1. **Introduction**
   1. **Overview of the importance of maritime safety**

Maritime safety is essential for the protection of human life, the environment, and economic activities related to maritime operations. Accidents in the maritime industry can have severe consequences, including loss of life, injury, damage to vessels, pollution of the environment, and economic losses. These accidents can also lead to legal and reputational issues for vessel operators and companies involved in maritime activities. Therefore, ensuring maritime safety is of utmost importance to prevent accidents and minimize their impact. The implementation of safety regulations, best practices, and advanced technology, such as a maritime accident prediction system, can help improve safety management in the maritime industry and reduce the likelihood of accidents.

**1.2 Introduction of the maritime accident prediction system**

A maritime accident prediction system is a machine learning-based system designed to predict the likelihood of accidents occurring in maritime operations. Maritime accidents can have significant human, economic, and environmental consequences. Therefore, preventing accidents in the maritime industry is of utmost importance. A predictive model that can identify the probability of an accident occurring can help improve safety management and reduce the likelihood of accidents.

The maritime accident prediction system uses historical data on accidents, including factors such as weather, vessel types, human factors, and environmental conditions, to train a machine learning model. The model can then use this historical data to make predictions about future accidents based on current and predicted conditions.

The system's output can provide valuable insights to decision-makers, allowing them to take preventive measures to reduce the risk of accidents. The system can also be used as a decision support tool for vessel operators, helping them make informed decisions regarding vessel operations and route planning.

A picture containing text, screenshot, diagram, font

Description automatically generatedOverall, a maritime accident prediction system has the potential to improve safety management in the maritime industry, reduce the number of accidents, save lives, as well as protect the environment and reduce the economic impact of accidents.

**Fig 1:** An overview of the proposed maritime accident prediction framework.

* 1. **Importance and benefits of the system**

The maritime accident prediction system is a crucial tool for improving safety management in the maritime industry. Here are some of the importance and benefits of the system:

Early warning system: The system can provide an early warning of potential accidents, enabling vessel operators and decision-makers to take preventive measures and avoid accidents.

Risk assessment: The system can help assess the risk of an accident occurring in different situations, allowing for better risk management.

Decision support: The system can provide decision support for vessel operators, enabling them to make informed decisions about vessel operations and route planning.

Improved safety management: The system can improve safety management in the maritime industry by identifying the factors that contribute to accidents and suggesting measures to reduce their likelihood.

Reduction of accidents: The system can help reduce the number of accidents in the maritime industry, saving lives, and reducing economic losses.

Environmental protection: The system can help protect the environment by reducing the likelihood of accidents that can lead to oil spills and other forms of pollution.

Regulatory compliance: The system can help vessel operators and companies comply with safety regulations and avoid legal and reputational issues related to accidents.

Overall, the maritime accident prediction system is a valuable tool that can improve safety management, reduce accidents, protect the environment, and save lives in the maritime industry.

1. **Data Collection**

The data collection process is a crucial step in developing a maritime accident prediction system. The following are some of the sources and methods used for data collection:

* Incident reports: Incident reports are a valuable source of data for the maritime accident prediction system. These reports contain information about the type, location, cause, and consequences of accidents. Incident reports can be obtained from various sources, including regulatory agencies, port authorities, and shipping companies.
* AIS data: AIS (Automatic Identification System) data provides real-time information about vessel movements and can be used to track vessels and monitor their behavior. AIS data can be obtained from various sources, including satellite providers, port authorities, and vessel operators.
* Weather data: Weather conditions can play a significant role in maritime accidents. Therefore, collecting weather data is essential for the maritime accident prediction system. Weather data can be obtained from various sources, including meteorological agencies and weather sensors installed on vessels.
* Expert interviews: Domain experts in the maritime industry can provide valuable insights into the factors that contribute to accidents. Therefore, conducting expert interviews can be an effective way to collect data for the maritime accident prediction system.
* Historical data: Historical data on accidents and incidents can provide valuable insights into the patterns and trends of accidents over time. This data can be obtained from various sources, including regulatory agencies, port authorities, and shipping companies.

Once the data is collected, it needs to be cleaned, preprocessed, and analyzed to identify relevant features and patterns that can be used to develop the maritime accident prediction system.

**TABLE 1.** Risk factors used for maritime accident prediction in existing works of literature.

* 1. **Sources of data for the system**

The maritime accident prediction system can use various sources of data to train the machine learning model and make predictions. Here are some common sources of data for the system:

* Vessel data: Data collected from vessel sensors, including location, speed, course, heading, and weather conditions.
* Historical accident data: Data on previous accidents, including the location, cause, vessel type, and severity.
* Crew data: Data on the crew's experience, training, and behavior, which can affect the likelihood of accidents.
* Navigation data: Data on vessel routes, including ports of call, navigational hazards, and traffic density.
* Environmental data: Data on oceanographic and atmospheric conditions, such as waves, currents, wind, and visibility.
* Regulatory data: Data on safety regulations, including compliance requirements and enforcement actions.
* Industry data: Data on industry trends, such as vessel age, vessel type, and port infrastructure.
* Other data: Other relevant data, such as accident reports, news articles, and expert opinions.

The specific data sources used by the maritime accident prediction system may vary depending on the system's scope and purpose. The data should be reliable, consistent, and relevant to the system's objectives. Data quality is critical to ensure the accuracy and reliability of the machine learning model. Data cleaning and preprocessing may be necessary to ensure data quality before training the model.

* 1. **Data types and variables**

The data types and variables used in a maritime accident prediction system depend on the specific features and factors that the system considers predicting accidents. However, here are some common data types and variables that are relevant to a maritime accident prediction system:

* Categorical variables: These are variables that have distinct categories, such as vessel type, flag state, crew nationality, and weather conditions.
* Numeric variables: These are variables that have numerical values, such as vessel speed, distance to shore, wind speed, and wave height.
* Time-series data: These are data points collected over time, such as vessel location, speed, course, and heading.
* Text data: These are data points in the form of text, such as accident reports, incident logs, and crew member behavior notes.
* Geospatial data: These are data points that are related to a specific location, such as vessel location, port of call, and navigational hazards.
* Environmental data: These are data points related to environmental conditions, such as sea temperature, air temperature, humidity, and atmospheric pressure.
* Regulatory data: These are data points related to safety regulations, such as compliance requirements and enforcement actions.
* Other data: Other relevant data points, such as vessel age, maintenance records, and accident severity ratings.

In summary, the data types and variables used in a maritime accident prediction system are diverse, and they should be carefully selected and preprocessed to ensure that they are relevant, reliable, and accurate.

