A REPORT
ON

Python Developer in AI & ML

Submitted by,

SAGAR M KODABAGI - 20211ISE0011

Under the guidance of,

Mr. Saptarsi Sanyal

in partial fulfilment for the award

of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION SCIENCE AND ENGINEERING

At

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Internship report **PYTHON DEVELOPER IN AI & ML** being submitted by SAGAR M KODABAGI bearing roll number 20211ISE0011 in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Information Science and Engineering is a bonafide work carried out under my supervision.

Mr. Saptarsi Sanyal
Assistant Professor
PSCS
Presidency University

Dr. Pallavi R
Head of Department
PSCS & IS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
PSCS
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean – PSCS & ISSS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

I hereby declare that the work, which is being presented in the project report entitled **PYTHON DEVELOPER IN AI & ML** in partial fulfilment for the award of Degree of **Bachelor of Technology** in Information Science and Engineering, is a record of my own investigations carried under the guidance of Mr. Saptarsi Sanyal, Assistant Professor, **Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name	Roll Number	Signature
SAGAR M KODABAGI	20211ISE0011	

INTERNSHIP COMPLETION CERTIFICATE

MIZZEN DIGITAL PRIVATE LIMITED

Registered Office: C-1 (A), Bagwe Bhavan, Jayprakash Nagar Rd No.1, Goregaon East, Mumbai - 400063
Corporate Office: DBS Business Centre, B-Wing, Kanakia Wall Street, Andheri (E), Mumbai - 400093
+91 22 48809984 // info@mizzendigital.com // www.mizzendigital.com
CIN - U74999MH2021PTC354011

MIZZEN DIGITAL

Dated: 28th April 2025

Mr. Sagar M Kodabagi
Presidency University
Bengaluru, Karnataka
Roll No- 20211ISE0011

Subject: Internship Completion Certificate

This is to certify that **Mr. Sagar Kodabagi**, a student at **Presidency University**, has successfully undergone his internship with the **Technology Team** at **Mizzen Digital Pvt Ltd.** from **27th January 2025** to **27th April 2025**.

During his internship, Sagar demonstrated strong analytical and problem-solving skills while working on key challenges in the marine industry. His primary focus was on research and development projects, including the development of operational calculatiors and AI bot.

Most of the tasks assigned required extensive R&D, and Sagar approached them with curiosity and diligence. He was able to navigate complex problems and deliver functional solutions, contributing meaningfully to our ongoing efforts in digital innovation for maritime operations.

Sagar's performance reflects a solid foundation in technical skills and a proactive attitude toward learning and experimentation. **Mr. Sagar** exhibited professionalism, enthusiasm, and a willingness to learn, and contributed positively to our team.

We appreciate the time and efforts put in by **Mr. Sagar** and wish him success in all future endeavours.

Issued on: 28th April 2025
Location: Mumbai, India

For **Mizzen Digital Private Limited**

Authorized Signatory

ISO 27001:2013 Certified by IRQS

ISO 9001:2015 Certified by IRQS

ABSTRACT

The increasing demand for real-time environmental data has spurred the development of intelligent systems to interpret weather factors effectively. The Weather Factor API is an innovative predictive platform designed to analyze and forecast outcomes based on key meteorological parameters. Utilizing machine learning techniques, specifically the XGBoost regression model, this system processes environmental features to provide accurate and actionable predictions for downstream applications.

The API empowers developers and analysts to integrate predictive weather analytics into broader applications such as agriculture, logistics, and climate-aware planning. It accepts input features related to weather conditions and returns reliable estimates of target variables, facilitating proactive decision-making. The system is trained on synthesized or real-world datasets and employs modern data preprocessing and validation techniques to ensure robust performance.

Key functionalities include efficient data ingestion, model training using gradient boosting, and serialization for deployment through a lightweight, responsive API. The model's performance is evaluated using metrics like Mean Absolute Error (MAE) and Mean Squared Error (MSE), ensuring it meets expected accuracy standards. This modular design allows for easy retraining with new data, enhancing adaptability to changing climatic patterns.

By automating the prediction process and encapsulating complex computations into a simple interface, the Weather Factor API reduces the technical burden on users while improving forecast reliability. This project represents a significant step towards data-driven environmental intelligence, offering a scalable and efficient solution for weather-based predictive modeling.

The architecture of the Weather Factor API emphasizes modularity and scalability. It includes a FastAPI-based backend for managing input requests, performing real-time weather data retrieval from external APIs, computing intermediate waypoints, and calculating weather resistance factors using vessel and environmental attributes. The use of external services like Open-Meteo and OpenWeather ensures the system is fed with reliable and up-to-date meteorological data, which is then translated into meaningful insights using domain-specific formulas and machine learning inference.

Furthermore, the system supports adaptive learning by enabling retraining of the XGBoost model with new or domain-specific datasets, making it suitable for evolving applications such as maritime route optimization, energy efficiency modeling, or weather risk assessment.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair**, Presidency School of Computer Science and Engineering, Presidency University, and Dr. Pallavi R, Head of the Department, Presidency School of Computer Science Engineering and Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Mr. Saptarsi Sanyal, Assistant Professor,** Presidency School of Computer Science and Engineering and Reviewer **Ms. Monisha Gupta, Assistant Professor**, Presidency School of Computer Science and Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 Internship Coordinators **Mr. Md Zia Ur Rahman and Dr. Sampath A K,** department Project Coordinators **Mr. Srinivas Mishra** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

SAGAR M KODABAGI - 20211ISE0011

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

Within the scope of maritime travel, the impact of environmental conditions is not only understood, but has become essential to a vessel's operational speed, fuel use, and safety. Wind speed, wave height, and currents are but three of many variables that can either help or harm a vessel's performance. Historically, seafarers relied on skilled navigators to compute routes based on cyclic patterns derived from historical data and weather prognostics. However, the increasing climate unpredictability combined with the digitized era of maritime navigation has rendered human selected solutions incredibly futile. Fleet managers and navigators are constantly burdened with an overwhelming amount of meteorological data collected from a variety of platforms. Each platform has different rates of data refreshment and varying levels of precision, further complicating the situation. Mixing datasets without precise modeling results in static outputs stripped of dynamism, making these snapshots of ever-changing reality lose relevance over time. Consequently, route plans can be generated on incomplete or out-of-date information, leaving vessels vulnerable to avoidable delays, fuel losses, and even safety risks. In an industry where timing and cost of operation are of critical concern, this lack of utilization of data is a significant impediment to efficiency.

Identifying these issues, this project presents a Weather Factor API—a smart, machine learning-based system that automatically measures and predicts environmental resistance. The system puts meteorological data in real time together with vessel-specific factors into an equation to arrive at a measurable value known as the "weather factor," which is a measure of the environmental resistance a ship is likely to encounter on a journey. Fundamentally, the API uses XGBoost, an aggressive ensemble learning algorithm, to predict weather factors in the future with great accuracy. By training the algorithm on historical and real-time data, the API enables dynamic and informed route planning under dynamic ocean conditions that constantly change.

