applied sciences

MDPI

Article

Application of Machine Learning for Predictive Analysis and Management of Mediterranean-Farmed Fish Mortalities: A Risk Management Case Study Using Apache Spark

Marios C. Gkikas [1,2], Dimitris C. Gkikas [3,*], Gerasimos Vonitsanos [4], John A. Theodorou [3] and Spyros Sioutas [4]

[1] OWEB Digital Experience, 302 00 Mesolonghi, Greece; info@oweb.gr
[2] Department of Management Science and Technology, School of Economics and Business Administration, University of Patras, 265 04 Patras, Greece
[3] Department of Fisheries and Aquaculture, School of Agricultural Sciences, University of Patras, 302 00 Mesolonghi, Greece; jtheo@upatras.gr
[4] Department of Computer Engineering and Informatics, School of Engineering, University of Patras, 265 04 Patras, Greece; mvonitsanos@ceid.upatras.gr (G.V.); sioutas@ceid.upatras.gr (S.S.)
* Correspondence: dgkikas@upatras.gr

Featured Application: Machine learning model applications provide enough evidence to predict fish stock using aquaculture data, allowing for significant practical implications. By applying advanced statistical models, including Random Forest, Decision Tree, and Linear Regression, aquaculture stakeholders can improve their decision-making processes by predicting fish populations, mortality rates, and other critical metrics. Among the predictive models, the Random Forest model outperformed other models, reaching the highest accuracy. This allows more efficient and productive management strategies, supports sustainable development, and helps policymakers in enhancing decision-making to protect marine ecosystems. The dynamic for integrating such tools, such as predictive statistical models into real-time monitoring systems, provides a proactive strategy for aquaculture growth, risk mitigation, and long-term sustainability.

Abstract: The current study evaluates the performance of three machine learning models—Decision Trees, Random Forest, and Linear Regression—applied to aquaculture data to mitigate risks in aquaculture management. The performances of these models are analyzed and properly demonstrated using metrics including the Mean Squared Error (MSE), R-squared (R^2), Root Mean Squared Error (RMSE), and Concordance Index (C-index). The Random Forest model achieved the highest prediction accuracy among all machine learning models, followed by Linear Regression and the Decision Trees. The scatter plot for Linear Regression demonstrates good predictive accuracy for mid-range values. However, it shows significant deviations at the extremes, indicating that the model struggles to capture the full range of variability in the data. The bar chart of coefficients pinpoints the variables with the greatest impact on the predictions, providing suggestions for potential areas that can be improved and providing model interpretability. Future work could incorporate more predictive statistics models focusing on improving the models for extreme values by assessing non-linear models, feature engineering methods, and expanding research into less influential variables. The results greatly impact several sections, including aquaculture management, policy-making, and operational strategies, providing valuable insights for stakeholders and decision-makers. Apache Spark was used for data processing and machine learning model implementation; Apache Cassandra was also used for data storage, ensuring efficient large dataset management and SQL tools for structured data handling; Oracle VM VirtualBox for cross-platform virtualization; and Spark Connector was also used.

check for updates

Citation: Gkikas, M.C.; Gkikas, D.C.; Vonitsanos, G.; Theodorou, J.A.; Sioutas, S. Application of Machine Learning for Predictive Analysis and Management of Mediterranean-Farmed Fish Mortalities: A Risk Management Case Study Using Apache Spark. *Appl. Sci.* **2024**, *14*, 10112. https://doi.org/10.3390/app142210112

Academic Editors: Yujin Lim and Hideyuki Takahashi

Received: 2 October 2024
Revised: 29 October 2024
Accepted: 1 November 2024
Published: 5 November 2024

Keywords: machine learning; data mining; algorithms assessment; decision trees; random forest; linear regression; Mediterranean-farmed fish; fish mortality; aquaculture performance metrics; Apache Spark

1. Introduction

The aquaculture industry faces a major challenge of increasing mortality rates in farmed fish. Certain factors seem to affect the death ratio in fish populations. Factors like pathogens, the denormalization of ecosystems, and climate shifts contribute to the increased rates. Marine heatwaves seriously threaten marine aquaculture's sustainability, demonstrating that addressing this challenge is more than ever essential [1,2].

Moreover, pollution and pathogens from human, industrial, and agricultural activities have initiated water quality degradation, which has affected fish populations [3].

Unsustainable practices in the aquaculture industry, such as overstocking, degraded water quality, and excessive antibiotic use, can initiate diseases and cause an incremental increase in mortality rates. The rapid evolution of the aquaculture industry, supported by ecosystem degradation and disease outbreaks, is the main cause of a serious increase in fish mortality rates [4,5].

Existing studies in aquaculture have explored several approaches for addressing fish mortality risks, including digital twin technologies and real-time monitoring systems integrating IoT devices with cloud technology. While these systems aid in early detection and provide insights into the environmental conditions affecting fish health, limitations remain. Prior research often focuses on single factor analyses or small datasets, which limit predictive accuracy and scalability in complex, real-world scenarios. Additionally, traditional modeling approaches, such as Gaussian models or multivariate anomaly detection, may lack the flexibility needed for high-dimensional and non-linear data in aquaculture settings [4,5].

This study combines artificial intelligence (AI) models and agricultural sciences into a project aiming to optimize caged fish farming by introducing innovative technologies and best practices towards sustainability. This project aims to build the foundations of an advanced system for fish disease diagnosis and treatment, aiming to increase the competitiveness of Greek aquaculture. This research proposal introduces a data-driven approach that evaluates machine learning (ML) techniques—specifically Decision Trees (DTs), Random Forest, and Linear Regression—for classifying instances of fish mortality and providing actionable insights. This study overcomes previous limitations by addressing scalability and complexity by integrating these models with a scalable data storage solution (Apache Cassandra) and a distributed processing framework (Apache Spark). Key performance metrics, such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared (R^2), and the Concordance Index (C-index), are used to assess model accuracy regarding mortality prediction and factor impact. The research question centers on whether a data-driven ML approach can accurately predict Mediterranean-farmed fish species' mortality rates and identify key factors, enabling more effective management and sustainable practices in aquaculture. The model computes various factors, including locations, husbandry methods, water quality, weather conditions, and biological features. The identified thresholds operate as class value ranges, enabling the segmentation of the results into specific classes. The project aims to help through insights into sustainable aquaculture management, offering a proactive mechanism to address emerging fish mortality issues. The findings can lead to the development of targeted survival practices, enabling stakeholders to support caged-farmed fish populations.

The current study complies with the United Nations Sustainable Development Goals to reduce seafood waste and unit losses due to climate change, serving the purpose of communicating sustainable marine aquaculture best practices in Greece under the project entitled "Improving Greek Fish Farming Competitiveness", focusing on the implementation of an

intelligent system for diagnosing fish diseases and aiming to optimize the competitiveness of Greek caged fish farming by introducing AI and sustainable practices [5–7].

This article consists of eight sections, each one providing evidence of the proposed data-driven practice for caged fish stock management prediction and classification. These sections include an introduction, related work, research methodology, results, discussion, and conclusions. Each section is a step toward the completion of this task. Therefore, the introductory part clearly explains the research scope regarding caged fish losses due to various factors. In related work, we review the existing scientific literature, studies, and projects addressing similar challenges, applying ML models to predict caged fish mortality, including key research objectives. The research methodology part refers to the data analysis, software setup, and data mining techniques used to complete such a task. This section outlines the design, development, and execution of the project to address the research objectives effectively. The Results section presents the findings of this research, including initial data preprocessing and analysis, software installation, and configuration. It includes tables and figures that illustrate classifier performance and the impact of various factors on mortality rates. The final part provides a Discussion that highlights and summarizes the study's key findings, offers an overview of rule extraction, addresses limitations, and concludes with insights to help the reader grasp the core elements of this classification approach.

Research Objectives

The proposed research objectives (ROs) will try to effectively respond to the critical issue of fish mortality factors in Greek caged fish farming (Table 1). The primary objective is to manage the use of a series of software tools to conduct classifier assessment analysis. However, it is not included in scientific research as a research objective but more as a tool. This process involves a series of tool installations and configurations (Tables A1–A7). The following parts clearly demonstrate the steps and actions required for this multidisciplinary project to provide evidence not only for the aquaculture field but also to inform engineers about the potential of its implementation.

Table 1. Research objectives.

RO No	Research Objectives
1	To evaluate the classification performance of different machine learning models (DTs, Random Forest, and Linear Regression) using caged fish data.
2	To analyze the classification accuracy using performance metrics like Mean Squared Error (MSE), R-squared (R^2), Root Mean Squared Error (RMSE), and Concordance Index (C-index).
3	To identify the key factors affecting the model's predictions and provide insights into the model's interpretability.
4	To explore potential areas of improvement for model performance.
5	To assess the broader implications of these insights for fish stock management, aquaculture policy-making, and operational strategies.

2. Related Work

The current study investigates machine learning (ML) classifiers for predicting fish mortality by analyzing relevant factors in caged fish management systems. This project involves developing, configuring, and applying ML techniques and installing Apache Spark and other tools for managing fish stock. The following literature reviews maps related works, focusing on ML classifier performance, fish mortality factors, or both, to provide a context for this study.

A key concept related to this research is the "Digital Twin", which is a model designed to support a sophisticated AI Internet of Things (AIoT) system for monitoring caged fish populations in aquaculture. This system incorporates IoT devices and cloud technology

to enable the AI-driven monitoring of fish stocks. Along with ML models, the system employs sensors and hardware extensions for "smart" fish feeding, metric estimation, environmental monitoring, and fish health assessment. These sensors are real-time data collectors and transmit data to cloud services via communication networks. The system facilitates advanced decision-making processes such as data analysis, prediction, and optimization to promote more sustainable and efficient fish farming practices [8].

Advanced monitoring technologies are crucial for sustainable aquaculture, particularly in early warning systems, disease outbreak detection, and managing mass mortality risks. A study addressing these challenges proposes a modified Gaussian distribution model tailored explicitly for caged fish stock monitoring. This model offers graphical representations of the production states using a scale of alert levels, ranging from normal to dangerous. The system uses 2D image recognition techniques to monitor the health and status of fish populations, providing critical data patterns that support decision-making in aquaculture [9].

In other research, fish poisoning from unidentified causes significantly threatens public health in regions like Fiji, where fish constitutes a primary food source. A study on this issue combines fishermen's insights with ML-based association rule mining (ARM) techniques to identify hidden patterns in fish poisoning risk factors. The research highlights the contribution of environmental factors, such as contaminated migration pathways and polluted waters, to increased fish mortality rates. These findings emphasize the importance of early warning diagnostic systems for fish poisoning and the development of risk management strategies to prevent human health risks [10].

Aquaculture, a key sector in the food industry, relies on analyzing environmental factors such as salinity, temperature, bromine, ammonia, nitrogen dioxide, and hydrogen to predict and manage fish mortality. A multivariate Gaussian probability model has been applied to these factors to detect anomalies in raw data from caged fish farms. By processing daily training data, the model generates highly accurate real-time predictions of fish mortality rates, helping to mitigate risks and improve decision-making in fish farming. Such predictive models provide valuable tools for enhancing sustainability and profitability in the industry [11].