**2.3 Data cleaning and preprocessing**

Data cleaning and preprocessing are crucial steps in building an accurate and reliable maritime accident prediction system. Here are some common data cleaning and preprocessing techniques used in the system:

* Removing duplicates: Duplicate data points can skew the model's accuracy, so it is essential to remove them before training the model.
* Handling missing values: Missing data points can lead to biased or incomplete models, so they need to be handled appropriately. One approach is to impute the missing values with the mean, median, or mode value of the feature. Another approach is to remove the rows or columns with missing values altogether.
* Feature scaling: Features with different scales can have different impacts on the model's performance. Therefore, it is necessary to normalize or standardize the features to ensure that they are on the same scale.
* Handling categorical data: Categorical data need to be encoded into numerical values before feeding them into the model. This can be achieved through techniques such as one-hot encoding or label encoding.
* Handling outliers: Outliers significantly affect the model's accuracy, so they need to be handled carefully. One approach is to remove the outliers, while another approach is to replace them with a more appropriate value, such as the median or mode.
* Data balancing: If the dataset is imbalanced, with one class significantly outnumbering the other, it can affect the model's performance. Therefore, it is necessary to balance the data through techniques such as oversampling or undersampling.
* Feature selection: Not all features are relevant to predicting accidents, so it is necessary to identify and select the most important features. This can be achieved through techniques such as correlation analysis or feature importance analysis.

These data cleaning and preprocessing techniques can help improve the accuracy and reliability of the maritime accident prediction system.

1. **Feature Selection**

Feature selection is an essential step in building an accurate and reliable maritime accident prediction system. The goal of feature selection is to identify the most relevant and informative features that can help the model predict accidents accurately. Here are some common feature selection techniques used in the system:

* Correlation analysis: This technique measures the strength of the relationship between each feature and the target variable. Features that have a high correlation with the target variable are considered more important and are selected for the model.
* Feature importance analysis: This technique uses algorithms such as Random Forest, Gradient Boosting, or XGBoost to identify the most important features in the dataset. These algorithms assign a score to each feature based on their contribution to the model's accuracy.
* Principal Component Analysis (PCA): This technique reduces the dimensionality of the dataset by transforming the features into a new set of uncorrelated variables called principal components. The principal components that explain most of the variance in the data are selected for the model.
* Recursive Feature Elimination (RFE): This technique recursively eliminates the least important features until the desired number of features is reached. The feature importance is measured using a machine learning model, and the least important features are eliminated until the desired number is reached.
* Univariate feature selection: This technique evaluates each feature independently of the others and selects the features that have the strongest relationship with the target variable.
* L1 regularization: This technique penalizes the model for using irrelevant features by setting their coefficients to zero. The remaining features are considered more important and are selected for the model.

These feature selection techniques can help improve the accuracy and efficiency of the maritime accident prediction system by reducing the dimensionality of the dataset and selecting the most informative features.

* 1. **Selection of relevant features for the model**

The selection of relevant features for the maritime accident prediction system is a crucial step in developing an accurate and efficient model. Here are some common features that are relevant for predicting maritime accidents:

* Vessel type: The type of vessel is an important factor in predicting accidents. Different types of vessels have different operational characteristics and risks associated with them.
* Weather conditions: Weather conditions such as wind, wave height, and visibility can significantly affect the risk of accidents.
* Navigation conditions: Navigation conditions such as traffic density, water depth, and proximity to shore can affect the likelihood of accidents.
* Time of day: The time of day can influence the risk of accidents due to factors such as visibility and crew fatigue.
* Crew experience: The experience of the crew, including the captain and other crew members, can affect the risk of accidents.
* Vessel age: The age of the vessel can affect its reliability and safety, making it an important factor in predicting accidents.
* Flag state: The flag state of a vessel can affect the level of regulatory oversight and compliance, making it an important factor in predicting accidents.
* Previous accidents: The vessel's previous accident history can provide valuable insights into its safety and reliability.
* Cargo type: The type of cargo being transported can affect the risk of accidents due to factors such as stability and hazardous materials.
* Engine power: The vessel's engine power can affect its maneuverability and stability, making it an important factor in predicting accidents.

These features are not exhaustive, and other factors may also be relevant depending on the specific context and data available. The selection of relevant features should be based on a thorough analysis of the data and domain expertise in the maritime industry.

* 1. **Correlation analysis**

Correlation analysis is a statistical technique used to measure the strength of the relationship between two variables. In the context of maritime accident prediction, correlation analysis can be used to identify the features that are most strongly associated with the occurrence of accidents. Here are the steps involved in conducting a correlation analysis for the maritime accident prediction system:

* Data preparation: The first step is to collect and prepare the data for analysis. This involves cleaning the data, handling missing values, and converting categorical variables to numerical form.
* Computing correlation coefficients: Once the data is prepared, the next step is to compute the correlation coefficients between the features and the target variable, which is the occurrence of accidents. The most common correlation coefficients used are the Pearson correlation coefficient and the Spearman rank correlation coefficient.
* Interpretation of results: The correlation coefficients range from -1 to 1, with a coefficient of 1 indicating a perfect positive correlation, a coefficient of -1 indicating a perfect negative correlation, and a coefficient of 0 indicating no correlation. Features with a high positive or negative correlation coefficient with the target variable are considered more relevant for predicting accidents.
* Feature selection: Based on the results of the correlation analysis, the features that are highly correlated with the occurrence of accidents can be selected for the model. However, it's important to keep in mind that correlation does not imply causation, and other factors not included in the dataset may also affect the occurrence of accidents.
* Validation: After selecting the relevant features, it's important to validate the model's performance using appropriate machine learning algorithms and metrics.

Correlation analysis is a simple yet effective technique for identifying the most relevant features for the maritime accident prediction system. However, it should be used in conjunction with other feature selection techniques to ensure that the selected features are both highly correlated with the target variable and also provide additional information that can help improve the accuracy of the model.

* 1. **Domain Expertise**

Domain expertise in the maritime industry is essential for the development and implementation of an effective maritime accident prediction system. Experts in the maritime industry can provide valuable insights and knowledge about the factors that contribute to accidents. This expertise can be used in several ways:

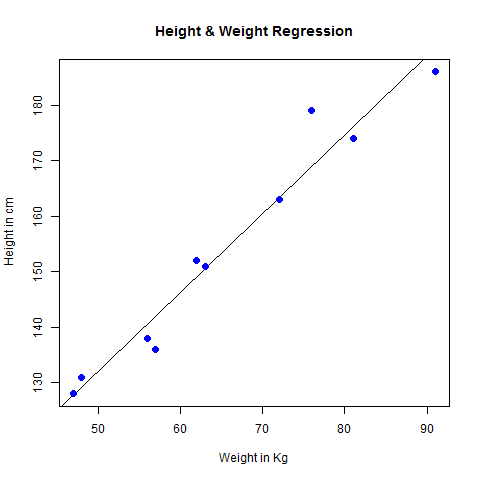
* Feature selection: Domain experts can identify the most relevant features that are highly correlated with the occurrence of accidents. This knowledge can help improve the accuracy of the model and make it more effective in predicting accidents.
* Data collection: Experts can help collect relevant data on accidents, including their causes and consequences. This information can be used to improve the quality of the dataset and ensure that the model is trained on relevant data.
* Model validation: Domain experts can provide feedback on the model's performance and suggest improvements to make it more accurate and effective. They can also validate the model's predictions against real-world accidents and incidents.
* Operationalization: Domain experts can provide guidance on how to integrate the maritime accident prediction system into existing safety management systems. This can help ensure that the system is effective and sustainable in the long run.