The API is not intended solely to automate current processes but to radically change the entire approach to maritime route planning. With its ease of use interface and modular framework, the Weather Factor API blends well with available logistics software shipped by shipping agencies, port organizations, and self-governing vessel systems. The solution does away with most of the mental load on mariners, and they can give their undivided attention to strategic choices as automated systems conduct weather analysis for them. Finally, this project is a significant step towards modernizing and optimizing maritime navigation for a future where data and automation take center stage.

1.1 Challenges in Traditional Weather Impact Assessment for Maritime Operations

The present practices employed to evaluate impacts of weather on maritime navigation, though operational, are found increasingly insufficient for the requirements of contemporary shipping. Some particular issues illustrate the shortcomings of conventional usage of weather data:

• Manual Data Collection:

Maritime route planning remains reliant on the manual gathering of weather information from many meteorological sources. This is a time-consuming process involving significant domain knowledge. Operators need to decipher numerous charts, bulletins, and satellite feeds, frequently with different formats and vocabularies. This piecemeal strategy creates inconsistencies in applying and interpreting data.

• Inconsistent Forecast Accuracy:

Weather information from various sources can differ in accuracy and frequency of update. Such discrepancies create uncertainty when predictions provide contradictory advice, resulting in poor routing choices or excessive reliance on stale information. Since weather can change swiftly, particularly in unstable areas such as the North Atlantic or South China Sea, the capacity to make swift and decisive action is most critical.

• **Delayed Decision-Making:**

Without automated analysis and interpretation capabilities for weather information, much of the responsibility to convert weather inputs into actionable plans rests with the navigators or operations personnel who have to manually interpret weather inputs. This delay in decision-making is especially costly during emergencies or when voyage conditions change en route, as it can lead to lost opportunities for optimization.

• **Lack of Predictive Insight:**

Probably the biggest disadvantage of classical methods is their failure to give forward-looking information. Although predictions might give generalized expectations of weather, they don't give quantitative evaluations of how the conditions will influence the performance of an individual vessel. This leaves operators having to make educated guesses, thereby raising operational risk.

1.2 The Need for Automation in Weather-Based Route Optimization

To surpass the disadvantages of the conventional approach, there exists an urgent need for systems that can process, analyze, and predict weather-related information in real time. Automation provides a number of strategic benefits:

• **Real-Time Analysis:**

Whereas human inspections, automated systems are able to consume and analyze live streams of data in real-time. Such ability makes on-the-fly recalculation of routes in response to sudden weather changes possible, such that vessels are always on the most efficient and safest course possible.

• **Enhanced Reliability:**

Machine learning models can be taught to analyze data consistently with respect to time and geography. Standardization of the variability caused by human judgment and enabling more consistent decision-making across fleet operations.

• **Efficiency in Planning:**

By predicting environmental resistance on certain routes, the system allows operators to foresee slowdowns and fuel inefficiencies. This predictive capability assists in more precise scheduling, fuel budgeting, and risk management.

• **Scalability Across Routes:**

One of the key benefits of automated analysis is the scalability. The system is able to analyze dozens or even hundreds of waypoints or port pairs in one go, something that would be impossible or very impractical if done by hand. This creates new possibilities for large shipping companies operating large-scale global logistics.

1.3 Why Use Machine Learning for Weather Factor Prediction?

Machine learning, and specifically advanced models like XGBoost, provides a robust alternative to statistical estimation and rule-based forecasting. These algorithms are best at identifying patterns in extensive multivariate datasets and making accurate predictions from real-world data inputs.

• Complex Pattern Recognition:

Weather dynamics are nonlinear and dependent. For instance, the joint influence of wind direction, wave swell, and current speed on ship drag is difficult to model by classical linear equations. Machine learning models are capable of capturing nonlinear interactions and learn from actual instances in order to make better predictions in the future.

• Adaptability:

One of the key strengths of ML is its ability to adapt. The more data there is whether from new expeditions, other types of vessels, or abnormal weather conditions the model can be retrained for increased accuracy. This allows the system to be resilient in the presence of long-term changes in the environment or the implementation of new ship models.

• **Quantitative Output:**

The Weather Factor API not only indicates possible problems, but also gives a specific, numeric "weather factor" value. This value is the amount of resistance to be expected and can be used directly in subsequent analysis, e.g., voyage time calculations or fuel consumption estimates.

• **Integration with APIs:**

By placing ML models within APIs, the prediction results become readily available to a broad variety of maritime applications. These integrations allow operators to get real-time predictions within the applications they already use, facilitating usability and minimizing training overhead.

1.4 Project Objectives

This project will create a modular and smart API that can compute and predict environmental resistance in maritime travel. The primary goals of the Weather Factor API are as follows:

• **Accurate Prediction:**

Apply ML models to approximate the influence of weather conditions on ship movement, representing this effect as a normalized weather factor. The model must provide high accuracy even with shifting conditions and across different ship specifications.

• **Multi-Source Data Handling:**

Integrate environmental information from reliable meteorological sources. This comprises wave reports, wind patterns, and current charts from global public APIs such as NOAA, ECMWF, or other sea-oriented services.

• End-User Usability:

Make the API interface developer-friendly and intuitive. This involves simple documentation, quick response times, and straightforward deployment options. The API must be plug-and-play with current maritime routing tools with little customization.

1.5 Scope of the Weather Factor API

The scope of the project involves both technical and functional ones:

• **Core Features:**

The API is created to compute a dynamic weather factor score from current and predicted environmental inputs. It also has a scaling system to provide comparability across routes and ships, and can predict weather resistance at future time intervals.

• Target Use Case:

First and foremost, the API is optimized for commercial shipping routes, where fuel economy and delivery times are paramount. Still, the design makes it feasible for future extension into other maritime arenas like fishing, naval operations, or autonomous surface ships.

• Extensibility:

The modular design of the system allows for future integration of features. Future extensions in planning are modules for ice detection, emission control area (ECA) notification, and real-time ingestion of satellite images for improved forecasting.

• Technical Constraints:

The project acknowledges its third-party weather API dependencies, which can be different in terms of granularity and update rate. It also accounts for constraints in ML generalization, e.g., suboptimal performance in hitherto unknown oceanic areas or unusual weather conditions.

1.6 Significance of the Project

The Weather Factor API meets a growing critical requirement for smart tools in the shipping sector. With environmental conditions growing more extreme and shipping routes increasingly crowded, decision-making on data is not only a business necessity but an imperative for safety.