Emerging technologies, particularly AI and blockchain, are transforming the aquaculture sector. These technologies enable real-time data collection, storage, and analysis, significantly impacting supply chain transparency, market competition, and consumer trust. Blockchain ensures transparency across the farmed fish industry by providing secure access to critical data, while AI-driven solutions improve decision-making and operational efficiency. Together, these technologies address challenges in the fish farming industry by enhancing transparency and building trust among stakeholders, from producers to consumers [12].

The performance of ML classifiers is influenced by various factors, including the characteristics of the data and their specific application. Research comparing k-Nearest Neighbors (k-NNs) and Naïve Bayes (NB) classifiers reveals that these models perform differently depending on the dataset. For instance, k-NN shows high accuracy with small datasets but incurs a high computational cost. At the same time, NB is effective with large datasets but may suffer from reduced accuracy when feature independence assumptions are violated. This research highlights the need for the careful selection of classifiers based on the characteristics of the data to ensure high performance and accuracy in predictive analytics [13].

Further studies compare DTs and NB classifiers enhanced by Genetic Algorithms (GA) for feature selection. One such study, using UCI Machine Learning Repository data, demonstrates how incorporating GA improves classification accuracy for both models, with DTs generally outperforming NB in accuracy. This highlights the role of GA in optimizing model performance through more efficient feature selection processes [14].

Additional research explores the comparative performance of classifiers such as Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), k-NN, and NB, identifying

strengths and limitations. ANN offers high classification accuracy and powerful learning capabilities but requires long training times and is prone to overfitting. SVM excels in handling non-linear relationships and binary classification tasks with small and medium datasets but struggles with multi-class problems. k-NN is effective for small datasets but incurs high computational costs for predictions and becomes inefficient with large datasets. NB is well-suited for large datasets, demonstrating efficiency in classifying examples, but its assumption of feature independence can lead to performance limitations in complex real-world scenarios. On the other hand, DTs perform well with high interpretability and ease of decision rule generation, but they are vulnerable to overfitting and struggle with complex datasets [15–17].

Evaluating ML classifier performance often relies on traditional metrics like Recall, Precision, and F-measure. However, recent studies have shown that these metrics may introduce bias, particularly in multi-class classification tasks, leading to misleading assessments of model performance. One study suggests the adoption of alternative performance metrics to enhance the reliability and trustworthiness of ML model evaluations. By incorporating these new metrics, the research aims to improve model interpretability and the effectiveness of decision-making processes, ensuring more accurate evaluations of predictive systems [18].

3. Research Methodology

3.1. Research Scope

This study applies ML models to predict farmed fish mortality rates in aquaculture environments, providing information about its true contribution to science, industry, and humanity [19,20].

Regarding collected and processed data, this study analyzes two datasets, one gathered from March 2016 to June 2022 and the second from January 2018 to November 2022. These datasets have environmental, operational, and biological data from growing marine cage fish farming facilities in the Ionian Sea, Greece. The scope of the data introduces variables including water temperature, salinity, oxygen, fish populations, feeding rates, number of deaths, etc. Among a series of ML models, only three ML models known for their predictive accuracy performance were assessed to finally process and predict insights into forecasting fish mortality rates [21].

This study examines how environmental conditions and operational decisions impact fish death rates to determine the most influential environmental and operational factors. It also assesses the significance of these factors through feature importance analysis.

The data were analyzed using various software tools and configurations, including Apache Spark 3.40, Apache Cassandra 4.1.0, Spark Connector, Java Development Kit 17 (JDK), Ubuntu Linux 22.04, and Oracle VM VirtualBox 7.0. These tools were selected for their capability to operate in a distributed computational environment, enabling the efficient management of large datasets [22–27].

Data explanation and model interpretability are used to interpret the results in a beneficial way, allowing the aquaculture industry to understand the methodology. This leads to optimized fish stock management practices that consider environmental factors and operational strategies.

Regarding the contributions to science, industry, and society, there are potential future practical implications in the field of aquaculture ecosystems. The insights from this study can enhance decision-making in areas like policy-making, sustainable practices, and operational optimizations to preserve ecosystem resources.

3.2. Method Overview

The current study leverages machine learning (ML) models to analyze data from aquaculture facilities in the Ionian Sea, Greece, aiming to predict and manage farmed fish mortalities. The research methodology follows a structured, eight-stage process, as illustrated in Figure 1.

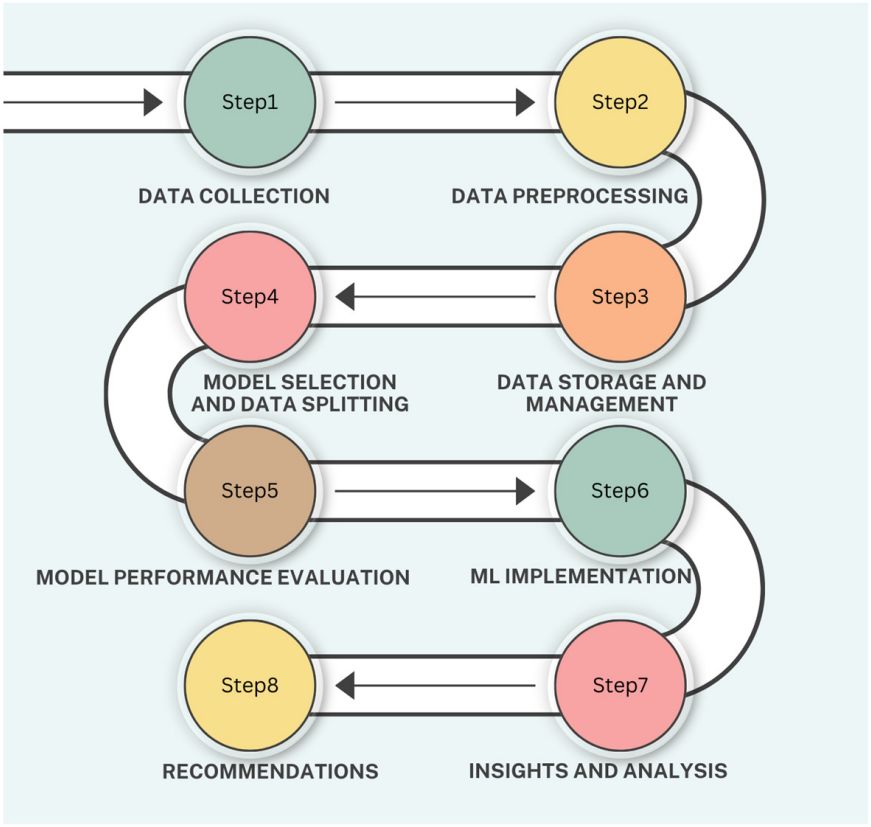

Figure 1. Process overview.

Each stage of the methodology builds upon the previous one, ensuring a systematic flow of data and processes. Below is a detailed breakdown of each step, specifying the inputs and outputs at each stage:

Stage 1: Data Collection

The first step involves collecting two datasets from aquaculture facilities. The first dataset includes water chemical and microbiological data collected from March 2016 to June 2022. The second dataset contains operational data, including fish counts, mortality rates, feeding practices, and environmental conditions, collected from January 2018 to November 2022 [21]. These datasets serve as the **input** for subsequent preprocessing.

Stage 2: Data Preprocessing

Both datasets undergo preprocessing to ensure data quality and consistency in this step. This includes handling missing values, normalizing variables, and removing outliers [21]. Preprocessed data are the input for data storage and management in the next stage.

Stage 3: Data Storage and Management

After preprocessing, the data are stored and managed using an Apache Cassandra 4.1.0 setup for distributed data storage. Data processing and ML tasks are executed with Apache Spark 3.4.0 and integrated via the Spark–Cassandra Connector. This step uses a virtual machine setup (Oracle VM VirtualBox 7.0 running Ubuntu Linux 22.04 on a Windows 10 host OS) [22–27]. The input to this stage is the clean, preprocessed data, which is stored and managed for subsequent model selection and analysis.

Stage 4: Model Selection and Data Splitting

In this stage, three ML models are chosen: DTs, Random Forest, and Linear Regression. These models are selected for their performance in predictive tasks using the collected data. At this stage, the preprocessed dataset is split into training and testing subsets [28–33]. The input is the managed dataset, which is split for training and testing the ML models.

Stage 5: Model Performance Evaluation

Model performance is assessed using the following metrics: MSE, RMSE, R^2, and C-index. These metrics help measure how well each model predicts fish mortality rates [28–33]. The trained models and the testing dataset are input to this stage. The output is the performance metrics that guide model refinement and future ML implementations.

Stage 6: ML Implementation

This stage focuses on applying the selected ML models to the dataset to identify the key factors affecting fish mortality. The input to this stage includes the trained ML models and the testing data [28–33]. The ML implementation helps predict fish mortality and provides insights into factors influencing survival rates.

Stage 7: Insights and Analysis

In this stage, the results from ML model predictions are analyzed. This analysis helps identify trends, anomalies, and relationships between variables. The input is the predictions generated by the ML models. The outcome of this stage offers new insights into aquaculture management, helping to fine-tune practices and strategies [28–33].

Stage 8: Recommendations

The final stage compiles the study's findings and provides recommendations for improving aquaculture management. The recommendations focus on optimizing water quality, feeding practices, and environmental management to enhance fish survival and sustainability [28–33]. The input includes the insights from Stage 7, which are translated into practical, actionable strategies for aquaculture facilities.

3.3. Software

Several tools were employed throughout this study to ensure efficient data processing. The research was conducted in a Linux environment using Ubuntu Linux and Oracle VM VirtualBox. Apache Cassandra, a database system capable of handling large volumes of data, was used for data storage and management [22–26].

The ML and data processing tasks were carried out using Apache Spark. This platform provided the necessary tools for implementing various models, such as Linear Regression, Random Forest, and DTs.

In addition, various tools were utilized for managing structured data. The various tools used to analyze and predict the study's results provided the necessary accuracy and comprehensiveness. Therefore, a series of software installation steps took place in a Microsoft Windows 10 Environment.

- Installing and using Oracle VM VirtualBox as a virtualization tool.
- Installing Ubuntu Linux in an Oracle VM VirtualBox.
- Installing Apache Cassandra on Ubuntu Linux.
- Installing Apache Spark on Ubuntu Linux.
- Installing Apache Cassandra and Spark Connector on Ubuntu Linux [20,22–25].

3.4. Dataset

3.4.1. Dataset Introduction

The dataset provides insights into the data collected, including the environmental and operational features crucial for predicting caged fish populations and mortality rates. This part includes data acquisition and preprocessing sections [21].

3.4.2. Data Acquisition

The dataset collected from aquaculture farm cages in Greece's Ionian Sea was used to analyze fish mortality, offering detailed information on various practices and serving as a valuable source of insight.