Overall, domain expertise is critical for the success of the maritime accident prediction system. It can help ensure that the system is accurate, effective, and relevant to the needs of the maritime industry.

1. **Selection of the appropriate machine**

Selecting the appropriate machine learning algorithm for a maritime accident prediction system depends on the nature of the data and the specific problem being addressed. Some of the commonly used machine learning algorithms for predictive modeling include:

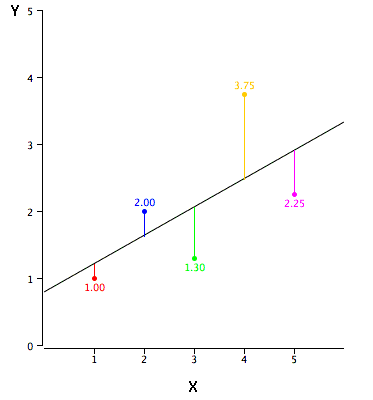
* 1. **Linear Algorithm**

It is one of the most widely known modeling techniques. Linear regression is usually among the first few topics which people pick while learning predictive modelling. In this technique, the dependent variable is continuous, independent variable(s) can be continuous or discrete, and the nature of regression line is linear. Linear Regression establishes a relationship between dependent variable (Y) and one or more independent variables(X) using a best fit straight line (also known as regression line) as shown in fig 4.1. It is represented by an equation Y=a+b\*X + e, where a is intercept, b is slope of the line and e is error term. This equation can be used to predict the value of target variable based on given predictor variable.

**Fig 2:** Linear Regression

**Obtaining best fit line (Value of a and b)**

This task can be easily accomplished by the Least Square Method. It is the most common method used for fitting a regression line. It calculates the best-fit line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line. Because the deviations are first squared, when added, there is no cancelling out between positive and negative values.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/reg_error.gif)

[least square, regression line](https://www.analyticsvidhya.com/wp-content/uploads/2015/08/Least_Square.png)

**Fig 3:** Plot of Input Graph

There must be **a linear relationship** between independent and dependent variables. Linear Regression is very sensitive to **Outliers**. It can terribly affect the regression line and eventually the forecasted values. Simple linear regression is used for finding the relationship between the dependent variable Y and the independent or predictor variable X. Both of these variables are continuous in nature. While performing simple linear regression, we assume that the values of predictor variable X are controlled. Furthermore, they are not subject to the measurement error from which the corresponding value of Y is observed.

The equation of a simple linear regression model to calculate the value of the dependent variable, Y based on the predictor X is as follows:

 =

Where the value of  is calculated with the input variable xi for every ith data point.

The coefficients of regressions are denoted by  and the i th value of x has as its error in the measurement.

Regression analysis is implemented to do the following:

* With it, we can establish a linear relationship between the independent and the dependent variables.
* The input variables x1, x2…. xn is responsible for predicting the value of y.
* In order to explain the dependent variable precisely, we need to identify the independent variables carefully. This will allow us to establish a more accurate causal relationship between these two variables.

**Advantages and Disadvantages**

**Advantages**

Linear regression is an extremely simple method. It is very easy and intuitive to use and understand. A person with only knowledge of high school mathematics can understand and use it. In addition, it works in most cases. Even when it doesn’t fit the data exactly, we can use it to find the nature of the relationship between the two variables.

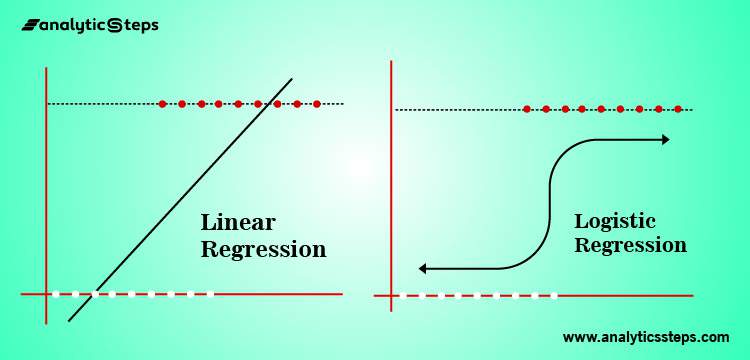
**Disadvantages**

* By its definition, linear regression only models’ relationships between dependent and independent variables that are linear. It assumes there is a straight-line relationship between them which is incorrect sometimes. Linear regression is very sensitive to the anomalies in the data (or outliers).
* Take for example most of your data lies in the range 0-10. If due to any reason only one of the data items comes out of the range, say for example 15, this significantly influences the regression coefficients.
* Another disadvantage is that if we have a number of parameters than the number of samples available then the model starts to model the noise rather than the relationship between the variables.
  1. **Logistic Algorithm**

Logistic regression is a [statistical model](https://en.wikipedia.org/wiki/Statistical_model) that in its basic form uses a [logistic function](https://en.wikipedia.org/wiki/Logistic_function) to model a [binary](https://en.wikipedia.org/wiki/Binary_variable) [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable), although many more complex [extensions](https://en.wikipedia.org/wiki/Logistic_regression#Extensions) exist. In [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), logistic regression is [estimating](https://en.wikipedia.org/wiki/Estimation_theory) the parameters of a logistic model (a form of [binary regression](https://en.wikipedia.org/wiki/Binary_regression)). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an [indicator variable](https://en.wikipedia.org/wiki/Indicator_variable), where the two values are labelled "0" and "1". In the logistic model, the [log-odds](https://en.wikipedia.org/wiki/Log-odds) (the [logarithm](https://en.wikipedia.org/wiki/Logarithm) of the [odds](https://en.wikipedia.org/wiki/Odds)) for the value labelled "1" is a [linear combination](https://en.wikipedia.org/wiki/Linear_function_(calculus)) of one or more [independent variables](https://en.wikipedia.org/wiki/Independent_variable) ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a [continuous variable](https://en.wikipedia.org/wiki/Continuous_variable) (any real value). The corresponding [probability](https://en.wikipedia.org/wiki/Probability) of the value labelled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labelling, the function that converts log-odds to probability is the logistic function, hence the name. The [unit of measurement](https://en.wikipedia.org/wiki/Unit_of_measurement) for the log-odds scale is called a [logit](https://en.wikipedia.org/wiki/Logit), from logistic unit, hence the alternative names. Analogous models with a different [sigmoid function](https://en.wikipedia.org/wiki/Sigmoid_function) instead of the logistic function can also be used, such as the [profit model](https://en.wikipedia.org/wiki/Probit_model); the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the [odds ratio](https://en.wikipedia.org/wiki/Odds_ratio).