• Operational Efficiency:

With more efficient route planning capability, operators can cut fuel consumption by a considerable amount, leading to cost savings as well as emissions reduction. This supports IMO decarbonization goals and other sustainability requirements.

• Data-Driven Decision Support:

By converting raw meteorological data into actionable metrics, the API enables stakeholders to make informed decisions. Whether choosing an optimal departure time or steering clear of high-risk zones, the system improves situational awareness and confidence.

• Driving Maritime Tech:

This initiative forms part of an even larger drive toward smart shipping solutions. By merging machine learning, real-time information, and new API architecture, it advances the future of autonomous and semi-autonomous navigation for vessels.

• Impact to Climate-Conscious Routing:

The API facilitates green routing approaches through highlighting environmentally expensive routes. Such functionality makes it an asset to companies looking to align profitability with conscience.

• User Role Customization:

The API will accommodate various user roles (e.g., ship operators, logistics coordinators, and analysts), providing customized outputs and levels of access. This means that every stakeholder will get only the appropriate information required for their role, enhancing usability and operational concentration.

• Cross-Platform Compatibility:

Developed for integration with multiple platforms ranging from desktop routing tools, mobile apps, to cloud-based fleet management dashboards the API provides accessibility and usability in various operational settings.

CHAPTER-2

LITERATURE REVIEW

Table 2.1 Literature Review

Author(s)	Year	Title	Key Findings
Kim, J., & Yoon, S.	2017	Weather-Based Route Optimization for Commercial Vessels	Demonstrates how integrating weather data into route planning reduces fuel usage and voyage time.
Liu, M., & Zhang, T.	2019	Predictive Modeling of Marine Weather Impact Using ML	Explores machine learning techniques, such as regression models, for estimating weather resistance.
Fernandez, D., & Roy, A.	2020	Real-Time Marine Weather Data Integration in Navigation APIs	Highlights the value of API-driven systems for incorporating live environmental data into ship routing.
Sharma, N., & Dey, R.	2021	Application of XGBoost in Maritime Forecasting	Investigates the use of XGBoost for predicting ship behavior under varying weather conditions.
Singh, K., & Balan, M.	2018	Weather Risk Management in Shipping Logistics	Discusses operational risks in maritime transport due to weather and the need for proactive forecasting.
Zhou, L., & Chen, X.	2022	API-Based Systems for Oceanographic Data Processing	Reviews architecture for scalable systems handling real-time marine data through web services.
Rajan, S., & Kulkarni, P.	2020	Machine Learning-Driven Maritime Route Planning	Presents a framework for ML-powered systems that enhance safety and fuel efficiency on maritime routes.
Tanaka, H., & Lee, D.	2019	Impact of Wind and Wave Parameters on Vessel Speed Loss	Analyses how wind and wave metrics affect ship performance and how models can help predict the impact.
Alam, R., & Mehta, V.	2021	Intelligent APIs for Environmental Prediction	Emphasizes the role of intelligent APIs in transforming raw environmental data into usable predictions.
Chen, Y., & Kumar, A.	2022	Data-Driven Decision Support for Maritime Navigation	Highlights the growing role of ML and real-time analytics in automated decision-making for shipping.

Patel, R., & Wong, H.	2020	Adaptive Routing Using Real-Time Satellite Data	Shows how dynamically adjusting ship routes based on satellite weather updates leads to safer and more efficient voyages.
Nakamura, Y., & Singh, R.	2021	Deep Learning for Wave Forecasting in Coastal Waters	Demonstrates the effectiveness of deep learning models in forecasting near-shore wave activity to support coastal navigation.
Costa, M., & Liu, Y.	2018	Enhancing Fuel Efficiency Through Weather-Aware AI Models	Proposes an AI framework that leverages weather patterns to make smarter route decisions, helping ships save fuel.
Jansen, T., & Roy, P.	2019	Vessel Tracking and Predictive Weather Avoidance	Combines AIS tracking data with predictive weather models to anticipate and avoid rough sea conditions.
Zhao, J., & Malik, S.	2022	Integrating IoT Sensors with ML for Maritime Safety	Explores how onboard sensors feeding into ML systems improve the real-time monitoring of vessel responses to harsh weather.
Ahmed, K., & Luo, F.	2020	Smart Navigation Dashboards for Ship Captains	Introduces an interface that synthesizes environmental data into actionable insights, empowering better onboard decision-making.
Huang, L., & Ortega, J.	2021	Multi-Model Forecast Blending for Route Optimization	Shows how combining multiple weather prediction models enhances accuracy in route optimization tools.
Gupta, A., & Khan, T.	2019	Case Study: AI-Based Risk Mitigation in North Atlantic Routes	Analyzes how AI-powered simulations helped avoid delays and damage during winter storms in the Atlantic.
Park, S., & Ibrahim, N.	2021	Cloud-Based Maritime Route Forecasting Platform	Describes a scalable, cloud-hosted system that enables fleet operators to evaluate route conditions in near real-time.
Torres, E., & Dubois, M.	2022	Quantifying Weather Impact on Port Turnaround Times	Provides evidence that incorporating weather predictions into port logistics significantly reduces delays and congestion.

Over the past few years, the shipping sector has seen a dramatic shift, driven to a large extent by the introduction of smart technologies and real-time data fusion. The convergence of environmental sensing and machine learning has created new avenues for innovation in ship navigation, route planning, and operational planning. As ships move through intricate and frequently turbulent ocean conditions, conventional navigation techniques—formerly heavily dependent on human knowledge and static predictions—are being supplemented and replaced by data-centric systems providing real-time information and predictive functionality. This transition is not only technological but strategic, an expression of industry-wide focus on safety, fuel economy, regulatory compliance, and environmental stewardship.

Early foundational research in this field appeared in the late 2010s. Researchers like Kim and Yoon (2017) were among the earliest to emphasize the direct link between optimized route planning and decreases in fuel use and operational expenditures. Their work provided the conceptual basis for automated decision-support systems by demonstrating that the inclusion of real-time weather information in routing calculations could substantially enhance voyage performance. This vantage point pushed back against classic deterministic frameworks that all too frequently discounted the dynamic, nonlinear character of forces in the environment at sea.

Drawing on these initial findings, Liu and Zhang (2019) took the discussion further by using machine learning methods to address the challenge of weather-related resistance. They constructed predictive models that could convert raw environmental inputs—e.g., wind speed, wave height, and current direction—into ordered, quantifiable resistance scores. This innovation is particularly pertinent to our Weather Factor API project, which relies on the premise that environmental resistance can and ought to be calculated dynamically from vessel-specific and time-dependent inputs. Their research confirmed the technical viability of deriving actionable weather metrics from intricate datasets, a central tenet of our system design.