The data consisted of 37,203 unique instances of one nominal and four numerical factors, including variables like the fish's median atomic (individual) weight (MAB), the

volume of the cage occupied by the fish (Vol), the stocking density of fish within the cage (i_f), the water temperature (Temp), and the number of "Deaths" (Table 2) [21].

Table 2. Caged fish farming parameters.

Variable	Type	Description
Location	Text	Geographical Location Where the Sample Was Collected
Sample Code	Text	Unique Identifier for Each Sample Collected
Day	Text	Specific Day on Which the Data Were Recorded
Description	Text	Additional Details or Context About the Sample
State	Text	Condition Or State of The Sample at The Time of Collection
Analysis	Text	Type Of Analysis Performed on The Sample
Parameter	Text	Specific Measured Parameters (e.g., Water Temperature, Ph)
Value	Double	Numerical Value of The Measured Parameter
Unit	Text	Unit Of Measurement for The Parameter Value
Limit	Double	Threshold Critical Level Values for Certain Parameters
Aa_X	Text	Additional Coordinate or Identifier Associated with The Sample
Month	Text	Month Of Data Collection, Providing Temporal Context
Cell	Text	Section or Cell (Cage) within a Location Where the Sample Was Taken
Portion	Text	Portion of the Sample or Area Being Analyzed
Fishno	Integer	Identifier for an Individual Fish Within a Sample
Mab	Text	Code or Identifier Possibly Related to a Management Area
Fish_No	Integer	Count of Fish Within a Sample or Study Area
Deaths	Integer	Number of Fish Deaths Recorded Within the Sample
Corrections	Integer	Adjustments Made to the Data (e.g., Due to Errors)
Nofishing	Text	Information About Areas or Periods Where No Fishing (No Harvesting) Occurred
Damage	Text	Notes on Any Damage Observed in the Sample
Kg_Fishing	Double	Weight of Harvested Fish (in Kilograms)
Sample	Text	Further Identification or Notes About the Sample
Medication	Text	Details on Any Medication Administered to Fish
Food	Text	Information on the Type of Food Provided to the Fish
Food_Model	Text	Description of the Feeding Model or Regime Used
Sfrpercent	Double	Percentage (%) of a Specific Food Ratio or Supplement in the Fish Diet
Temp	Double	Water Temperature at the Time of Sampling
Vol	Double	Volume of Water or Sample Analyzed
i_f	Text	Specific Index or Factor Relevant to the Sample
Medicine	Text	Details on Any Additional Medicine Used
Aa_Y	Text	Coordinate or Identifier Potentially Linked to Location Data

The collected data are organized into ten columns and have various attributes, such as the sample description, date, analysis parameter, area, sample code, and value [21].

The second dataset analyzed the daily operations of aquaculture facilities. These data, which are part of a larger study, consist of 237,969 instances and 21 columns. It has information on various aspects of the industry, such as the month, day, day, cage, batch, number of fish, average individual weight, number of transfers, deaths, corrections, kilograms, sample, medication, food, type of food, SFR, water temperature, and i_f [21].

The data collected during this study provided valuable information on water quality and benthos' condition, which can help improve the industry's sustainability. This study aims to provide decision-makers with an understanding of the different aspects of the aquaculture industry's environmental factors, aiming to improve the quality of production and management practices (Table 2).

3.4.3. Dataset Preprocessing

This study section outlines the various steps involved in data preparation, including data preprocessing before the final analysis. The process begins with data cleaning to ensure the correctness of values, followed by data transformation and the normalization of attributes to make them compatible with machine learning algorithms. The data

preprocessing phase is crucial for improving the ML models' prediction accuracy and overall performance.

A series of steps were included in the data preprocessing phase. Data preprocessing steps refer to the following:

1. Data export and loading from the CSV files into Pandas DataFrames 1.5.0, which are used for data analysis and processing in the Python 3.10 programming language.
2. Data completeness checks identify inconsistent or missing data such as null or NaN values. A data frame is analyzed to reveal whether there are any anomalies or values that need replacement.
3. Data cleansing is used to identify and eliminate data inconsistencies. This may include low or high values, exclusions, or correcting certain typos.
4. Data transformation is conducted in the data preprocessing phase, such as discretization or normalization.

After data processing, the two datasets are merged using the Pandas 1.5.0 'merge' function. The resulting combined dataset includes a column with microbiological and physicochemical data. It is stored in the Apache Cassandra database, which, with its NoSQL structure, provides an ideal environment for analyzing large datasets. The ML models' algorithms are implemented using a powerful tool for big data analysis: the Apache Spark framework [20]. Apache Spark SQL is used for data extraction and preprocessing from Cassandra and Mllib. MLlib library of Apache Spark 3.4.0 is also used for the overall data analysis [23–27].

3.4.4. Dataset Insertion into Apache Cassandra

This part of the study refers to loading preprocessed data into Apache Cassandra. It provides information on how the data were managed and stored in the NoSQL database. It also explains how Apache Spark and Cassandra can be used to perform ML model performance analysis. The current setup is segmented into steps to make the entire process easier to follow, analyze, and handle the massive load of data related to aquaculture. CQL is a type of query language that Apache Cassandra uses to handle data. CQL was used to import CSV data into the platform in this section. The command "cqlsh" is used to access the Cassandra Query Language shell, enabling interactions with the Apache Cassandra database system [23–27].

1. Step 1: Create a keyspace (Table A1).
2. Step 2: Select keyspace (Table A2).
3. Step 3: Create the table (Table A3).
4. Step 4: Import the data from the CSV file (Table A4).

3.4.5. Dataset Import from Apache Cassandra into Apache Spark

The process of importing preprocessed data from Apache Cassandra into Apache Spark is described in the following steps:

1. Installation and configuration of the Cassandra Spark Connector. The Spark Connector is installed and configured to allow the creation of DataFrames and Spark RDDs in the Spark environment using data from Cassandra.
2. The entry point for every operation is the creation of an Apache Spark session. The connection parameters for the Cassandra database are also configured (Table A5).
3. Since the creation of the Apache Spark session signifies the entry point for every operation, data loading from Cassandra occurs. Following this, the DataFrame is created using the "*read.format*" method. The "*table_name*" and the "*keyspace_name*" are replaced with the names of the tables and keyspaces in the Cassandra database (Table A6).
4. Data load verification is performed by executing queries on the DataFrame to ensure accuracy and consistency. The "show()" method is then used to pinpoint the first few elements of the frame (Table A7).

3.5. Classification Algorithms

The dataset used in this study was collected from the aquaculture industry and required extensive preprocessing to ensure suitability for ML model performance. Relevant attributes were selected based on their alignment with the study's research objectives, including environmental factors, such as salinity and water temperature, and operational variables, like gear types and fishing effort. The data underwent cleansing, adjustment, and normalization steps to handle missing values and normalize features before being processed by the selected ML models.

For this study, we focused on three specific ML algorithms: Random Forest, DTs, and Linear Regression. These algorithms were chosen based on their compatibility with the characteristics of aquaculture data and their respective strengths in managing environmental and operational datasets. Random Forest was selected for its robustness in handling high-dimensional data and its ability to reduce overfitting, which is critical in datasets with noise and numerous environmental factors. DTs were chosen for their interpretability and versatility with both categorical and numerical data, making them ideal for classification tasks where explicit, interpretable models are needed. Linear Regression was included to address cases where the data suggested linear relationships, allowing us to analyze specific attributes' effects on outcomes straightforwardly and interpretably. These algorithms offer a complementary balance of robustness, interpretability, and efficiency, particularly well-suited for modeling in aquaculture, where both structured and environmental data types are common.

Data were stored in the Apache Cassandra database, ensuring efficient management within the system, and were then loaded into the Apache Spark framework, where the Random Forest, DTs, and Linear Regression models were trained, validated, and evaluated. The dataset was split into a training set (70%) and a testing set (30%) to assess model effectiveness through the training and testing processes [28–33]. Table 3 below summarizes each algorithm's strengths, weaknesses, and best use cases, clarifying the unique value each adds to this study's predictive analysis.

Table 3. Algorithm performance comparison.

Algorithm	Strengths	Weaknesses	Best Use Cases
Random Forest	Reduces overfitting. Manages high-dimensional data well. Provides feature importance.	Computationally strong. Less interpretable.	Complex datasets with noise.
Decision Trees	Simple to interpret. Handles categorical and numerical data.	Prone to overfitting. Unstable.	Classification tasks, especially where model interpretability is key.
Linear Regression	Easy to interpret. Efficient for small datasets.	Assumes linearity. Affected by outliers.	Regression problems with a clear linear relationship.

3.5.1. Decision Trees

DTs are tools that can help analyze and interpret a user's decisions and provide recommendations on their possible consequences. Linear Regression is a widely used and basic method for calculating the relationship between a variable and one or more independent ones [28–33].

DTs are a widely used technique in machine learning for performing regression and classification tasks. They recursively divide a dataset into constituent parts according to the most salient attributes [28–33].

DTs' structure is composed of various internal nodes and branches that represent the various attributes of the data. These nodes are used to determine the best classification or regression model for the given dataset. The most common algorithms used in DT inductions are C4.5, ID3, and CART. These algorithms use different approaches to choose the best classification or regression model [28–33].

The splitting criterion for DTs is usually based on the information gained, which considers the amount of information that a feature provides about a class.

The value of the information that is gained is expressed gradually. The Shannon function's mathematical expression is shown as follows:

$$I(P(u_1), \ldots, P(u_n)) = \sum_{i=1}^{n} - P(u_i)\, log_2 P(u_i) \quad (1)$$

where the information gain that is obtained by splitting a dataset is referred to as $I(P(u_1), \ldots, P(u_n))$. The value of $P(u)$ is dependent on the attributes of the data that are split. While the $P(u_i)$ refers to the probability of the answer that is given by the split (u_i) [28–33].

$$Remainder\ (A) = \sum_{i=1}^{n} \frac{p_i + n_i}{p + n}\ I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right) \quad (2)$$

$$Gain\ (A) = I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right) - Remainder\ (A) \quad (3)$$

$$Gini\ (A) = 1 - \sum_{i=1}^{n} p_i^2 \quad (4)$$

One of the main disadvantages of DTs is that they are prone to overfitting, which can occur when a tree grows too big and collects noise in the collected data. To minimize this issue, pruning techniques are usually used. These can be performed either during the building phase or post-pruning. Although DTs are generally interpretable and simple, they can also be affected by slight variations in data [28–33].

3.5.2. Random Forest

According to the scientific literature, one of the most powerful methods for developing ML models is Random Forest, which involves constructing several DTs and estimating the mean prediction and classification of individual trees. Random Forest is a machine learning technique that enhances performance by combining multiple DTs during training. It addresses one of the most common issues in DTs, overfitting, which occurs when multiple models are used to analyze the data. Training each tree on a randomly selected subset of the data allows for more robust models [28–33].