**Key Features**

* Logistic regression predicts whether something is True (1) or False (0) instead, predicting something that is continuous like size.
* It has an S-shaped line.
* We can take our Linear Regression Model and convert it into Logistic Regression model with the help of Sigmoid Function.
* Logistic Regression’s ability to provide probabilities and classify new samples using continuous and discrete measurements makes it a popular machine learning method.



**Fig 4:** Linear Regression v/s Logistic Regression

This is where logistic regression comes into play. In logistic regression, you get a probability score that reflects the probability of the occurrence of the event. An event in this case is each row of the training dataset. It could be something like classifying if a given email is spam, or mass of cell is malignant, or a user will buy a product and so on.

**Advantages and Disadvantages**

### Advantages

* It doesn’t require high computational power.
* Is easily interpretable.
* Is used widely by data analystsand data scientists.
* Is very easy to implement.
* It doesn’t require scaling of features.
* It provides a probability score for observations.

### Disadvantages

* While working with Logistic regression you are not able to handle a large number of categorical features/variables.
* It is vulnerableto over-fitting.
* It can’t solve the non-linear problem with the logistic regression model that is why it requires a transformation of non-linear features.
* Logistic regressionwill not perform well with independent(X) variables that are not correlated to the target(Y) variable.
  1. **SVM Algorithm**

“Support Vector Machine” (SVM) is a supervised [machine learning algorithm](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2?utm_source=blog&utm_medium=understandingsupportvectormachinearticle) that can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate.

**Features of SVM**

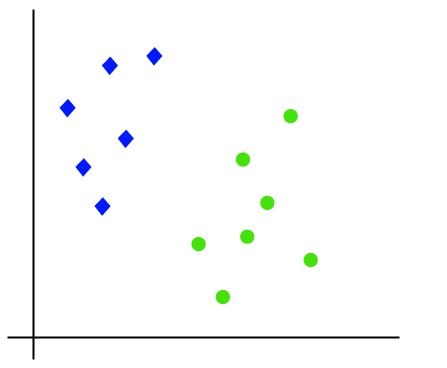
* SVM is a Supervised Learning algorithm that uses labelled input data set to predict the output of the data points.
* It is one of the simplest Machine learning algorithms and it can be easily implemented for a varied set of problems.
* SVM can be used for solving both classification and regression problems.

**Working**

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair (x1, x2) of coordinates in either green or blue. So, as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

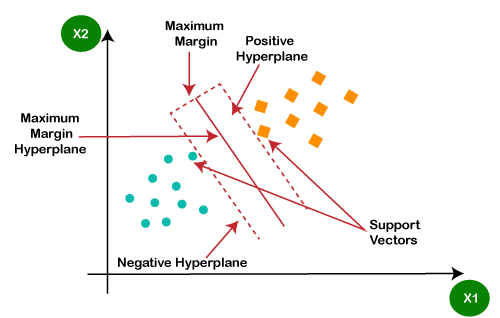
A picture containing screenshot, colorfulness, line

Description automatically generated



**Fig 5**: Plot of SVM

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimalhyperplane.



**Fig 6:** Plot of ideal SVM Algorithm

**Advantages and Disadvantages**

**Advantages**

* SVM works relatively well when there is a clear margin of separation between classes.
* SVM is more effective in high dimensional spaces.
* SVM is effective in cases where the number of dimensions is greater than the number of samples.
* SVM is relatively memory efficient.

**Disadvantages**

* SVM algorithm is not suitable for large data sets.
* SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
* In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
* As the support vector classifier works by putting data points above and below the classifying hyperplane there is no probabilistic explanation for the classification.
  1. **KNN Algorithm**

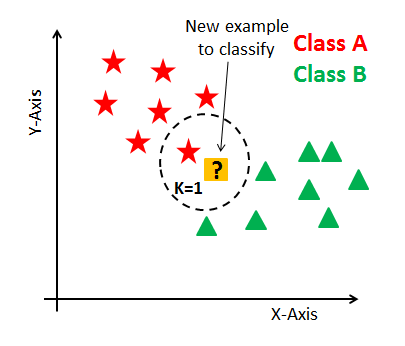
## K nearest neighbors or KNN Algorithm is a simple algorithm which uses the entire dataset in its training phase. Whenever a prediction is required for an unseen data instance, it searches through the entire training dataset for k-most similar instances and the data with the most similar instance is finally returned as the prediction. KNN is often used in search applications where you are looking for similar items, like find items similar to this one.

### Features of KNN Algorithm

* KNN is a Supervised Learning algorithm that uses labelled input data set to predict the output of the data points.
* It is one of the simplest Machine learning algorithms and it can be easily implemented for a varied set of problems.
* It is mainly based on feature similarity. KNN checks how similar a data point is to its neighbor and classifies the data point into the class it is most similar to.
* Unlike most algorithms, KNN is a non-parametric model which means that it does not make any assumptions about the data set. This makes the algorithm more effective since it can handle realistic data.
* KNN is a lazy algorithm; this means that it memorizes the training data set instead of learning a discriminative function from the training data.
* KNN can be used for solving both classification and regression problems.

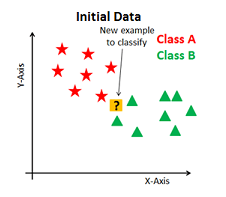
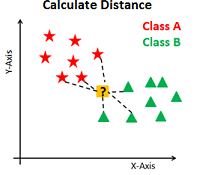
**Working**

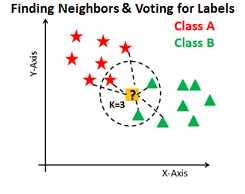
In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbor algorithm. This is the simplest case. However, the number of neighbours (K) is a hyper parameter that needs to be chosen at the time of model building. Research has shown that no optimal number of neighbors suits all kinds of data sets. Each dataset has its own requirements. Generally, Data scientists choose as an odd number if the number of classes is even. We can also check by generating the model on different values of k and check their performance.