Concurrently with breakthroughs in machine learning, scientists also turned their attention to studying the infrastructure requirements for rolling out such systems on a large scale. API-based platforms took center stage when it came to innovation, especially with regards to maritime software integration. Fernandez and Roy (2020) looked at the vision of intelligent navigation platforms that consume real-time wave and wind inputs through standardized APIs. Their results highlighted the need for frictionless data integration to enhance situational awareness and reduce manual effort. APIs, they contended, not only facilitate streamlined information exchange but also enable scalable deployments that can support a global fleet of ships with different operationals. This focus on scalable infrastructure was further detailed by Zhou and Chen (2022), who discussed the architectural implications of developing marine data services.

Their task was on how to develop strong data pipelines that are able to process frequent weather updates yet keep low-latency responses as a core to the real-time performance needs of our Weather Factor API. They promoted modular, microservice architectures that are capable of processing asynchronous data streams, a strategy that we have emulated in our system design in order to facilitate responsiveness and fault tolerance. One of the key innovations in the maritime AI domain has been the use of ensemble learning models, specifically XGBoost, for predictive analytics.

Sharma and Dey (2021) performed a comprehensive analysis of the applicability of XGBoost for modeling ship behavior against environmental parameters. Their results indicated that XGBoost was more accurate and faster compared to other regression models, especially when trained on time-series weather data. Since our Weather Factor API employs XGBoost as its predictive engine, their work is a strong empirical validation of our method. It also validates the notion that contemporary ensemble methods are well-suited to deal with the multidimensional and frequently noisy datasets characteristic of maritime operations. Supporting this technical basis, Tanaka and Lee (2019) offered a more domain-specific perspective by investigating the specific ways wind and wave conditions affect vessel speed and fuel consumption.

Their research entailed quantitative modeling of hydrodynamic and aerodynamic resistance, providing key insights into how environmental forces affect ship performance. The weather factor scoring mechanism implemented in our API is based on analogous physics-based calculations that are translated into easily interpretable normalized scores that maritime operators can readily understand. Their efforts verified the assumption that while such mappings may be highly complicated under the hood, they can still create easily understandable outputs that enable everyday decisions on the vessel and command centers. Apart from efficiency and cost advantages, safety is also a top concern in the incorporation of intelligent systems into maritime navigation.

Rajan and Kulkarni (2020) believed that machine learning models, having been adequately trained and integrated, can be used as early warning systems, indicating possibly dangerous routes and averting accidents. Their focus on proactive decision support is very much in line with our project goals. By being able to forecast high-weather-resistance regions ahead of time, the Weather Factor API enables users to make route corrections prior to experiencing adverse weather, thus improving onboard safety and overall reliability of voyage. Alam and Mehta (2021) investigated further the place of smart APIs through exploring the extent to which well-constructed APIs could condense complicated environmental inputs into useful and meaningful outputs.

Their case studies showed that with the right abstraction layers, even non-technical users could leverage advanced analytics tools to make strategic decisions. This insight has greatly influenced our user interface design, which prioritizes clarity, responsiveness, and integration simplicity. Our goal, much like theirs, is to democratize access to high-value predictive insights without requiring users to engage with the underlying complexity. Lastly, Chen and Kumar (2022) offered an overall perspective of how data-driven approaches are transforming operational planning in shipping.

They highlighted the growing dependence on predictive analytics not just for navigation but also for wider strategic objectives, including emissions reduction, environmental compliance, and long-term fleet optimization. Their efforts weave together a strong narrative of a seafaring business evolving—one that is quickly adopting real-time data and artificial intelligence to address the challenges of the 21st century. In aggregate, this cumulative body of research forms a strong conceptual and technical foundation upon which to base the Weather Factor API project.

Every work adds to a swelling consensus: that clever systems—particularly those fueled by live environment data and machine learning are not merely augmentations but requirements in contemporary maritime life. The aggregate evidence justifies our project's design decisions, from the application of XGBoost for the prediction of resistance to the utilization of RESTful APIs for external integration and microservice-based architectures for scalability. Most importantly, the literature highlights the practical relevance of our research. By streamlining environmental resistance estimation, we are not merely creating a technological device—we are solving a real-world and urgent problem.

In addition to the underlying research and technological advancements outlined above, there is a larger literature investigating the operational, environmental, and human aspects of weather-informed maritime routing.

With increasingly complex shipping routes especially in busy or environmentally protected waters researchers have been investigating interdisciplinary approaches that integrate oceanography, artificial intelligence, and operations research to more effectively model and control navigational risk. These have unveiled further aspects of weather resilience estimation, especially in the form of voyage forecasting under uncertainty, route adaptability, and emissions reduction. For example, the research by Navarro and Silva (2018) was centered on the incorporation of stochastic models that would allow for the unpredictability of weather along long routes.

Their model enabled a probabilistic comprehension of route feasibility, which is different from the deterministic method that hypothesizes regular weather conditions. This line of inquiry is highly relevant to our API's forecasting capability, which similarly aims to offer multi-day weather resistance estimates with enough granularity to account for day-to-day fluctuations. The idea that weather should be modeled not just as a static input but as a shifting, probabilistic landscape is foundational to any serious attempt at voyage optimization under real-world constraints. To complement this vision, Bakshi and Andersson (2020) investigated how fleet-level route coordination can be improved with mutual weather intelligence.

Their simulation-based study highlighted that ships sailing in adjacent areas could use collective weather information and mutual predictive results to coordinate their planning processes. Although our API is intended for singular route analysis, these results present possible directions for future development like multi-vessel coordination software or fleet-wide resistance tests specifically for shipping firms with large fleets. The environmental impact of weather-aware routes has also been a significant concern.

Peters and Hwang (2021) examined the way resistance modeling can assist with tracking and reducing carbon emissions in commercial shipping. Through linking high-resistance intervals to heightened fuel consumption and emissions levels, they established a basis for embedding weather impact calculations in carbon compliance mechanisms. They helped shape the overall narrative that analytics based on machine learning could simultaneously maximize operation efficiency and play a role towards sustainability objectives. This dual benefit strengthens the value proposition of the Weather Factor API, especially in an era of tightening emissions regulations under frameworks like IMO 2020 and the EU MRV. In the domain of real-time maritime monitoring, Singh and Verma (2021) introduced a cloud-based maritime intelligence platform that continuously updated ship operators on weather anomalies and safety warnings.

Their solution proved the value in near-immediate feedback loops, wherein predictive warnings could be derived from real-time sensor streams and relayed to ships in mere minutes. Although our API is not a complete monitoring platform, it does have that focus on responsiveness and integration. Their effort underlines the understanding that the use of a prediction is not merely in its correctness, but in its timeliness and simplicity to understand. On the technical side, advances in meteorological data fusion have greatly enriched the capabilities of systems such as ours.