A Random Forest is a combination of DTs. Although it does not have a single formula, it relies on the predictions of various DTs to arrive at its results.

For classification, the following is calculated:

$$\hat{y} = majority\ vote(T_1(x), T_2(x), \ldots, T_n(x)) \quad (5)$$

where $T_i(x)$ is the prediction from the ith tree, and $\hat{y}$ is the final prediction (the class with the most votes).

For regression, the following is calculated:

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} T_i(x) \quad (6)$$

where $T_i(x)$ is the predicted value from the ith tree, and $\hat{y}$ is the average of all predicted values.

Random Forest trees are constructed on a randomly selected subset of the training data. The features are also randomly chosen at each split.

One of the main advantages of the Random Forest is its ability to handle large-scale datasets. It can analyze various features of the data, and its feature importance scores are useful in determining the impact of each one on a prediction's outcome. However, it is not as interpretable as a single DT. In addition, the number of trees in the cluster makes it harder to perform well [28–33].

3.5.3. Linear Regression

One of the most common algorithms for performing regression tasks is Linear Regression. This type of algorithm assumes that the input and output variables have a linear relationship. It can be performed by fitting a line or hypervariable plane to minimize errors between the predicted and actual values [28–33].

Although Linear Regression is straightforward to implement and use, its performance can be limited by the relationship between the variables' underlying values. It can also be sensitive to outliers. Despite these limitations and disadvantages, it is still widely used due to its effectiveness and simplicity in performing tasks where a linear relationship exists [28–33].

Linear Regression is a type of statistical model that shows the relationship between a given variable and another variable. It uses a linear equation to arrive at its formula.

$$y = \beta_0 + \beta_1 x + \varepsilon \tag{7}$$

The same formula for multiple independent variables, or multiple Linear Regressions, is performed as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon \tag{8}$$

where β_0 is the constant, β_1, β_2, β_n are the coefficients for each independent variable, x_1, x_2, x_n are the independent variables, y is the dependent variable, and ε is the error term, accounting for the difference between the actual and predicted values [28–33].

Linear Regression aims to minimize the sum of squared errors, which are calculated using the following formula:

$$SSE = \sum_{i=1}^{n} (y_1 - \hat{y}_1)^2 \tag{9}$$

where y_1 is the actual value, and $\hat{y}_1$ is the predicted value [28–33].

4. Results

This study analyzes the DTs, Random Forest, and Linear Regression models in detail. The results are presented in the following sections.

4.1. Decision Trees (DTs)

The following describes the steps to apply the DT model and create a report and graphical representations. The training results include the features and their importance alongside plots that show the model predictions against the actual values.

1. The training set was utilized to train the DTs model.
2. The evaluation of the model was carried out using the testing framework.
3. The importance of attributes will be discussed according to the model's findings (Table 4).
4. After the model has been implemented, reports and graphical representations were created to show the model's predictions against the test set's values (Figures 2 and 3).
5. A Python code was also used to conduct DTs analysis using Cassandra data through Apache Spark (Table A9) [34–38].

Table 4. DT results analysis.

Variable	Coefficient Value
RMSE	64.21
R^2	−1.16
MSE	4123.37
C-index	0.65

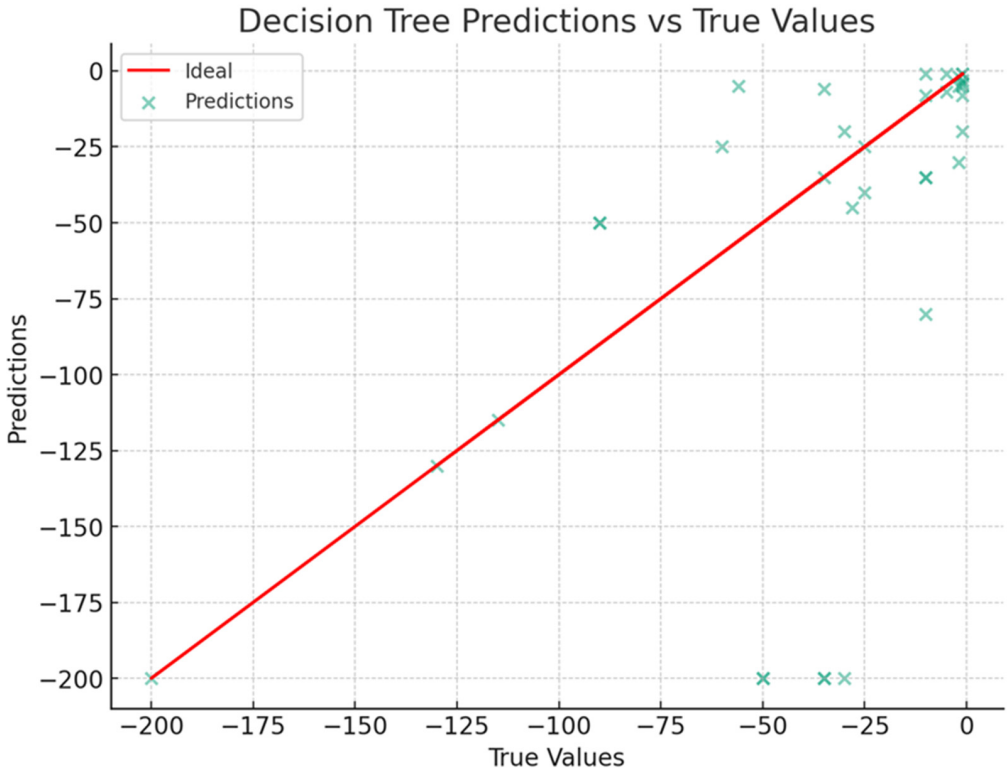

Figure 2. DTs predictions versus true values.

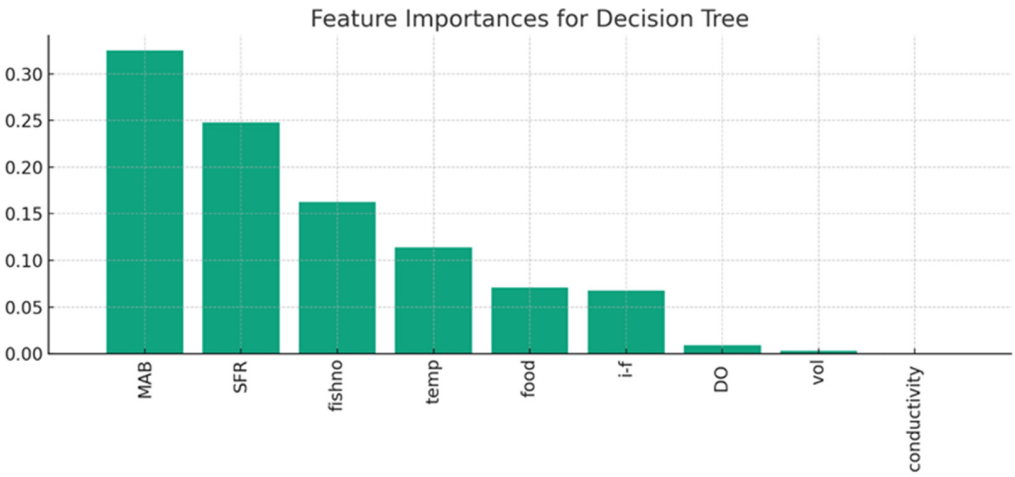

Figure 3. DTs predicted factors affecting fish mortality.

Figure 2 illustrates the performance of the DTs model by comparing predicted fish mortality values against the actual mortality values. The red line represents the ideal scenario where predictions perfectly match the true values (i.e., the line of perfect fit). Each data point (marked by green 'x') corresponds to an actual value plotted against its respective prediction. The scatter of points around the red line indicates the level of the

prediction error. While some points align closely with the ideal line, reflecting accurate predictions, a significant number deviates from it, particularly in the higher and lower ranges, which suggests that the model struggles with extreme mortality values. This visualization highlights both the strengths and limitations of the DT model in this context, where it may underperform in predicting the more extreme outcomes.

- The RMSE value is 64.21, indicating a significant variance between the predicted values and the actual data. A lower RMSE would indicate greater model accuracy, suggesting there is room for improvement in prediction precision (Table 4).
- The R^2 value is −1.16, which is significantly lower than the ideal value of 1.0. A negative R^2 indicates that the model performs worse than simply predicting the mean of the data, suggesting that the current model does not explain the variance in the dataset effectively (Table 4).
- The MSE value is 4123.37, representing the average squared difference between the predicted and actual values. A high MSE suggests that the model makes substantial prediction errors, further highlighting the need for optimization or an alternative modeling approach (Table 4) [34–38].
- The C-index of 0.65 reflects its moderate ranking accuracy (Table 4).

4.2. Random Forest

1. The evaluation metrics for the Random Forest model were measured and presented based on its predictions for the validation set (Table 5).
2. A prediction graph was created to compare the model's results with the actual values of the test set (Figure 4).
3. The significance of the Random Forest model's features is depicted in Figure 5.
4. A Python code was also used to conduct a Random Forest analysis using the Apache Cassandra data through Apache Spark (Table A8) [34–38].

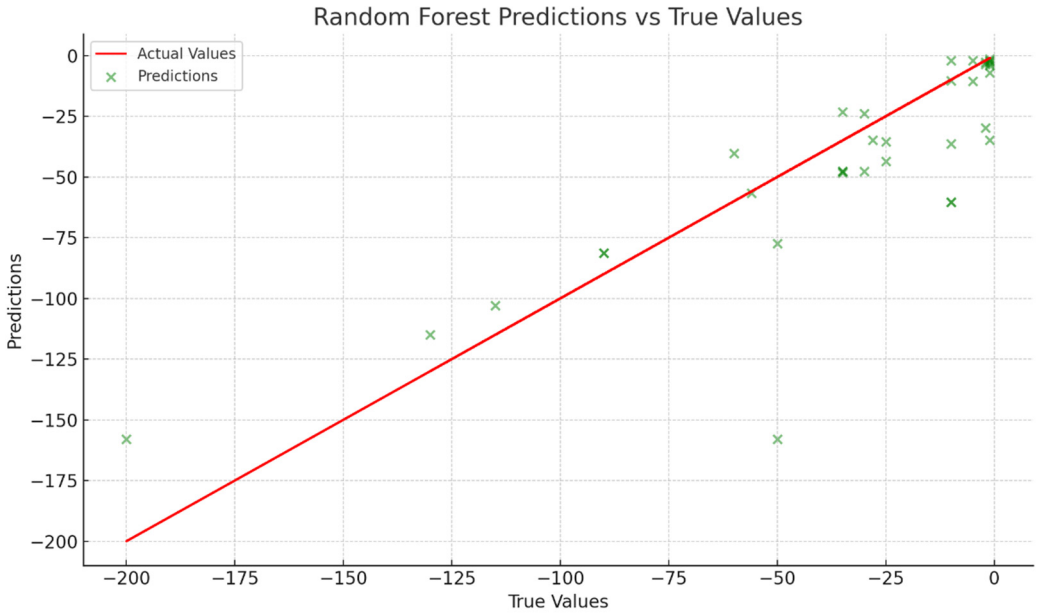

Figure 4. Random Forest predictions versus true values.