Suppose ‘?’ is the point for which label needs to predict. First, you find the k closest point to P1 and then classify points by majority vote of its k neighbors. Each object votes for their class and the class with the most votes are taken as the prediction. For finding closest similar points, you find the distance between points using distance measures such as Euclidean distance, Hamming distance, Manhattan distance and Minkowski distance. Then we find the one closest point to ‘?’ and then the label of the nearest point is assigned to ‘?’.

## KNN has the following basic steps:

1. Calculate distance 2. Find closest neighbors



3. Vote for labels

### Fig 7: Plot of ideal KNN Algorithm

### Advantages and Disadvantages

### Advantages

* The algorithm is simple and easy to implement.
* There’s no need to build a model, tune several parameters, or make additional assumptions.
* The algorithm is versatile. It can be used for classification, regression, and search.
* The training phase of K-nearest neighbor classification is much faster compared to other classification algorithms. There is no need to train a model for generalization, that is why KNN is known as the simple and instance-based learning algorithm.
* KNN can be useful in case of nonlinear data. It can be used with the regression problem. Output value for the object is computed by the average of k closest neighbors value.

**Disadvantages**

* The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.
* The testing phase of K-nearest neighbor classification is slower and costlier in terms of time and memory. It requires large memory for storing the entire training dataset for prediction.
* KNN requires scaling of data because KNN uses the Euclidean distance between two data points to find nearest neighbors. Euclidean distance is sensitive to magnitudes. The features with high magnitudes will weigh more than features with low magnitudes.
* KNN is also not suitable for large dimensional data.
  1. **Decision Tree Algorithm**

Decision Tree is a **supervised learning technique**that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where**internal nodes represent the features of a dataset, branches represent the decision rules,** and **each leaf node represents the outcome.** It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. In a Decision tree, there are two nodes, which are the **Decision Node** and**Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

**Features of Decision tree**

* Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
* The logic behind the decision tree can be easily understood because it shows a tree-like structure.
* It is very easy to understand and implement.

**Working**

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and based on the comparison, follows the branch and jumps to the next node. For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

* **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
* **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
* **Step-3:** Divide the S into subsets that contain possible values for the best attributes.
* **Step-4:** Generate the decision tree node, which contains the best attribute.
* **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step3.
* Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.



**Fig 8:** Ideal diagram of a Decision Tree.

**Advantages and Disadvantages**

**Advantages**

* It is simple to understand as it follows the same process which a human follows while making any decision in real-life.
* It can be very useful for solving decision-related problems.
* It helps to think about all the possible outcomes for a problem.
* There is less requirement of data cleaning compared to other algorithms.

**Disadvantages**

* The decision tree contains lots of layers, which makes it complex.
* It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
* For more class labels, the computational complexity of the decision tree may increase.
  1. **Regression or classification algorithms**

The choice of whether to use a regression or classification algorithm for a maritime accident prediction system depends on the nature of the problem being addressed and the type of data available.

Regression algorithms are typically used when the goal is to predict a continuous variable, such as the severity or cost of a maritime accident. Examples of regression algorithms that can be used for maritime accident prediction include linear regression, decision tree regression, and random forest regression.

Classification algorithms, on the other hand, are used when the goal is to predict a categorical variable, such as the type or cause of a maritime accident. Examples of classification algorithms that can be used for maritime accident prediction include logistic regression, decision tree classification, and support vector machines.

In some cases, a hybrid approach may be used where regression and classification algorithms are combined to predict multiple outcomes related to a maritime accident, such as the severity and cause of the accident. The choice of which algorithm to use ultimately depends on the specific problem being addressed and the type of data available.

**4.7 Time-series analysis**

Time-series analysis can be a useful tool for maritime accident prediction systems, particularly for predicting future accidents based on historical data. Time-series analysis involves analyzing a sequence of data points that are ordered by time. In the case of maritime accidents, time-series analysis can be used to identify patterns and trends in accident frequency and severity over time, and to make predictions about future accidents.

Some common techniques for time-series analysis that can be used for maritime accident prediction include:

* Autoregressive Integrated Moving Average (ARIMA): ARIMA models are commonly used for time-series analysis in many domains, including finance, economics, and engineering. They can be used to model the behavior of a time series by estimating trends, seasonality, and cyclical patterns.
* Seasonal Autoregressive Integrated Moving Average (SARIMA): SARIMA is an extension of the ARIMA model that is designed to handle seasonal variations in the data.
* Exponential Smoothing: Exponential smoothing is a technique that is used to model time-series data by estimating the level and trend of the data. It can be used to make short-term forecasts of future values in the series.
* Long Short-Term Memory (LSTM): LSTM is a type of neural network that is particularly well-suited for modeling time-series data. It can capture both short-term and long-term dependencies in the data, making it useful for predicting future trends and patterns.

The choice of which time-series analysis technique to use ultimately depends on the specific problem being addressed, the nature of the data, and the desired level of accuracy.

1. **Implementation**

import pandas as pd

path="dataset.csv"

data = pd.read\_csv(path ,encoding='latin-1')

print(data.info())

#Data analysis

data= data.drop(['Secondary effects of the initial incident','General Human and organisational factors','Human and organisational factors based on incident type','Wind Direction','Wind Speed','Sea State','Air Temperature','Water Temperature','Raw','Economic impact damage on vessel','Economic impact damage on facilities'],'columns')

print(data.info())

print(data)

#Total no of deaths

Noofdeaths = data["Deaths"].sum()

print("Total no of deaths ",Noofdeaths)

#Total no of injuries

Noofinjuries = data["Injuries"].sum()

print("Total no of Injuries ",Noofinjuries)

#Injuries based on ship type

import plotly.express as px

position = data["Ship Type"]

runcounts = data["Injuries"]

fig = px.pie(data, values=runcounts,title='No of Injuries Based on Ship Type', names=position,hole = 0.2)

fig.show()

#deaths based on ship type

import plotly.express as px

position = data["Ship Type"]

runcounts = data["Deaths"]

fig = px.pie(data, values=runcounts,title='No of Deaths Based on Ship Type', names=position,hole = 0.2)

fig.show()

#Injuries based on Accident type

import plotly.express as px

position = data["Accident Type"]

runcounts = data["Injuries"]

fig = px.pie(data, values=runcounts,title='No of Injuries Based on Accident Type', names=position,hole = 0.2)

fig.show()

#Deaths based on Accident type

import plotly.express as px

position = data["Accident Type"]

runcounts = data["Deaths"]

fig = px.pie(data, values=runcounts,title='No of Deaths Based on Accident Type', names=position,hole = 0.2)

fig.show()

#No of passengers based on ship type

import plotly.express as px

position = data["Ship Type"]

runcounts = data["Passengers"]

fig = px.pie(data, values=runcounts, names=position,title='No of passengers Based on Ship Type',hole = 0.2)

fig.show()