For instance, Ito and Nielsen (2020) came up with a hybrid data pipeline that integrated satellite data, in-situ buoy measurements, and forecasting models to generate highly localized marine weather charts. These multi-source methods reflect the data architecture we've implemented in the Weather Factor API, where various sources (e.g., NOAA, ECMWF) are harmonized to produce a composite input layer for the machine learning model. Their focus on redundancy, resolution, and harmonization of datasets shows how data quality will directly determine the strength of machine learning predictions. In addition, the importance of user-centric design in maritime analytics software has been a developing topic of interest.

Patel and Srinivasan (2019) researched AI-based decision support system adoption in maritime environments and discovered that users more frequently valued simplicity of interface, interpretability, and ease of integration than pure computational power. This realization directly guided our choice to construct the API with well-documented, RESTful endpoints, and normalized scoring outputs that reduce technical post-processing requirements. Their research highlights that even the most precise model needs to be usable and easy to use in order to achieve real-world adoption. In the meantime, from a regulatory perspective, researchers such as Torres and Ahmed (2022) have made the case for standardizing weather-resistance metrics in voyage planning software.

They suggested the implementation of a worldwide schema for environmental scoring, enabling interoperability between ship operators' systems, port authorities' systems, and classification societies' systems. Although our Weather Factor API is presently implemented with internal logic and scaling systems, such studies anticipate the direction of future compliance-ready data structures, particularly as international maritime regulations develop to encompass dynamic weather-based performance criteria. Finally, the application of geospatial intelligence in weather resistance analysis has shown great promise.

Studies by Hu and Moreno (2021) integrated GIS layers with predictive environmental models to visualize resistance scores across geographies, revealing how sea lanes vary in risk and efficiency based on location and season. Their application of heatmaps and geospatial overlays has influenced some of our visualization techniques, enabling decision-makers to more effectively contextualize the numeric results of the API using intuitional map-based presentation. Aside from academic and industrial application, cross-research activities among research centers and maritime technology suppliers have also played a critical part in speeding up weather-aware routing innovation.

One such project is the Blue Routes Project (2022), a collaborative effort between oceanographers and software engineers to enhance environmental resistance modeling using open data collaboration. The project generated high-resolution environmental layers from historical and real-time observations, which were then utilized to train machine learning models for adaptive routing. Their results identified the massive worth of community-crafted datasets and the power of open-access platforms to spur innovation in commercial usage. This ethos of open-endedness and interoperability aligns with the vision of the Weather Factor API, which should be modular and extensible for use by developers, shipping companies, and academic researchers. Finally, recent discussions in maritime analytics shifted towards ethical AI and data governance, particularly with respect to safety-critical predictions.

Wu and El-Sayed (2023) warned that black-box models being used in navigation systems could contribute to unforeseen outcomes if not sufficiently audited or explained. They suggested using a hybrid model strategy, combining interpretable physics-based rules and machine learning to preserve traceability and trust. Our project, which has been inspired by this work, features explainability components that enable end-users to trace back resistance scores to particular weather conditions like wave height, wind direction, or sea current. This integration of predictive accuracy and operational transparency enhances the overall trust in the system—ensuring it is not only powerful but responsible and deployable in real-world maritime environments.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Over the past few years, the maritime sector has experienced major developments in predictive modelling, automation, and the use of environmental data. Nevertheless, there is still room for improvement in the current methodologies to evaluate environmental resistance in maritime navigation, as current approaches remain incomplete in some aspects. These shortfalls call for more intelligent, responsive, and integrable solutions especially ones that can provide real-time, vessel-specific predictions at scale. This section discusses the shortcomings in existing methodologies, setting out the underlying issues that the Weather Factor API project aims to tackle.

1. Inadequate Use of Dynamic Weather Inputs

Perhaps one of the most significant flaws in conventional maritime resistance models is their lack of engagement with dynamic, real-time weather feeds. Most systems function on static databases or scheduled weather updates that fail to realize the fast-changing nature of oceanic conditions. This static methodology greatly diminishes prediction accuracy, particularly in areas of high variability, like the North Atlantic or South China Sea. Without continuous integration from various meteorological sources, these models are unable to provide the level of granularity or responsiveness necessary for contemporary route planning. Dynamic integration like grabbing hourly updates from APIs like Open-Meteo is underexploited in the majority of research and commercial solutions, resulting in lost opportunities for optimization.

2. Lack of Vessel-Specific Tailoring

Ship performance is intrinsically linked with its physical characteristics length, draft, beam, displacement, type of engine, and cruising speed, to mention a few. And yet, most resistance estimation tools still perpetuate a one-size-fits-all strategy, implementing generalized assumptions that discount vessel-to-vessel variations. For instance, a deep-draft container ship will react enormously differently to the impact of waves compared to a high-speed ferry with a shallow hull. Ignoring these differences results in poor weather factor predictions and less-than-optimal operational choices. In addition, the inability to accommodate user-defined vessel inputs reduces model flexibility and hinders adoption in a heterogeneous fleet.

3. Underdeveloped Temporal Forecasting Capabilities

Although forecasting models for weather conditions themselves have become more advanced, converting those forecasts into multi-day resistance predictions is underdeveloped. Numerous current tools are able to assess current conditions or provide a straightforward next-day forecast but do not measure up when being challenged to review resistance patterns in a 3–7 days' timespan. Longer voyages, particularly ocean routes, call for advanced route planning ahead by days. Forward-looking resistance modeling deficiency means that shipowners have a tendency to make routes plan ahead with antiquated or over-simplified hypotheses, thus exposing vessels to avoidable inefficiency or security risk. Widening temporal forecasting horizons is necessary to bridge this gap.

4. Oversimplified Environmental Resistance Calculations

Actual sea environments are influenced by a multifaceted interplay of variables—wave height, wave direction, wave period, wind speed, wind direction, and even sea surface temperature in certain situations. Yet, most current resistance models address these variables in isolation or leave some out altogether. This reductionist method does not capture the compound effects that typically characterize real-world resistance situations. For example, a head-on wave with a crosswind generates more resistance than either situation in isolation. Overlooking these interactions can bias resistance scores and lead to suboptimal planning results. Advanced calculations that integrate several parameters into a single metric are still largely lacking.

5. Minimal Use of Machine Learning for Predictive Modeling

Although machine learning has found its way into weather forecasting and ship behavior modeling, its application in environmental resistance prediction is still a developing field. Current tools are mostly rule-based or deterministic equations that fail to adjust to past trends or changing patterns. Machine learning algorithms, particularly ensemble-based techniques such as XGBoost, provide the capability to feed complex data and derive predictive information that traditional approaches cannot. Few naval tools, though, capture this potential for applying it to calculate resistance scores from multivariable combinations of inputs. Subsequently, most available tools then miss the benefit of increased precision, flexibility, and stability offered by machine learning.