Figure 4 shows the predictions of the Random Forest model compared to the actual fish mortality values. The red line represents the ideal fit, where predictions match the actual values perfectly. Each green 'x' marks a predicted value plotted against its corresponding true value. The proximity of these points to the red line suggests that the Random Forest model performs effectively, with most predictions closely aligning with the actual values.

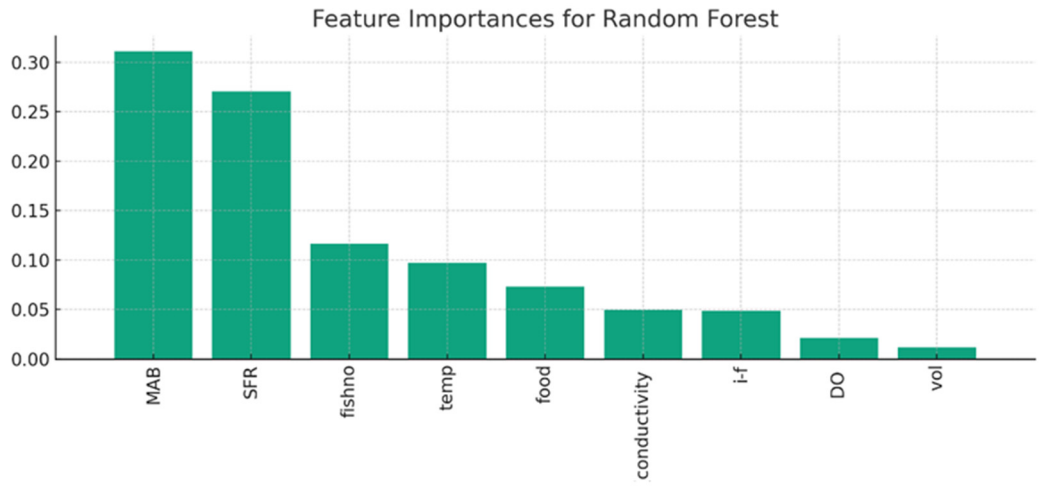

Figure 5. Random Forest-predicted factors affecting fish mortality.

Table 5. Random Forest results analysis.

Variable	Coefficient Value
RMSE	64.21
R^2	0.63
MSE	698.54
C-index	0.85

When comparing this plot to the DT model, the Random Forest model's predictions are consistently closer to the red line, indicating better prediction accuracy. While there are still some deviations, particularly for the extreme mortality values, the overall performance of Random Forest is superior, showing fewer outliers and more accurate predictions across a wider range of values.

Figure 5 illustrates the importance of different features in the Random Forest model's prediction of fish mortality. The size of each feature's bar indicates its relative contribution to the model's accuracy, with larger bars signifying a greater influence on mortality predictions. For example, MAB and SFR are the most important, suggesting that they play a significant role in forecasting, while other factors like DO and vol have comparatively smaller impacts.

The Random Forest model has a lower MSE than the *DTs*, indicating more accurate predictions. Its R^2 score shows that it can explain approximately 63% of the variance in target values, demonstrating a good predictive capacity. This makes the Random Forest model more reliable for forecasting target values in fish culture data.

- The RMSE value is 64.21, indicating the average deviation of the predicted values from the actual values. While this value is still significant, it shows a moderate level of prediction error. A lower RMSE would indicate better accuracy (Table 5).
- The R^2 value is 0.63, suggesting that the model explains 63% of the variance in the data. While it is not perfect (with 1.0 being ideal), an R^2 of 0.63 indicates a decent fit, meaning that the model captures a good portion of the variability in the dataset (Table 5).
- The MSE value is 698.54, representing the average squared difference between the predicted and actual values. This lower MSE, compared to the previous scenario, indicates an improvement in the prediction accuracy, as a lower MSE reflects fewer errors in the predictions (Table 5).
- The C-index is 0.85, indicating a high level of ranking accuracy (Table 5).

4.3. Linear Regression

Linear Regression is one of the most common methods for analyzing various variable relationships. This method aims to predict the value that a dependent variable will exhibit after considering the independent variables' values. The results of this procedure are then analyzed using the coefficients.

Linear Regression can be performed in Spark using the MLlib library. However, before it can be performed, it is necessary to convert the data into numerical format using OneHotEncoder and StringIndexer. Furthermore, the VectorAssembler tool created a single feature vector for all the features. The Python code was also used to conduct a Linear Regression analysis using Cassandra data through Apache Spark (Table A10).

The results are illustrated in the following three graphs. The first graph (Figure 6) shows the parameters influencing fish mortality, displaying the coefficients of various features in the Linear Regression model used for prediction, focusing on "conductivity" and "DO". Each bar represents a feature's coefficient value, reflecting its impact on the model's predictions. Notably, "SFR" and "DO" exhibit significant coefficient values, indicating a stronger influence on the model's output than other features.

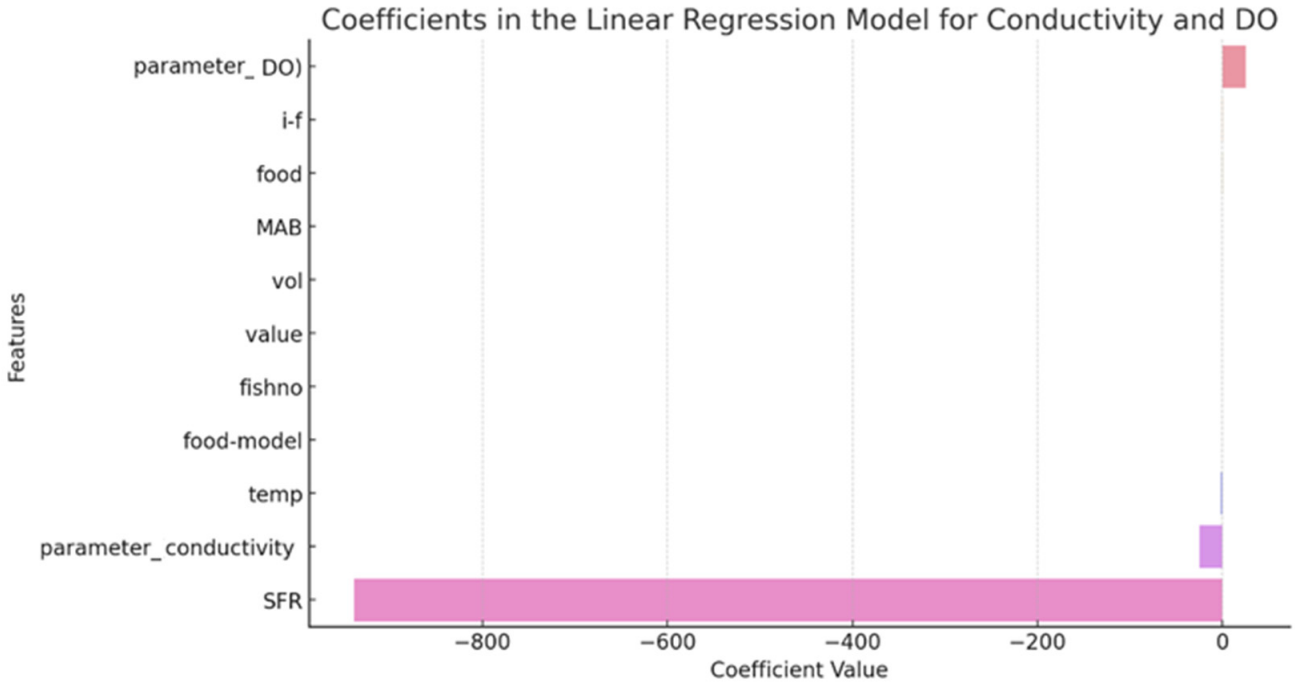

Figure 6. *Linear Regression*-predicted factors affecting fish mortality.

In the second graph, the x-axis scale is adjusted to enhance the readability of smaller coefficients, allowing their relative influence to stand out more clearly (Figure 7). This focused view highlights the minor but noteworthy contributions of features such as "i_f", "food", and "MAB" in the Linear Regression model, providing a more detailed perspective on how these variables impact fish mortality predictions.

The third graph shows the outcomes of the Linear Regression procedure (Figure 8).

Figure 8 illustrates the relationship between predicted and actual values. A perfect forecast is when the predicted values are like the actual ones.

Although these coefficients show that each of these elements impacts fish mortality, the magnitude of their effects varies. A brief analysis of the coefficients for each variable of the model's Linear Regression framework is provided. These coefficients indicate that each of these variables has some effect on fish mortality (Table 6) [34–38].

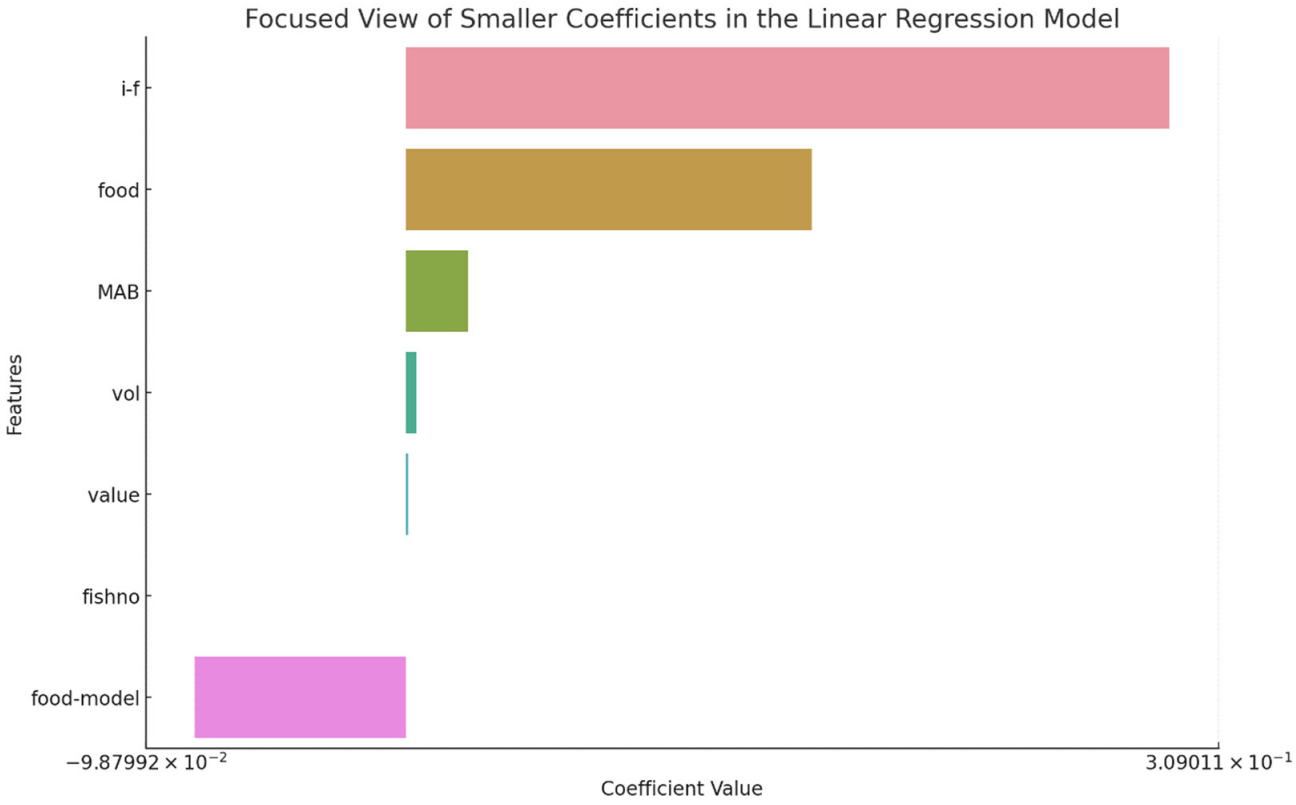

Figure 7. Factors of coefficient scale fish mortality.