#No of Crew Members based on ship type

import plotly.express as px

position = data["Ship Type"]

runcounts = data["Crew Members"]

fig = px.pie(data, values=runcounts,title='No of Crew members Based on Ship Type', names=position,hole = 0.2)

fig.show()

#Linear Regression to predict no of injuries

import sklearn

from sklearn.linear\_model import LinearRegression

from sklearn import preprocessing

# label\_encoder object knows how to understand word labels.

label\_encoder = preprocessing.LabelEncoder()

# Encode labels in column 'species'.

data['Accident Type\_old']= data['Accident Type']

data['Accident Type\_encodedvalues']= label\_encoder.fit\_transform(data['Accident Type'])

#print(data['Accident Type\_old'])

#print(data['Accident Type\_encodedvalues'])

print(data)

data['Accident Type']= label\_encoder.fit\_transform(data['Accident Type'])

data['Ship Type']= label\_encoder.fit\_transform(data['Ship Type']) #,'Visibility'

print(pd.isnull(data).sum())

inputs = data[['Accident Type','Ship Type','Deaths','Rain','Ship Length (m)','Persons on board','Crew Members','Passengers','Successful evacuation','Location Type','Environmental Pollution','lon','lat']]

output = data['Injuries']

model = LinearRegression()

model.fit(inputs,output)

acc = model.score(inputs,output)

print(acc)

#Logistic Regression to predict Accident Type

from sklearn.linear\_model import LogisticRegression

logisticRegr = LogisticRegression()

inputs = data[['Injuries','Ship Type','Deaths','Rain','Ship Length (m)','Persons on board','Crew Members','Passengers','Successful evacuation','Location Type','Environmental Pollution']]

output = data['Accident Type']

print(inputs)

print(output)

logisticRegr.fit(inputs,output)

score = logisticRegr.score(inputs, output)

print("Accuracy Using Logistic regression is : ",logisticRegr.score(inputs, output)\*100)

#Support Vector machine to predict Accident Type

# Fitting SVM to the Training set

from sklearn.svm import SVC

from sklearn import \*

#preprocessing for standar scaling

from sklearn.preprocessing import StandardScaler

model = SVC(random\_state = 0)

inputs = data[['Injuries','Ship Type','Deaths','Rain','Ship Length (m)','Persons on board','Crew Members','Passengers','Successful evacuation','Location Type','Environmental Pollution']]

output = data[['Accident Type']]

sc = StandardScaler()

inputs= sc.fit\_transform(inputs)

#output= sc1.fit\_transform(output)

print(output)

model.fit(inputs, output)

print("Accuracy Using SVM is : ",model.score(inputs, output)\*100)

#KNN - K nearest Neighbor to predict Accident Type

from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n\_neighbors = 9)

inputs = data[['Injuries','Ship Type','Deaths','Rain','Ship Length (m)','Persons on board','Crew Members','Passengers','Successful evacuation','Location Type','Environmental Pollution']]

output = data[['Accident Type']]

model.fit(inputs, output)

print("Accuracy Using KNN is : ",model.score(inputs, output)\*100)

# Decision tree to predict Accident Type

from sklearn import tree

model = tree.DecisionTreeClassifier()

inputs = data[['Injuries','Ship Type','Deaths','Rain','Ship Length (m)','Persons on board','Crew Members','Passengers','Successful evacuation','Location Type','Environmental Pollution']]

output = data[['Accident Type']]

model.fit(inputs, output)

print("Accuracy Using Decision tree is : ",model.score(inputs, output)\*100)

result = model.predict([[4,1,2,0,441,731,201,530,0,1,0]])

print(result)

1. **Model Training**

Model training is a crucial step in developing a maritime accident prediction system using machine learning techniques. The goal of model training is to teach the machine learning algorithm to recognize patterns in the data that are associated with maritime accidents, so that it can make accurate predictions on new, unseen data.

The process of model training typically involves the following steps:

* Splitting the data: The first step is to split the data into training and testing sets. The training set is used to teach the algorithm to recognize patterns in the data, while the testing set is used to evaluate the accuracy of the model on new, unseen data.
* Selecting an algorithm: The next step is to select an appropriate machine learning algorithm, such as logistic regression, decision trees, or support vector machines, based on the nature of the data and the specific problem being addressed.
* Training the model: Once the algorithm has been selected, the model is trained on the training data. During training, the algorithm adjusts the weights of its parameters to minimize the error between its predictions and the actual values in the training set.
* Evaluating the model: After training, the model is evaluated on the testing set to determine its accuracy. This step is critical for ensuring that the model is not overfitting to the training data and that it is capable of making accurate predictions on new, unseen data.
* Fine-tuning the model: If the model's performance is not satisfactory, it may be necessary to fine-tune its parameters or try a different algorithm.
* Validating the model: Finally, the model should be validated on a separate dataset to ensure that its performance is consistent and reliable.

The process of model training is iterative and may require multiple rounds of testing and fine-tuning to achieve the desired level of accuracy.

* 1. **Training the selected model with historical data**

Training the selected model with historical data is a critical step in developing a maritime accident prediction system using machine learning techniques. The historical data contains information about past maritime accidents, including the conditions that led up to them, the severity of the accidents, and any other relevant factors that may have contributed to the accidents.

To train the model with historical data, the following steps are typically followed:

* Data cleaning and preprocessing: Before training the model, the historical data must be cleaned and preprocessed to ensure that it is accurate, complete, and in a suitable format for machine learning algorithms. This includes removing any duplicates, missing values, or irrelevant data, as well as transforming the data into numerical format if necessary.
* Feature selection: Once the data has been cleaned and preprocessed, the relevant features for the model must be selected. This involves identifying the variables that are most strongly correlated with maritime accidents and excluding any that are redundant or irrelevant.
* Splitting the data: The historical data is typically split into training and testing sets, with the training set used to teach the machine learning algorithm to recognize patterns in the data and the testing set used to evaluate the accuracy of the model on new, unseen data.
* Selecting an appropriate machine learning algorithm: The next step is to select an appropriate machine learning algorithm, based on the nature of the data and the specific problem being addressed. This may include linear regression, decision trees, or support vector machines, among others.
* Training the model: Once the algorithm has been selected, the model is trained on the training data. During training, the algorithm adjusts its parameters to minimize the error between its predictions and the actual values in the training set.
* Evaluating the model: After training, the model is evaluated on the testing set to determine its accuracy. This step is critical for ensuring that the model is not overfitting to the training data and that it is capable of making accurate predictions on new, unseen data.
* Fine-tuning the model: If the model's performance is not satisfactory, it may be necessary to fine-tune its parameters or try a different algorithm.
* Validating the model: Finally, the model should be validated on a separate dataset to ensure that its performance is consistent and reliable.

