal shark attacks, I employed a Random Forest Classifier for feature importance analysis. First, all non-numeric and datetime columns were excluded to ensure compatibility with the model.

A screenshot of a computer code

AI-generated content may be incorrect.

After training the classifier, I extracted and ranked feature importances. Year and Age\_scaled emerged as the top two predictors, followed by features such as Country\_usa and is\_extreme\_sport, confirming the significance of both temporal and behavioral factors. Engineered feature is\_extreme\_sport ranks 4th most important in predicting fatal shark attacks. This suggests that involvement in high-risk activities such as surfing, diving, kiteboarding, or spearfishing is a significant predictor of fatal outcomes. This insight supports the idea that extreme sports increase exposure to high-risk environments, making this a valuable feature for the model.

A graph with blue bars

AI-generated content may be incorrect.

To refine the model and reduce dimensionality, I visualized cumulative importance and selected only the features that together contribute to 90% of the total importance. This resulted in 46 selected features.

A graph with a line

AI-generated content may be incorrect.

#### 3.4 Handling class imbalance

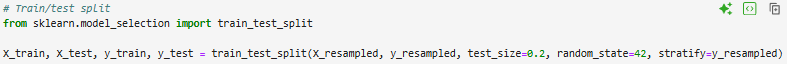
To address class imbalance in the target variable (Fatal), I applied SMOTE (Synthetic Minority Over-sampling Technique). Initially, the dataset had significantly more non-fatal cases than fatal ones, which can bias the model toward the majority class. SMOTE helps by generating synthetic examples of the minority class (fatal attacks), effectively balancing the dataset. This resampling ensures that the model receives enough signals from both classes during training, improving its ability to detect fatal cases accurately.

A computer code with black and green text

AI-generated content may be incorrect.

#### 3.5 Train/test split

To evaluate model performance, I split the balanced dataset into training and testing sets using an 80/20 ratio. I used the train\_test\_split function with stratify=y\_resampled to ensure that both the training and testing sets maintain the same class balance as the resampled data. This stratification is especially important after applying SMOTE, as it preserves the distribution of fatal and non-fatal cases in both subsets. The resulting training set is used for model learning, while the test set is reserved for unbiased performance evaluation.



## ***4. Model Training and Evaluation***

I trained three powerful ensemble-based classifiers: Random Forest, Gradient Boosting, and XGBoost.



1) **Random Forest** is an ensemble method that builds multiple decision trees and combines their predictions. It is known for being robust to overfitting, especially when working with high-dimensional data or many categorical features (like one-hot encoded variables). I chose Random Forest because it performs well overall and provides built-in feature importance, which was also helpful during the feature selection step.

2) **Gradient Boosting** builds decision trees sequentially, where each new tree corrects the errors made by the previous ones. This approach tends to produce high-performing models. I included Gradient Boosting due to its ability to capture complex patterns and interactions between features, which are expected in this context involving various activities, locations, and conditions of shark attacks.

3) **XGBoost** (Extreme Gradient Boosting) is an optimized and regularized version of Gradient Boosting, offering faster performance and often superior accuracy. It is especially effective for structured datasets. I chose XGBoost for its scalability, advanced regularization (to avoid overfitting), and strong performance with imbalanced datasets when paired with techniques like SMOTE.

#### **4.1 Random Forest**

The Random Forest model demonstrated strong performance on the test data, achieving an overall accuracy of 88%. Both precision and recall are well-balanced across the two classes (fatal and non-fatal), with precision at 0.88 for class 0 (non-fatal) and 0.87 for class 1 (fatal), and recall values of 0.87 and 0.88 respectively. This balance is reflected in the F1-score, which is 0.87 for class 0 and 0.88 for class 1, indicating the model is equally effective at minimizing false positives and false negatives.

Moreover, 5-fold cross-validation displayed a strong and consistent average accuracy of 0.8413 ± 0.0099, suggesting that the model generalizes well and is not overfitting. Overall, Random Forest proved to be a reliable baseline model for predicting fatal shark attacks.

A blue squares with white text

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

#### **4.2 Gradient Boosting**

The Gradient Boosting model demonstrates solid classification performance, although slightly behind Random Forest. The confusion matrix shows:

* True Negatives (TN): 484
* False Positives (FP): 118
* False Negatives (FN): 84
* True Positives (TP): 518

These results indicate the model predicts fatal and non-fatal outcomes fairly well, though it slightly overpredicts non-fatal cases (more false positives than false negatives).