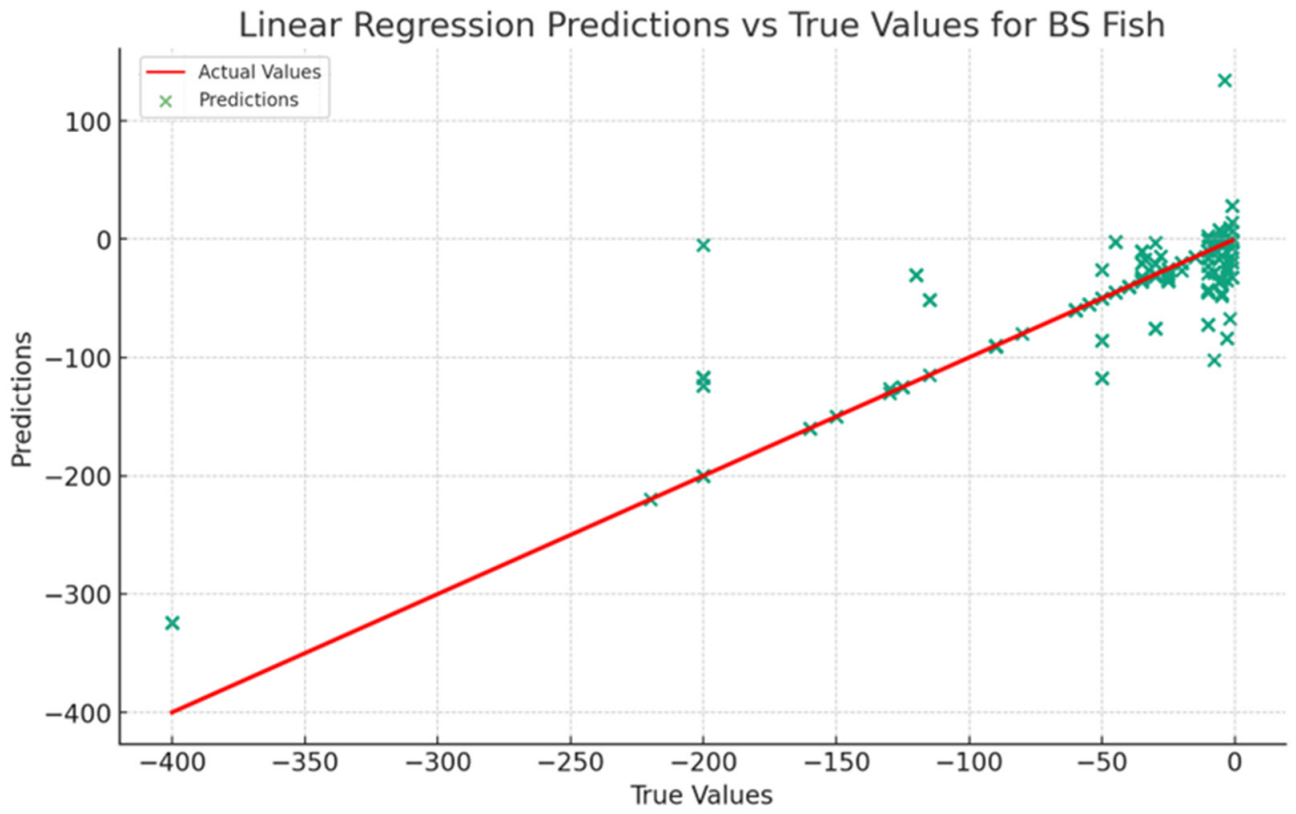

Figure 8. Linear regression predictions versus true values for BS fish.

Table 6. Variables coefficients.

Variable	Coefficient Value
Dissolved Oxygen (DO)	24,853
i_f	0.290
Food	0.154
MAB	0.024
Vol	0.004
Value	0.001
Fishno	0.0001
Temp	−2.139
Conductivity	−24,853
SFR	−939.007

- *Dissolved Oxygen (DO)* (24,853): the positive coefficient of *DO* (24,853) suggests that there is a link between the increasing levels of this element and the mortality rate.
- *i_f* (0.290): The increase in fish concentration is also linked to an increased mortality rate. However, the effect is not as strong as with *DO* (Table 6).
- *Food* (0.154): the food factor indicates that increased food quantity leads to higher mortality rates (Table 6).
- *MAB* (0.024): the small but positive relationship between the *MAB* and mortality rates suggests that higher atomic weights may contribute to increased deaths.
- *Vol* (0.004): the effect of the fish volume on mortality in each space is small.
- *Value* (0.001): the small positive effect of the value indicates that the relationship between the parameter and mortality rates is not strong (Table 6).
- *Fishno* (0.0001): the minimal effect of the *Fishno* variable on mortality is shown.
- *Food Model* (−0.080): the Food Model has a negative effect, indicating that certain kinds of food can help reduce fatalities.
- *Temp* (−2.139): the negative effect of temperature is shown, suggesting that increased temperatures can reduce the mortality rate.
- *Conductivity* (−24,853): The relationship between conductivity and mortality is negative. This indicates that elevated conductivity can reduce the mortality rate.
- *SFR* (−939.007): the relationship between the *SFR* and mortality is negative, indicating that higher values can substantially reduce mortality (Table 6).

The effect of food types on mortality is negative, as they are considered a single variable. This suggests that changing the type of food can have a negative effect on the number of deaths. Although individual food types have varying effects, the overall impact of this single variable on mortality is negative. This finding provides a more complete understanding of how feed type can affect the mortality rate in aquaculture. A Linear Regression analysis was performed on sea bass (BS). The results indicated that the fish behaved well in the study (Table 7) [34–38].

Table 7. Linear Regression results analysis.

Variable	Coefficient Value
RMSE	51.73
R^2	0.31
MSE	2675.84
C-index	0.70

Linear Regression analysis was performed on sea bass (BS), which is a type of fish. The results of the Linear Regression for 'BS' fish are as follows:

- The RMSE represents the variance between the predicted value and the actual dataset at approximately 51.73 percent. It indicates that the model's performance is better if the value is closer to zero (Table 7).

- R^2 is a statistical measure that predicts the likelihood that the model will be able to predict future samples. The value of R^2 is 0.31 when a grade of 1.0 is the ideal outcome (Table 7).
- The value of MSE is 2675.84, indicating the average squared difference between the predicted and actual values. A high MSE suggests that the model makes significant errors in its predictions, emphasizing the necessity for optimization or considering an alternative modeling approach (Table 7).
- The C-index of 0.70 highlights its relative performance in predicting mortality rates (Table 7).

5. Discussion

The results of this study demonstrate how different ML techniques, such as Random Forest, DTs, and Linear Regression, can enhance the prediction of fish mortality rates. Among these, the Random Forest model outperformed both DTs and Linear Regression, achieving the lowest MSE, and the highest R^2 value, as shown in Table 8.

Table 8. Algorithms performance results.

ML Model	MSE	R^2	RMSE	C-Index
Random Forest	698.54	0.63	26.43	0.85
Decision Tree	4123.37	−1.16	64.21	0.65
Linear Regression	2675.84	0.312	51.73	0.70

The C-index values indicate the models' ranking accuracy, providing a measure of the model's ability to rank-order outcomes accurately, where Random Forest achieved the highest C-index of 0.85, demonstrating superior predictive performance. DTs and Linear Regression recorded C-index values of 0.65 and 0.70, respectively, highlighting Random Forest's advantage in maintaining prediction consistency across data instances.

Although Linear Regression performed adequately when predicting mid-range values, it exhibited significant deviations in extreme data points, while the DTs model performed poorly overall.

In addition to prediction accuracy, computational efficiency was a key factor in the assessment. As shown in Table 9, the Random Forest model required the most computational resources, with a CPU time ranging from 100 to 200 s and memory usage between 1200 and 1800 MB. Conversely, DTs and Linear Regression were much more efficient, with Linear Regression being the fastest and most lightweight in terms of CPU and memory usage.

Table 9. CPU and memory comparison.

ML Model	CPU Time (Seconds)	Usage (MB)
Random Forest	100–200	1200–1800
Decision Tree	30–60	600–800
Linear Regression	15–30	200–300

These computational insights highlight a trade-off between model accuracy and resource efficiency. While Random Forest provided the most accurate predictions, its resource demands were substantially higher. DTs and Linear Regression, although more computationally efficient, struggled with prediction accuracy, particularly with more complex relationships in the data.

Specific environmental factors, such as temperature and water conductivity, significantly impacted fish mortality rates. Higher temperatures were correlated with lower death rates, while optimal water conditions and feeding techniques helped mitigate mortality. Notably, the type of food used also showed a negative correlation with mortality rates, indicating that tailored feeding programs could reduce mortality in aquaculture settings.

The findings of this study provide valuable insights into enhancing aquaculture management practices, particularly in refining feeding strategies and improving environmental standards [39].

A performance assessment evaluation of several ML models for predicting fish mortality rates in the aquaculture industry revealed that Random Forest, despite its higher computational cost, outperformed the other models. It achieved the lowest MSE and the highest coefficient of determination (R^2) at 0.63. These performance metrics indicate that the Random Forest model can explain the variance in the data and produce reliable predictions. The analysis also revealed that DTs performed poorly, with an MSE of 4123.37 and an R^2 of −1.16, indicating their inability to provide reliable predictions.

Linear Regression *was more interpretable, but it faced difficulties with* extreme values. This issue pinpoints the limitations when dealing with non-linearity.

Random Forest insights showed that certain factors, including water quality parameters and feeding rates, affected fish mortality rates. This finding strengthens the existing research on the operational and environmental variables or factors that impact fish mortality rates. Certain features in the model's performance were analyzed to determine the potential of developing targeted interruptions to minimize fish mortality rates in cages.