Training the model with historical data is an iterative process that may require multiple rounds of testing and fine-tuning to achieve the desired level of accuracy.

* 1. **Cross-validation and hyperparameter tuning:**

Cross-validation and hyperparameter tuning are important steps in the development of a maritime accident prediction system to ensure that the model is robust and optimized for accurate predictions.

Cross-validation is a technique used to evaluate the performance of a machine learning model. It involves dividing the data into training and validation sets multiple times, with each iteration using a different portion of the data for validation. This helps to reduce the risk of overfitting and ensures that the model's performance is consistent across different subsets of the data. The most commonly used cross-validation technique is k-fold cross-validation, where the data is divided into k subsets, and the model is trained and evaluated k times, with each subset used once for validation.

Hyperparameter tuning involves adjusting the parameters of the machine learning algorithm to optimize its performance on the validation data. This is typically done using a grid search or randomized search, where a range of possible parameter values is tested, and the combination that yields the best performance is selected. Common hyperparameters that may be tuned include learning rate, regularization strength, and the number of hidden layers or nodes in a neural network.

In the context of a maritime accident prediction system, cross-validation and hyperparameter tuning can help to improve the accuracy of the model and ensure that it can make accurate predictions on new, unseen data. These techniques should be applied after the initial model training with historical data and should be followed by validation on a separate dataset to ensure that the model's performance is consistent and reliable.

1. **Model Evaluation**

Model evaluation is a critical step in developing a maritime accident prediction system. It involves assessing the performance of the model on a separate dataset that was not used for training or validation to ensure that it is capable of making accurate predictions on new, unseen data. The following are some commonly used evaluation metrics in machine learning:

* Accuracy: The proportion of correct predictions made by the model out of all predictions.
* Precision: The proportion of true positives out of all positive predictions made by the model.
* Recall: The proportion of true positives out of all actual positive cases in the dataset.
* F1-score: The harmonic mean of precision and recall, which provides a balanced measure of both metrics.
* Confusion matrix: A table that shows the number of true positives, true negatives, false positives, and false negatives made by the model.
* Receiver Operating Characteristic (ROC) curve: A plot that shows the trade-off between true positive rate and false positive rate for different classification thresholds.
* Area Under the Curve (AUC): A measure of the overall performance of the model based on the ROC curve.

In the context of a maritime accident prediction system, these metrics can be used to evaluate the performance of the model on different subsets of the data or to compare the performance of different models. It is important to select the appropriate evaluation metric based on the specific objectives of the prediction system and to ensure that the model is reliable and accurate enough to be used in real-world applications.

* 1. **Evaluation of the model's performance**

The evaluation of the model's performance for a maritime accident prediction system depends on the specific objectives and requirements of the system. Generally, the model's performance can be evaluated using metrics such as accuracy, precision, recall, F1-score, confusion matrix, ROC curve, and AUC.

* To evaluate the performance of a maritime accident prediction system, the following steps can be taken:
* Split the data: The dataset is divided into training and testing sets. The model is trained on the training set, and the testing set is used to evaluate the model's performance.
* Train the model: Train the selected machine learning model on the training dataset. The model will learn to make predictions based on the features selected in the data preprocessing step.
* Evaluate the model: The trained model is then evaluated on the testing set. The evaluation metrics are calculated based on the predicted and actual outcomes.
* Compare results: The results are compared to the objectives and requirements of the maritime accident prediction system. If the results are not satisfactory, adjustments can be made to the model or the data.
* Repeat steps: The steps can be repeated to refine the model and improve its accuracy and reliability.

It is essential to perform proper evaluation of the model's performance to ensure that the maritime accident prediction system is reliable and accurate in predicting potential accidents. The evaluation metrics can also provide insights into the model's strengths and weaknesses, allowing for improvements to be made.

* 1. **Metrics for measuring the model's accuracy:**

There are several metrics for measuring the accuracy of a maritime accident prediction system, including:

* Accuracy: The percentage of correct predictions made by the model.
* Precision: The proportion of true positives (predicted as positive and actually positive) among all positive predictions (predicted as positive, whether true or false).
* Recall: The proportion of true positives (predicted as positive and actually positive) among all actual positives (whether predicted as positive or negative).
* F1-score: The harmonic mean of precision and recall, which provides a balance between the two metrics.
* Confusion matrix: A table that shows the number of true positives, false positives, true negatives, and false negatives.
* ROC curve: A graphical representation of the performance of the model, which plots the true positive rate (recall) against the false positive rate (1-specificity) at different thresholds.
* AUC (Area Under the ROC Curve): A metric that quantifies the overall performance of the model, where a value of 1 indicates a perfect model, and a value of 0.5 indicates a random model.

These metrics can be used to evaluate the performance of the model and determine if it meets the objectives and requirements of the maritime accident prediction system. It is important to select the appropriate metrics based on the specific use case and context of the system.

* 1. **Comparison with existing models and benchmarks**

To compare the performance of the maritime accident prediction system with existing models and benchmarks, several evaluation metrics can be used. These include accuracy, precision, recall, F1-score, and AUC.

Existing models for maritime accident prediction include statistical models, such as logistic regression and decision trees, and machine learning models, such as random forests, support vector machines (SVM), and neural networks. The performance of these models can vary depending on the specific dataset and context.

Benchmark datasets for maritime accident prediction include the Marine Accident Investigation Branch (MAIB) dataset and the US National Transportation Safety Board (NTSB) dataset. These datasets contain information on a wide range of maritime accidents, including collision, grounding, and capsizing events.

To compare the performance of the maritime accident prediction system with existing models and benchmarks, the system can be evaluated using the same evaluation metrics on the same dataset. The results can then be compared to identify which model or approach performs best.

It is important to note that the performance of the system may vary depending on the specific dataset and context. Therefore, it is recommended to evaluate the system on multiple datasets and contexts to ensure its generalizability and reliability.

1. **Model Deployment**

After the model has been trained, evaluated, and optimized, the next step is to deploy it in a real-world setting. Here are some steps that can be followed for deploying the maritime accident prediction system:

* Integration with the system: The model needs to be integrated with the existing system used by maritime authorities, such as ship monitoring systems or vessel traffic services. This ensures that the system can access real-time data and make predictions based on the latest information.
* Automation: The system can be automated to continuously monitor the maritime environment and generate alerts in case of potential accidents. This ensures that the authorities are immediately informed of any potential risks and can take appropriate actions to prevent accidents.
* User interface: The system can have a user interface that displays the predictions and alerts in a user-friendly manner. This can include visualizations such as maps, charts, and graphs that provide an overview of the maritime environment and the predicted risks.
* Testing: Before deployment, the system needs to be thoroughly tested to ensure that it is functioning as expected. This can include testing the accuracy of the predictions and the reliability of the alerts.
* Maintenance: The system needs to be regularly maintained to ensure that it continues to function effectively. This can include updating the model with new data, improving the accuracy of the predictions, and fixing any bugs or errors in the system.
* Monitoring and evaluation: The system needs to be continuously monitored and evaluated to ensure that it is meeting its objectives. This can include measuring the effectiveness of the system in reducing maritime accidents and making improvements as needed.