6. Lack of Scalable API-Based Architectures

Even though research prototypes that look promising are being explored, these are hardly geared toward practical application. Most of them are deployed in segregated environments restricted to offline usage or standalone applications without making them available as APIs. This puts a hindrance to integration with operational maritime platforms like Electronic Chart Display and Information Systems (ECDIS), route optimizers, or fleet management dashboards. An absence of scalable, RESTful APIs constrains practical utility in active marine operations, where access to real-time data and seamless integration are necessary for adoption.

7. Neglect of Intermediate Waypoints Along Routes

A second essential shortcoming of existing resistance modeling is a too narrow concentration on origin and destination points at the expense of understanding the importance of intermediate waypoints. In practice, sea routes are made up of many path segments, each exposed to varying environmental conditions. Disregarding these differences results in an incomplete resistance profile, which can lead to underestimation or overestimation of fuel consumption, time, or safety hazards. Successful modeling needs to assess environmental resistance not only at the endpoints but along the journey, taking into consideration changing weather patterns as the ship moves forward. Without this, navigation strategies are reactive instead of proactive.

8. Limited Geographic Generalization

Several resistance estimation models are trained or tested on data from a specific geographic area e.g., coastal Europe or East Asia—without requiring these models to generalize well across other marine ecosystems. Consequently, predictions outside of these areas (e.g., polar waters, equatorial shipping routes) may be much less accurate. This regional skew lessens the global applicability of such a tool, leaving operators in under-served areas with limited choices of either using less precise estimates or ditching the tool. An optimally useful model needs to be tested on varied maritime routes and meteorological zones to guarantee consistent operation around the world.

9. Lack of Interpretability in Model Outputs

One of the issues in implementing machine learning to operational fields is that it lacks transparency regarding how predictions are created. Numerous stakeholders in the maritime industry including port authorities and ship captains need explanations as to why a particular prediction is made, particularly when it has an impact on compliance or safety. Most machine learning models in existence today act like "black boxes," providing numerical values without explanation or context. This uninterpretability diminishes user confidence and hinders decision-making. Improving model explainability, such as indicating which variables made the largest contributions to a weather factor score, is an essential step toward making users more confident.

10. Lack of Benchmark Datasets and Evaluation Criteria

Lastly, there is an evident need for open, standardized datasets and uniform evaluation criteria for maritime resistance prediction. Most studies in this field work on proprietary data or domain-specific sources of data, which cannot be accessed publicly. This means the results cannot be reproduced or even compared against a new model, which hampers collaboration, innovation, and advancement. In addition to that, the absence of a globally accepted measure for evaluation complicates the comparison of the performance of various tools and measuring the advancements. The creation of common benchmarks would considerably speed development in this field by promoting openness, cooperation, and competition.

11. Insufficient Attention to Seasonal and Climatic Variability

The vast majority of current models do not take into consideration seasonal climatic changes like the cycles of the monsoons, hurricane periods, or increases in polar ice coverage. They greatly affect wave dynamics, the behavior of the winds, as well as currents within the seas over the duration. By failing to account for these macro-scale differences, current methods tend to present an oversimplified picture of voyage risks. For example, a summer-safe and efficient route can become winter-risky or inefficient due to higher wave heights or storm frequency. To bridge this gap, climatological datasets and season-aware predictive logic need to be incorporated into the resistance modeling process.

12. Weak Feedback Loops for Model Improvement

Existing maritime resistance estimation software infrequently has feedback loops that learn from past voyage results. After a forecast is generated and a route is traversed, the model is not generally updated with actual performance data (e.g., fuel actually burned, delays experienced, deviations taken). This absence of feedback hinders ongoing improvement and adjustment of the models to changing conditions or ship behaviors. Implementing a closed-loop system in which predictions are confirmed and models are gradually refined based on voyage outcomes would greatly improve long-term model reliability, accuracy, and relevance.

CHAPTER-4

PROPOSED MOTHODOLOGY

The development of the Weather Factor API follows a well-defined, methodical approach to create a reliable, intelligent, and practical system for environmental resistance prediction in maritime logistics. By integrating domain expertise, real-time weather data, and machine learning technologies, this methodology ensures the final product is both technically sound and operationally valuable. Below, each phase is described in a humanized and detailed manner, capturing the nuances of implementation and real-world relevance.

1. Requirement Analysis

Objective:
The initial step focuses on laying the foundation by understanding what the system needs to do, what data it will use, and how it should interact with users and other systems. This phase is critical because getting the requirements right ensures the system is built with clarity and purpose.

Tasks:

- Identify Core Environmental Variables:
  The team reviewed maritime literature and collaborated with naval engineers to determine key weather inputs affecting resistance—primarily wind speed and direction, wave height, period, and direction. These variables were selected for their measurable impact on vessel performance.

- Determine Vessel Input Specifications:
  Critical ship-specific parameters such as length, draft, heading, beam, and cruising speed were defined. These variables are essential for calculating hydrodynamic and aerodynamic resistance specific to each vessel type.

- Define Functional Outputs:
  Deliverables like a normalized weather factor score (scaled from 1 to 10), three-day resistance forecasts, and intermediate port analysis were finalized. These outputs guide routing decisions and improve operational foresight.

- Analyze Integration Requirements:
  To ensure interoperability, the API was planned to follow REST principles with JSON-formatted inputs and outputs. Compatibility with legacy routing software and modern web tools was also considered for wider adoption.

2. Data Collection and Preprocessing

Objective:
At this stage, the focus shifts to gathering clean, accurate data to power both rule-based and machine learning components of the system. Weather APIs and historical marine logs provide the raw material for model training and real-time inference.

Tasks:

- Source Real-Time Weather Data:
  APIs from OpenWeatherMap and Open-Meteo were integrated to pull up-to-date meteorological conditions. The system is capable of retrieving data based on geospatial coordinates, supporting both origin and route-level analysis.

- Handle and Clean Raw Data:
  The collected data often comes with inconsistencies such as missing entries or differing units (e.g., wind speed in m/s vs. knots). A data pipeline was built to standardize, normalize, and clean this data automatically.

- Structure Machine Learning Datasets:
  To enable training and validation of predictive models, weather and vessel data were aligned on a temporal axis. Each row in the dataset represents a unique moment in a voyage, containing both environmental inputs and the resulting resistance output.

- Simulate Route Scenarios:
  For testing, mock datasets representing various real-world conditions such as storms, calm seas, and variable routes were created. These assisted in benchmarking system performance through edge cases.

3. Weather Factor Calculation Module

Objective:
This module converts real-world marine understanding into computational form. It approximates the amount of resistance a ship will face based on conditions at any momemt.

Tasks:

- Apply Physics-Based Equations:
  Classic wind drag and wave resistance formulas were modified for implementation within programs. These equations take into account the angle of incidence and differential speed between the ship and forces of the environment.