The model's graphical representations of predictions showed that specific interventions, such as applying targeted practices, could mitigate fish mortality rates in the aquaculture industry [39,40].

Both Linear Regression and DT models performed poorly when calculating the actual values. The difference between these two ML models was insignificant since neither could efficiently provide valuable predictions for a given dataset (Table 8).

Despite its strong performance, there is still room for optimizing the Random Forest model, especially in predicting extreme mortality events. DTs faced issues with instability and overfitting, while linear regression struggled with capturing non-linear relationships. Future research will focus on developing more advanced ML models to address these challenges and improve the prediction of complex, non-linear relationships in fish mortality rates.

The overall insights provided in this study align with the goals of increasing operational effectiveness in aquaculture and contributing to achieving the United Nations Sustainable Development Goals (SDGs) and Disability [40–42] by promoting more sustainable and efficient practices in the industry.

6. Conclusions

This study successfully achieved its research objectives by evaluating the performance of various ML models for predicting fish mortality rates in aquaculture. The Random Forest model outperformed other ML techniques, achieving the highest accuracy and explaining a significant portion of the data variance. In contrast, Linear Regression and DTs encountered limitations, particularly with complex and extreme data points.

The key findings identified essential factors influencing fish mortality, including water quality, feeding rates, and temperature. This knowledge offers actionable insights for improving fish health and reducing stock losses in aquaculture settings [39,43].

Predictive visualizations confirmed that Random Forest most accurately aligned with the observed outcomes, whereas Linear Regression and DTs faced substantial errors. These insights suggest the potential for refining predictive models through advanced techniques, such as Deep Learning and Gradient Boosting, to enhance the prediction accuracy for complex relationships. Incorporating real-time data could further support these models' practical applications in aquaculture management [41].

These findings reveal the need for robust decision-making tools within the aquaculture industry. By implementing advanced models like Random Forest, sector managers can make data-informed decisions that optimize key operational factors, ultimately enhancing fish populations and ecological sustainability [43,44].

Overall, this study demonstrates the utility of ML in advancing aquaculture management and provides both practical recommendations and theoretical insights for the field.

Author Contributions: Conceptualization, M.C.G. and S.S.; methodology, M.C.G. and G.V.; software, M.C.G.; validation, M.C.G. and G.V.; formal analysis, M.C.G.; investigation, M.C.G., G.V. and D.C.G.; resources, J.A.T.; data curation, M.C.G. and D.C.G.; visualization, M.C.G. and D.C.G.; writing—original draft, M.C.G. and D.C.G.; writing—review and editing, M.C.G., D.C.G., G.V., J.A.T. and S.S.; supervision, S.S.; project administration, J.A.T.; funding acquisition, J.A.T. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the EU-Greece Operational Program of Fisheries (EPAL) 2014–2020, grant number (MIS) 5067321.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study is available on request from the corresponding author. The data is not publicly available due to copyright restrictions from the private company that owns it.

Acknowledgments: This work is supported by the action "Improving Competitiveness of the Greek Fish Farming through Development of Intelligent Systems for Disease Diagnosis & Treatment Proposal and Relevant Risk Management Supporting Actions", (MIS) 5067321.

Conflicts of Interest: The author Marios C. Gkikas is the founder and owner of the company OWEB Digital Experience. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

Appendix A. Data Preparation Using Cassandra

This section outlines the steps involved in setting up a database for managing fishery data using Cassandra, from creating a keyspace to importing data from a CSV file.

Table A1 describes the creation of the fishery keyspace, which serves as a dedicated namespace for the tables storing data related to the case study.

Table A2 shows the command to select a keyspace, ensuring all subsequent operations are applied within the correct context.

Table A3 details the structure of the fish_data table, which stores multiple attributes related to fishery operations, including metadata such as location, sample_code, state, and other essential parameters like deaths and medication.

Table A4 provides the command for importing data from a CSV file into the fish_data table. This imported data will then be analyzed in the subsequent sections using Apache Spark.

Table A1. Create a keyspace.

```
CREATE KEYSPACE IF NOT EXISTS fishery WITH replication = {'class': 'SimpleStrategy',
'replication_factor': '1'};
```

Table A2. Select keyspace.

```
USE fishery;
```

Table A3. Create the table.

```
CREATE TABLE IF NOT EXISTS fish_data (
location text,
sample_code text,
day text,
description text,
state text,
analysis text,
parameter text,
value double,
unit text,
limit double,
aa_x text,
month text,
cell text,
portion text,
fishno int,
MAB text,
fish_no int,
deaths int,
corrections int,
nofishing text,
damage text,
kg_fishing double,
sample text,
medication text,
food text,
food_model text,
SFRpercent double,
temp double,
vol double,
i_f text,
medicine text,
aa_y text,
PRIMARY KEY (sample_code)
);
```

Table A4. Import the data from the CSV file.

```
COPY fish_data (location, sample_code, day, description, state, analysis, parameter, value, unit,
limit, aa_x, month, cell, portion, fishno, MAB, fish_no, deaths, corrections, nofishing, damage,
kg_fishing, sample, medication, food, food_model, SFRpercent, temp, vol, i_f, medicine, aa_y)
FROM '/path/to/your/file.csv' WITH DELIMITER=';' AND HEADER=TRUE;
```

Appendix B. Data Processing with Apache Spark

After setting up the Cassandra database, we used Apache Spark to load and process the data.

Table A5 describes how to create a Spark session that connects to Cassandra. This step is crucial for ensuring that Spark can access the fish_data table.

Table A6 demonstrates the process of loading the data from Cassandra into a Spark DataFrame, which allows for the efficient processing and analysis of large datasets.

Table A7 verifies the successful loading of data by displaying a sample of the data. This step is important for checking data integrity before applying machine learning algorithms.

Table A5. Create an Apache Spark session.

```
from pyspark.sql import SparkSession
spark = SparkSession.builder I am running a few minutes late; my previous meeting is running over.


.appName('Fisheries')\
.config('spark.cassandra.connection.host', 'localhost')\
.config('spark.cassandra.connection.port', '9042')\
.config('spark.jars.packages', 'com.datastax.spark:spark-cassandra-connector_2.12:3.0.1')
\
.getOrCreate()
```

Table A6. Loading data from Cassandra.

```
df = spark.read.format("org.apache.spark.sql.cassandra")\
.options(table="fish_data", keyspace="fishery")\
.load()
```

Table A7. Data load verification.

```
df.show()
```

Appendix C. Machine Learning Models

In this section, we applied various machine learning models to predict fish mortalities.

Table A8 presents the Python code for implementing a Random Forest model. This model predicts fish mortality rates based on various features. The feature importance and predictions vs. true values are visualized to provide insight into the model's performance.

Table A9 outlines the implementation of a Decision Tree Regressor. Like the Random Forest model, the predictions are compared to actual values, and key evaluation metrics are printed to assess model accuracy.

Table A10 describes the use of Linear Regression, focusing on categorical and numerical features. The pipeline approach combines data preprocessing and model training into a single workflow. The model's coefficients are also visualized to interpret the influence of different features on the predicted outcomes.

Table A8. Random Forest Python code.

```
# Calculate and print the metrics for the Random Forest model
rf_mse = mean_squared_error(y_test, rf_predictions)
rf_r2 = r2_score(y_test, rf_predictions)
rf_cindex = concordance_index(y_test, rf_predictions)
# Plot feature importances for the Random Forest model
plot_feature_importances(rf_regressor, 'Random Forest', X_train.columns)
# Create a scatter plot for the Random Forest predictions vs. true values
plt.figure(figsize=(10, 6))
plt.scatter(y_test, rf_predictions, alpha=0.5, color='green', label='Predictions')
plt.plot(y_test, y_test, color='red', label='Actual Values')
plt.title('Random Forest Predictions vs. True Values')
plt.xlabel('True Values')
plt.ylabel('Predictions')
plt.legend()
plt.tight_layout()
plt.show()
# Print the evaluation metrics
rf_mse, rf_r2
```

Table A9. Decision Tree Python code.

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
import matplotlib.pyplot as plt
# Initialize the Decision Tree Regressor model
decision_tree_model = DecisionTreeRegressor(random_state=42)
# Fit the model to the training data
decision_tree_model.fit(X_train, y_train)
# Predict on the testing data
dt_predictions = decision_tree_model.predict(X_test)
# Calculate metrics
dt_mse = mean_squared_error(y_test, dt_predictions)
dt_rmse = mean_squared_error(y_test, dt_predictions, squared=False)
dt_mae = mean_absolute_error(y_test, dt_predictions)
dt_r2 = r2_score(y_test, dt_predictions)
dt_cindex = concordance_index(y_test, dt_predictions)
# Detailed report
dt_report = f"""
Decision Tree Regression Model Report:
Mean Squared Error (MSE): {dt_mse:.2f}
Root Mean Squared Error (RMSE): {dt_rmse:.2f}
Mean Absolute Error (MAE): {dt_mae:.2f}
R-squared (R2): {dt_r2:.2f}
"""
# Print the report
print(dt_report)
# Plot the prediction vs. actual values
plt.figure(figsize=(10, 6))
plt.scatter(y_test, dt_predictions, color='blue', label='Predictions', alpha=0.6)
plt.plot(y_test, y_test, color='red', label='Actual', linewidth=2)
plt.title('Decision Tree Predictions vs. True Values')
plt.xlabel('True Values')
plt.ylabel('Predictions')
plt.legend()
plt.show()
```

Table A10. Linear Regression Python code.