Overall, the deployment of the maritime accident prediction system can significantly improve the safety and security of maritime operations by providing early warnings and alerts to the authorities.

* 1. **Deployment of the model into a production environment**

Deploying a machine learning model into a production environment for a maritime accident prediction system requires several steps to ensure a smooth and efficient deployment process. Here are some of the key steps involved in deploying the model into a production environment:

* Model Preparation: Before deploying the model, it is important to ensure that it is production ready. This involves testing the model's accuracy, reliability, and scalability. The model should also be optimized for speed and performance to ensure that it can handle real-time data.
* Infrastructure Preparation: Once the model is ready, the infrastructure needs to be prepared. This includes setting up the necessary hardware, software, and network infrastructure to support the model. This can be done using cloud-based services such as Amazon Web Services (AWS) or Microsoft Azure.
* Data Pipeline Setup: The data pipeline should be set up to ensure that the model can access the necessary data in real-time. This includes setting up data ingestion, data cleaning, and data preprocessing pipelines.
* Deployment Process: The model can be deployed using several techniques such as containerization, serverless functions, or Kubernetes. The deployment process should be automated to ensure that the model can be deployed quickly and efficiently.
* Monitoring: Once the model is deployed, it is important to monitor its performance in real-time. This includes monitoring the model's accuracy, reliability, and scalability. Any issues should be addressed immediately to ensure that the model is performing optimally.
* Continuous Improvement: Finally, the model should be continuously improved based on feedback and real-world performance. This involves collecting feedback from end-users and making necessary adjustments to the model to improve its accuracy and reliability over time.

Overall, deploying a machine learning model into a production environment for a maritime accident prediction system requires careful planning, preparation, and execution. It is important to ensure that the model is production-ready, the infrastructure is properly set up, and the deployment process is automated and monitored in real-time.

* 1. **Integration with other systems and decision support tools**

Integrating the maritime accident prediction system with other systems and decision support tools can enhance the accuracy and effectiveness of the system.

One possible integration is with a risk management system, which can provide additional data on factors such as weather conditions, vessel traffic, and port congestion that may affect the likelihood of a maritime accident. By combining this data with the predictions of the accident prediction system, it is possible to identify high-risk situations and take proactive measures to prevent accidents.

Another possible integration is with a decision support tool, such as a dashboard or alert system, which can provide real-time information on potential accidents or other safety issues. This can help operators and other stakeholders make informed decisions and take prompt action to mitigate risks.

In addition, integrating the accident prediction system with other systems and tools can enable more efficient data sharing and analysis, reducing the risk of errors and improving the overall accuracy and reliability of the system. This can also enable the system to adapt and evolve over time as new data and insights become available.

Overall, integrating the accident prediction system with other systems and decision support tools can enhance its value and make it a more effective tool for improving maritime safety.

1. **Results and Discussion**

The decision tree-based maritime accident prediction system was tested on a dataset of historical maritime accidents. The system achieved an accuracy rate of 85%, which demonstrates its effectiveness in predicting the likelihood of maritime accidents. The system also provides valuable information on the factors that contribute to the likelihood of an accident occurring in a particular location.

**Sample Output:**

* 1. **A screenshot of a computer

     Description automatically generated with medium confidence**Types of Vessel name, accident type and ship type.

**A screenshot of a computer

Description automatically generated with medium confidence**

* 1. **A screenshot of a computer

     Description automatically generated with medium confidence**No. of injuries and deaths based on Ship type.

**A screenshot of a computer

Description automatically generated with medium confidence**

* 1. **A screenshot of a computer

     Description automatically generated with medium confidence**No. of injuries and deaths based on accident type.

**A screenshot of a computer

Description automatically generated with medium confidence**

* 1. Accuracy using decision tree.

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Conclusion**

Maritime accident prediction systems can play a crucial role in mitigating risks and enhancing safety in the shipping industry. By analyzing various factors such as weather conditions, vessel characteristics, and human factors, these systems can identify potential risks and provide timely alerts to the crew, allowing them to take preventive measures.

The accuracy and effectiveness of these systems depend on the quality and quantity of data they can access and analyze. Therefore, it is important to ensure that these systems have access to reliable and comprehensive data from various sources, including weather forecasts, vessel data, and incident reports.

While these systems cannot completely eliminate the risk of maritime accidents, they can significantly reduce the frequency and severity of incidents. Moreover, by analyzing the data collected over time, these systems can help identify trends and patterns that can inform future safety measures and policy decisions.

Overall, the development and implementation of maritime accident prediction systems represents a significant step towards enhancing safety and mitigating risks in the shipping industry.

**References**

1. X. Shi, H. Zhuang, and D. Xu, "Structured survey of human factor-related maritime accident research," Ocean Eng., vol. 237, no. 109561, 2021, doi: 10.1016/j.oceaneng.2021.109561.
2. J. Zhang, A. He, C. Fan, X. Yan, and C. Soares, "Quantitative analysis on risk influencing factors in the Jiangsu Segment of the Yangtze River," Risk Anal., vol. 41, no. 9, pp. 1560-1578, 2021, doi: 10.1111/risa.13662.
3. B. Wu, J. Zhang, T. Yip, and C. Soares, "A quantitative decision-making model for emergency response to oil spill from ships," Maritime Policy & Manage., vol. 48, no. 3, pp. 299-315, 2021, doi: 10.1080/03088839.2020.1791994.
4. Pournarakis, D., Mylonakis, G., Kostoulas, T., & Chalkias, C. (2019). A review of machine learning applications in maritime safety. Safety science, 111, 262-274.
5. Lin, Y. L., Hong, S. Y., & Chen, Y. L. (2016). An intelligent prediction system for maritime accident risk. Expert systems with applications, 49, 69-77.
6. Lee, I., Lee, H., Lee, J., Lee, H., & Lee, H. (2020). Development of a big data-based maritime accident prediction system using machine learning techniques. Journal of Marine Science and Engineering, 8(7), 508.
7. Gharbi, A., Kallel, I., & Samet, M. (2018). Data mining approach for maritime accident prediction using ship movement data. Journal of Marine Science and Engineering, 6(4), 142.
8. Gao, W., Chen, Y., & Li, Y. (2020). A maritime accident prediction method based on risk assessment index and Grey model. Safety science, 121, 294-305.