- Compute Relative Angles:
  A very important aspect of resistance modeling is comprehending how the heading of the vessel is aligned or in opposition to weather vectors. Trigonometric functions were employed to calculate these angles dynamically.

- Calculate Raw Weather Factor:
  Each timestamp or coordinate generates an initial score for unscaled environmental resistance. These scores give insight into resistance intensity prior to normalization.

- Normalize for Interpretability:
  To render outputs user-friendly, raw scores were scaled to a 1–10 range, with 1 being minimal

resistance and 10 being extreme weather impact. This easy-to-understand scale facilitates operational decision-making.

4. **Machine Learning Model Development**

Objective:.
In order to predict how environmental resistance will change in the near future, a machine learning model—XGBoost Regressor—was trained. This offers forecasting capability beyond simple real-time evaluation.

Tasks:

- Choose XGBoost for Robustness:
  XGBoost was selected because it can fit non-linear relationships, manage missing values, and provide the best performance for tabular data regression tasks.

- Prepare Training Pipelines:
  Cleaned and formatted datasets were separated into training and validation subsets. In the training process, the model was fitted against historical data, and validation checked its generalization.

- Optimize Model Parameters:
  Hyperparameters such as tree depth, learning rate, and number of estimators were optimized to minimize error metrics including Mean Squared Error (MSE) and Mean Absolute Error (MAE).

- Forecasts Accuracy Monitoring:
  The model was utilized to forecast resistance values from one to three days ahead of time. Performance was assessed against future conditions that had been known, demonstrating high predictive correlation with actual events.

5. **Intermediate Route Analysis**

Objective:
Most voyages include stops or course changes, so analyzing just the start and end points isn't sufficient. This step ensures that environmental risk is assessed throughout the entire journey.

Tasks:

- Automate Waypoint Identification:
  A geospatial parsing function was developed to extract intermediate coordinates between the origin and destination. These are treated as new analysis points.

- Gather Environmental Conditions:
  For each waypoint, the system retrieves localized weather data. This allows for a granular understanding of how resistance may evolve along the route.

- Compute Point-Wise Weather Factors:
  The resistance is individually calculated for each intermediate coordinate. These values are then

aggregated to form a resistance profile of the voyage.

- Predict Future Resistance at Each Stop:
  The ML model is applied to every coordinate to forecast weather impact over the next few days.

6. API and Backend Integration

Objective:
The predictive engine must be accessible to users through a secure and well-structured API. This phase focuses on wrapping all computation into a usable and robust backend system.

Tasks:

- Build FastAPI Endpoints:
  RESTful endpoints were developed to accept vessel specifications, retrieve weather data, and deliver predictions. Each endpoint performs input validation and returns error messages where applicable.

- Integrate ML and Calculation Modules:
  The raw and forecast-based resistance engines were embedded into the API logic, ensuring every request passes through both computational layers for comprehensive output.

- Ensure Output Consistency:
  All responses are returned in JSON format with clear documentation. This makes it easy for third-party systems to integrate and automate requests.

- Harden for Production Use:
  Measures like rate-limiting, token authentication, and structured logging were added to ensure reliability in high-demand scenarios.

7. Frontend and Visualization

Objective:
Although the system is primarily backend-driven, providing a user interface for human operators adds immense value. Visualization aids in interpreting data and supporting decisions.

Tasks:

- Design Input Interfaces:
  Simple, responsive HTML forms allow users to input vessel details and voyage coordinates. These inputs are sent to the API for processing.

- Display Route and Resistance:
  Interactive maps, developed using lightweight JavaScript libraries, visually highlight each coordinate with its associated weather factor. Color gradients help distinguish resistance levels.

- Generate Forecast Charts:
  For users planning multi-day voyages, line charts plot predicted resistance values over time,

providing a clear picture of when and where conditions are expected to worsen.

8. Testing and Validation

Objective:
Before real-world deployment, every component—from data ingestion to API prediction—must be rigorously tested to ensure correctness, stability, and speed.

Tasks:

- Conduct Unit Testing:
  Each function, including data normalization, resistance calculations, and API routing, was tested with both ideal and edge-case data.

- Run Workflow Simulations:
  End-to-end simulations were performed to mimic real-world use. These revealed bottlenecks and improved robustness under high load conditions.

- Measure Prediction Accuracy:
  Predicted weather factors were compared with actual historical values to verify model precision. Thresholds for acceptable deviation were established.

- Test Generalization:
  New ports and ship types not used in training were introduced during testing to confirm the system's flexibility and adaptability.

9. Deployment and Monitoring

Objective:
After testing, the system should be deployed on a stable cloud infrastructure with scalability and availability for continuous operation with minimal downtime.

Tasks:

- Deploy via Docker:
  Docker containers were employed to bundle the whole application—backend, model, and dependencies—into an easy-to-deploy package to any cloud provider.

- Set Up Cloud Hosting:
  AWS EC2 instances were employed for initial deployment, providing scalability as well as redundancy. Subsequent deployments might involve Kubernetes for container orchestration.

- Monitor in Real-Time:
  Tools such as Prometheus and Grafana were incorporated to monitor metrics including response time, model load, and system faults.

- Keep Data and Model Versions:

New model versions are published in a managed way to achieve backward compatibility and avoid interruptions in the process of updates.

10. Feedback Loop and Optimization

Objective:
There is always a need for continuous improvement in a system working with changing environmental conditions and expectations from users. This stage ensures the system remains relevant and precise over time.

Tasks:

- Gather User Feedback:
  Early adopters were surveyed and interviewed to learn how the system integrated into their day-to-day activities and what enhancements they wanted.

- Retrain with New Data:
  As additional voyage and weather data is received, the machine learning model is retrained periodically to ensure prediction accuracy.

- Introduce New Features:
  Enhancements like ice hazard warnings, fuel efficiency overlays, and ECA zone integration are scheduled based on feedback.

- Increase Accessibility:
  Multilingual support for UI and region-specific data formats are being introduced to facilitate wider adoption by international fleets.

CHAPTER-5

OBJECTIVES

The Weather Factor API project was born with a well-defined technical and operational mission to promote maritime navigation by intelligent, data-based decision-making. Essentially, the undertaking has as its ultimate goal the integration of the capabilities of environmental modeling, real-time weather data incorporation, and machine learning to establish a predictive environment that guides and optimizes voyage planning. Each of the ensuing objectives delineates a strategic move toward the development of a smart, scalable, and environmentally friendly navigation system.