```
from pyspark.ml.feature import StringIndexer, OneHotEncoder, VectorAssembler
from pyspark.ml.regression import LinearRegression
from pyspark.ml import Pipeline
# Define StringIndexer and OneHotEncoder for categorical columns
categorical_columns = ['location', 'day', 'parameter', 'cell', 'MAB', 'food', 'i_f']
indexers = [StringIndexer(inputCol=column, outputCol=column+"_index") for column in categorical_columns]
encoders = [OneHotEncoder(inputCol=column+"_index", outputCol=column+"_encoded") for column in categorical_columns]
# Define VectorAssembler
features = [column+"_encoded" for column in categorical_columns] + ['value', 'fishno', 'SFRpercent', 'temp', 'vol']
assembler = VectorAssembler(inputCols=features, outputCol="features")
# Define Linear Regression model
lr = LinearRegression(featuresCol='features', labelCol='deaths')
# Define Pipeline
pipeline = Pipeline(stages=indexers + encoders + [assembler, lr])
# Load data from Cassandra
df = spark.read.format("org.apache.spark.sql.cassandra").options(table="fish_data", keyspace="fishery").load()
# Fit the model
```

Table A10. *Cont.*

```
model = pipeline.fit(df)
# Make predictions
predictions = model.transform(df)
lr_predictions = predictions.select('prediction').toPandas().values.flatten()
y_test = predictions.select('deaths').toPandas().values.flatten()
# Calculate and print the performance metrics
lr_mse = mean_squared_error(y_test, lr_predictions)
lr_r2 = r2_score(y_test, lr_predictions)
lr_cindex = concordance_index(y_test, lr_predictions)
# Show the coefficients of the model
print("Coefficients: " + str(model.stages[-1].coefficients))
print("Intercept: " + str(model.stages[-1].intercept))
# Plot the results
predictions.select('deaths', 'prediction').toPandas().plot(kind='scatter', x='deaths', y='prediction')
# Plot the coefficients
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
coefficients = pd.DataFrame({
'Feature': np.append(np.array(features), 'Intercept'),
'Coefficient': np.append(model.stages[-1].coefficients.toArray(), model.stages[-1].intercept)
})
coefficients.sort_values('Coefficient').set_index('Feature').plot(kind='barh', legend=False)
plt.title('Linear Regression Coefficients')
```

References

1. FAO. Available online: https://www.fao.org/3/ca9229en/ca9229en.pdf (accessed on 12 June 2024).
2. Darmaraki, S.; Denaxa, D.; Theodorou, I.; Livanou, E.; Rigatou, D.; Raitsos, E.D.; Stavrakidis-Zachou, O.; Dimarchopoulou, D.; Bonino, G.; Mcadam, R.; et al. Marine Heatwaves in the Mediterranean Sea: A Literature Review. *Mediterr. Mar. Sci.* **2024**, *25*, 586–620. [CrossRef]
3. Cheung, W.W.L.; Lam, V.W.Y.; Sarmiento, J.L.; Kearney, K.; Watson, R.; Pauly, D. Projecting global marine biodiversity impacts under climate change scenarios. *Fish Fish* **2009**, *10*, 235–251. [CrossRef]
4. Naylor, R.; Goldburg, R.; Primavera, J.; Kautsky, N.; Beveridge, M.C.M.; Clay, J.; Folke, C.; Lubchenco, J.; Mooney, H.; Troell, M. Effect of aquaculture on world fish supplies. *Nature* **2000**, *405*, 1017–1024. [CrossRef] [PubMed]
5. Gkikas, D.C.; Gkikas, M.C.; Theodorou, J.A. Fostering Sustainable Aquaculture: Mitigating Fish Mortality Risks Using Decision Trees Classifiers. *Appl. Sci.* **2024**, *14*, 2129. [CrossRef]
6. Stentiford, G.D.; Neil, D.M.; Peeler, E.J.; Shields, J.D.; Small, H.J.; Flegel, T.W.; Vlak, J.M.; Jones, B.; Morado, F.; Moss, S.; et al. Disease will limit future food supply from the global crustacean fishery and aquaculture sectors. *J. Invertebr. Pathol.* **2012**, *110*, 141–157. [CrossRef] [PubMed]
7. Klinge, D.; Naylor, R. Searching for solutions in aquaculture: Charting a sustainable course. *Annu. Rev. Environ. Resour.* **2012**, *37*, 247–276. [CrossRef]
8. Ubina, N.A.; Lan, H.-Y.; Cheng, S.-Y.; Chang, C.-C.; Lin, S.-S.; Zhang, K.-X.; Lu, H.-Y.; Cheng, C.-Y.; Hsieh, Y.-Z. Digital twin-based intelligent fish farming with Artificial Intelligence Internet of Things (AIoT). *Smart Agric. Technol.* **2023**, *5*, 100285. [CrossRef]
9. Silva, L.C.B.d.; Lopes, B.D.M.; Blanquet, I.M.; Marques, C.A.F. Gaussian distribution model for detecting dangerous operating conditions in industrial fish farming. *Appl. Sci.* **2021**, *11*, 5875. [CrossRef]
10. Nahar, J.; Sharma, N.A.; Kumar, K.; Prasad, A.; Kumar, A. Fishermen's expert views on the causes of fish poisoning in fiji: An investigation through data mining technique. In Proceedings of the 2017 4th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), Nadi, Fiji, 10–12 December 2017; pp. 99–105.
11. Bruna, D.M.L.; Silva, L.C.B.; Blanquet, I.M.; Georgieva, P.; Marques, C.A.F. Prediction of fish mortality based on a probabilistic anomaly detection approach for recirculating aquaculture system facilities. *Rev. Sci. Instrum.* **2021**, *92*, 025119. [CrossRef]
12. Probst, W.N. How emerging data technologies can increase trust and transparency in fisheries. *ICES J. Mar. Sci.* **2020**, *77*, 1286–1294. [CrossRef]
13. Muhamedyev, R.; Yakunin, K.; Iskakov, S.; Sainova, S.; Abdilmanova, A.; Kuchin, Y. Comparative analysis of classification algorithms. In Proceedings of the 2015 9th International Conference on Application of Information and Communication Technologies (AICT), Rostov on Don, Russia, 14–16 October 2015; pp. 96–101. [CrossRef]
14. Karim, M.; Rahman, R.M. Decision Tree and Naïve Bayes Algorithm for Classification and Generation of Actionable Knowledge for Direct Marketing. *J. Softw. Eng. Appl.* **2013**, *6*, 196–206. [CrossRef]

15. Wu, Z.; Zhang, J.; Hu, S. Review on Classification Algorithm and Evaluation System of Machine Learning. In Proceedings of the 13th International Conference on Intelligent Computation Technology and Automation (ICICTA), Xi'an, China, 24–25 October 2020; pp. 214–218. [CrossRef]
16. Yadav, K.; Thareja, R. Comparing the Performance of Naive Bayes and Decision Tree Classification Using R. *Int. J. Intell. Syst. Appl.* **2019**, *11*, 11–19. [CrossRef]
17. Rahmadani, S.; Dongoran, A.; Zarlis, M.; Zakarias. Comparison of Naive Bayes and Decision Tree on Feature Selection Using Genetic Algorithm for Classification Problem. *J. Phys. Conf. Ser.* **2018**, *978*, 012087. [CrossRef]
18. Powers, D.M.W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. *J. Mach. Learn. Technoly* **2011**, *2*, 37–63.
19. Bostock, J.; McAndrew, B.; Richards, R.; Jauncey, K.; Telfer, T.; Lorenzen, K.; Little, D.; Ross, L.; Handisyde, N.; Gatward, I.; et al. Aquaculture: Global status and trends. *Philos. Trans. R. Soc. B* **2010**, *365*, 2897–2912. [CrossRef]
20. Tacon, A.G.J.; Metian, M. Global overview on the use of fish meal and fish oil in industrially compounded aquafeeds: Trends and future prospects. *Aquaculture* **2008**, *285*, 146–158. [CrossRef]
21. FishAI. Available online: http://fishai.gr (accessed on 22 July 2024).
22. Vonitsanos, G.; Kanavos, A.; Mylonas, P.; Sioutas, S. A NoSQL Database Approach for Modeling Heterogeneous and Semi-Structured Information. In Proceedings of the 9th International Conference on Information, Intelligence, Systems and Applications (IISA), Zakynthos, Greece, 23–25 July 2018; pp. 1–8. [CrossRef]
23. Zaharia, M.; Chowdhury, M.; Franklin, M.J.; Shenker, S.; Stoica, I. Spark: Cluster computing with working sets. In Proceedings of the HotCloud 2010: 2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '10), Boston, MA, USA, 22 June 2010; Volume 2, p. 10.
24. Lakshman, A.; Malik, P. Cassandra: A decentralized structured storage system. ACM SIGOPS Oper. *Syst. Rev.* **2010**, *44*, 35–40.
25. Oracle Corporation. *Oracle VM VirtualBox User Manual*; Oracle Corporation: Redwood City, CA, USA, 2021.
26. Apache Spark. Available online: https://spark.apache.org/ (accessed on 18 July 2024).
27. Oracle. Available online: https://www.oracle.com/java/technologies/downloads/ (accessed on 18 July 2024).
28. Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montréal, QC, Canada, 20–25 August 1995; Volume 2, pp. 1137–1143.
29. Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
30. Quinlan, J.R. Induction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
31. Russel, S.; Norvig, P. Artificial Intelligence. In *A Modern Approach*, 3rd ed.; Prentice Hall: Hoboken, NJ, USA, 2003.
32. Witten, I.; Frank, E.; Hall, M. *Data Mining*; Morgan Kaufmann Publishers: Burlington, MA, USA, 2011.
33. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2009; ISBN 0387848576.
34. Matplotlib. A Plotting Library for Python and Its Numerical Mathematics Extension, NumPy. It Provides an Object-Oriented API for Embedding Plots into Applications. 2023. Available online: https://matplotlib.org/stable/users/index.html (accessed on 25 June 2024).
35. NumPy. A Library for the Python Programming Language, Adding Support for Large, Multi-Dimensional Arrays and Matrices, along with Mathematical Functions to Operate on These Arrays. 2023. Available online: https://numpy.org/doc/stable/ (accessed on 25 June 2024).
36. Pandas. A Powerful and Flexible Open-Source Data Analysis and Manipulation Library for Python. It Was Used to Read, Clean, and Manipulate the Data. 2023. Available online: https://pandas.pydata.org/docs/ (accessed on 25 July 2024).
37. Scikit-Learn. A Machine Learning Library in Python, Built on NumPy, SciPy, and Matplotlib. It Was Used for Linear Regression and Correlation Analysis. 2023. Available online: https://scikit-learn.org/stable/index.html (accessed on 25 June 2024).
38. Seaborn. A Data Visualization Library Based on Matplotlib, Providing a Higher-Level Interface for Drawing Attractive and Informative Statistical Graphics. 2023. Available online: https://seaborn.pydata.org/ (accessed on 25 June 2024).
39. Garcia, S.M.; Rice, J.; Charles, A. Governance of Marine Fisheries and Biodiversity Conservation. In *Governance of Marine Fisheries and Biodiversity Conservation*; Garcia, S.M., Rice, J., Charles, A., Eds.; Wiley: Hoboken, NJ, USA, 2014. [CrossRef]
40. FAO. Contributing to Food Security and Nutrition for All. The State of World Fisheries and Aquaculture. 2016. Available online: https://www.fao.org/3/i5555e/i5555e.pdf (accessed on 12 December 2023).
41. Worm, B.; Hilborn, R.; Baum, J.K.; Branch, T.A.; Collie, J.S.; Costello, C.; Fogarty, M.J.; Fulton, E.A.; Hutchings, J.A.; Jennings, S.; et al. Rebuilding Global Fisheries. *Science* **2009**, *325*, 578–585. [CrossRef]
42. United Nations. The Sustainable Development Goals (SDGs) and Disability. 2015. Available online: https://social.desa.un.org/issues/disability/news/the-sustainable-development-goals-sdgs-and-disability (accessed on 25 June 2024).
43. IPCC. *Global Warming of 1.5 °C*; Intergovernmental Panel on Climate Change: Geneva, Switzerland, 2018.
44. Uno, H.; Cai, T.; Pencina, M.J.; D'Agostino, R.B.; Wei, L.J. On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Stat. Med.* **2011**, *30*, 1105–1117. [CrossRef] [PubMed]