A screenshot of a graph

AI-generated content may be incorrect.

Performance evaluation scores:

* Precision: 0.85 for non-fatal (0), 0.81 for fatal (1)
* Recall: 0.80 for non-fatal, 0.86 for fatal
* F1-score: Balanced around 0.83–0.84, with overall accuracy of 0.83
* Macro/Weighted Avg**:** Both at 0.83, showing consistent performance across classes

A screenshot of a computer screen

AI-generated content may be incorrect.

The 5-Fold Cross-Validation Accuracy of 0.8241 ± 0.0107 confirms that the model is stable and generalizes reasonably well. Overall, Gradient Boosting shows strong recall on fatal cases, which is valuable in scenarios where predicting fatal outcomes correctly is more critical.

#### **4.3 XGBoost**

The XGBoost classifier demonstrated strong performance in predicting fatal shark attacks, achieving an overall accuracy of 84%. The confusion matrix shows that the model correctly predicted:

* 493 out of 602 non-fatal cases (class 0), and
* 520 out of 602 fatal cases (class 1),  
  indicating a good balance between sensitivity and specificity.

A blue squares with white numbers

AI-generated content may be incorrect.

Its precision and recall scores were well-balanced across both classes:

* Precision: 0.86 (non-fatal), 0.83 (fatal)
* Recall: 0.82 (non-fatal), 0.86 (fatal)

A screenshot of a computer

AI-generated content may be incorrect.

The f1-score for both classes was 0.84, which confirms that the model performs consistently in terms of precision and recall. Moreover, the 5-fold cross-validation accuracy was 0.8398 ± 0.0089, indicating stable performance across different data splits.

Compared to other models, XGBoost provided a strong combination of generalization ability and class-wise balance, making it a reliable candidate for final deployment.

## ***5. Hyperparameter Tuning***

After evaluating all three models, Random Forest was selected as the best-performing model due to its strong balance of precision, recall, and f1-score across both classes. To further enhance its performance, hyperparameter tuning was conducted using GridSearchCV with 5-fold cross-validation, optimizing for the f1-score. A comprehensive parameter grid was defined, exploring combinations of n\_estimators, max\_depth, min\_samples\_split, min\_samples\_leaf, and max\_features. The best model configuration identified was:

A screenshot of a computer code

AI-generated content may be incorrect.

When evaluated on the test set, the tuned model achieved an impressive accuracy of 87%, with an f1-score of 0.87 for both fatal and non-fatal classes, as shown in the classification report. The confusion matrix confirmed this with 502 correct predictions for non-fatal cases and 544 for fatal cases. The 5-fold cross-validation f1-score on the training data was 0.8506 ± 0.0142, indicating consistent performance and generalization ability. This fine-tuned Random Forest model is therefore well-suited for predicting the fatality of shark attacks.

A screenshot of a computer screen

AI-generated content may be incorrect.

## ***Conclusion***

In this project, I explored a real-world shark attack dataset and built a classification model to predict whether an attack would be fatal. Through extensive preprocessing — including missing value handling, outlier removal, encoding of categorical variables, and feature engineering — I prepared a clean dataset suitable for modeling.

Feature importance analysis using a Random Forest model revealed that variables such as Year, Age, extreme sport involvement, and geographic or seasonal context play significant roles in fatality prediction. We addressed class imbalance using SMOTE and ensured robust evaluation through a 5-fold cross-validation strategy.

Among the models trained — Random Forest, Gradient Boosting, and XGBoost — Random Forest demonstrated the best performance with an F1-score of 0.87 on the test set. Further hyperparameter tuning improved its predictive accuracy even more, confirming its reliability and generalization.

The model revealed several valuable insights. Fatal shark attacks have slightly declined over time, as indicated by the importance of the "Year" feature. Age also emerged as a strong predictor, with younger individuals showing varied risk levels. Notably, participation in extreme sports like surfing and diving significantly increased the likelihood of fatal outcomes. Seasonal and geographical patterns also played a role, with certain countries and seasons showing higher fatality rates. These insights can support more targeted awareness and prevention strategies.