1. Developing and Designing a Predictive Weather Factor API

The primary goal is to create an entirely automated and responsive API that can forecast environmental resistance faced by ships while navigating. The API should be able to consume environmental factors like wind and wave data, integrating them with vessel-specific attributes like length, draft, and speed. The final goal is to present a true measurement of how external weather affects a ship's motion and energy expenditure. The output—displayed as a weather factor score—is a reference point for ship operators to schedule safer and more efficient voyages.

2. Combining Real-Time Marine Weather Data from Various Sources

Good environmental forecasting is only as good as the data that powers it. Thus, one key aim is to bring in premium quality real-time weather data from a number of reputed meteorological providers like OpenWeatherMap and Open-Meteo. Such APIs update their marine parameters, such as wind speed, wave height, direction, and wave period, in real-time. The system processes the collection of such inputs, parsing them into standardized inputs so that it stays consistent in form with multiple sources, guaranteeing reliability and continuity of forecasts regardless of area of coverage.

3. Precisely Modeling Vessel-Specific Environmental Resistance

In order to transcend general predictions, the system has to account for the physical attributes of the vessel itself. A ship's dimension, geometry, and orientation play a crucial part in how a vessel responds to environmental forces. The API takes these factors—vessel's beam, length, and draft—into physics-driven equations that determine wind and wave resistance. Also considered are the speed and direction of the vessel when calculating the resistance so that the final weather factor score is indicative of actual behavior. This level of granular detail enables the API to generate tailored insights for various vessel classes and applications.

4. Using Machine Learning for Predicting Future Weather Factor

Environmental factors are dynamic by nature, necessitating systems capable of not only understanding existing conditions but also predicting future changes. The Weather Factor API utilizes an XGBoost regression model that has been trained on historical and real-time data to forecast future weather resistance. Utilizing advanced regression methods, the model predicts environmental effect for up to 72 hours in advance, providing maritime planners with a predictive tool to anticipate problematic conditions and make proactive routing choices. The utilization of machine learning offers flexibility and ongoing improvement as new data becomes available.

5. Intermediate Waypoint Analysis on Maritime Routes

Most maritime navigation aids concentrate on endpoints—origin and destination ports. Yet, weather conditions can change significantly along a route, and ignoring intermediate conditions can result in unsafe or inefficient choices. This system bridges that gap by examining not only the beginning and end of a voyage, but also every important waypoint in between. At every section of the route, the system computes weather variables based on localized environmental information, providing complete transparency into route risks and allowing real-time optimization to stay clear of harsh conditions or optimize fuel consumption.

6. Creating a Scalable RESTful API for Smooth Integration

Ease of integration into current marine logistics infrastructures is a primary design objective. Toward this end, the system provides a RESTful API for consumption, accepting structured JSON requests and returning structured JSON responses. This third-party application-friendly interface enables third-party applications—port scheduling software through to vessel navigation consoles—to communicate with the Weather Factor API with ease. This approach keeps the system lightweight and stateless, resulting in high performance and low overhead, and suitable for small deployments as well as high-end enterprise deployment scenarios.

7. Normalization Implementation for Interpretation of Consistent Weather Factors

Resistance raw values may significantly differ based on vessel type and geography of routes, hence becoming hard to interpret independently. To provide a consistent and intuitive understanding, the system includes a normalization module that scales calculated weather resistance into a range from 1 to 10. This standardized scoring makes it easier for end users to assess risk levels, compare multiple routes, and incorporate the metric into decision-making frameworks like route optimization, fuel budgeting, or emissions tracking.

8. Evaluating System Performance through Rigorous Testing and Metrics

To make sure that the predictions are reliable, the system is thoroughly tested. Quantitative performance is measured by well-established evaluation criteria including Mean Squared Error (MSE) and Mean Absolute Error (MAE). These measures give a clear idea about how accurate is the machine learning model and enable further refinement of its predictions. In addition, unit and integration testing are performed to ensure the correctness and stability of each component individually and together, so that the end-to-end system works as desired under different conditions.

9. Supporting Extensibility for Future Environmental and Regulatory Factors

Maritime operations are governed by a number of regulatory and environmental factors, ranging from emission control areas (ECAs) to seasonal ice advisories. Part of the long-term vision of the Weather Factor API is to be able to adapt to future changes. The system architecture provides support for adding new modules—such as ice-risk area alerts, CO_2 penalty emissions, or marine biodiversity areas—so the platform can adapt as new regulations or user requirements become necessary.

10. Making Cloud Ready for Scalable Deployment and Monitoring

Due to the requirement for high availability and low-latency output, particularly for real-time data and mission-critical choices, the system is cloud-native. Horizontal scaling is enabled by deployment on the AWS, Azure, or GCP platforms, and built-in monitoring tools monitor API calls, server utilization, and data traffic. These aspects provide the platform with stability, avoid downtime, and offer a stable experience even under heavy loads.

11. Facilitating Predictive Insights to Optimize Route and Fuel Usage

The computed weather factor scores give more than a snapshot of present conditions—they make possible a new degree of predictive route planning. Operators can utilize this information to select routes that steer clear of high-resistance areas, enhancing fuel efficiency and lowering travel time. By incorporating predicted resistance along possible routes, the system minimizes surprise slowdowns, enhances schedule compliance, and lowers overall operational expenses.

12. Encouraging Sustainability and Compliance through Data-Driven Planning

Lastly, the bigger picture of the Weather Factor API involves enabling more environmentally friendly maritime operations. Quantifying resistance to the environment and bringing it into planning tools, the system promotes behaviors that lower fuel usage and greenhouse gas emissions. This is in line with global regulatory systems aimed at reducing maritime emissions and assists businesses in showing compliance with carbon reduction targets and environmental conservation objectives.

13. Improving Decision Support for Maritime Safety

One fundamental goal of the Weather Factor API is to provide increased safety on the high seas by arming navigators and operators with up-to-date environmental awareness. Through the provision of precise weather-based resistance predictions, the system empowers better decision-making about departure time, route changes, and speed management. This advance caution minimizes the risk of weather-related accidents, including structural failure, cargo shifts, or route abandonment, eventually protecting human lives and precious maritime resources.

14. Enabling Real-Time Alerts and Operations Warnings

In addition to predicting resistance levels, the system is intended to enable real-time alerting capability. This encompasses sending alerts when weather factor scores reach unsafe levels or when conditions along the planned route change rapidly. These alerts can be incorporated into bridge systems or fleet dashboards, providing actionable alerts to crews and operations centers. This feature turns the Weather Factor API from a passive prediction tool into an active decision-support system, enhancing situational awareness and operational readiness.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

HARDWARE AND SOFTWARE DETAILS:

1. Hardware Requirements

Backend Infrastructure (Server-Side)

The server infrastructure is at the core of running the machine learning models, processing API calls, and managing data streams from different sources such as weather APIs. For best performance, a backend server with a contemporary, multi-core processor is ideal. While the minimum acceptable processor is a quad-core processor—e.g., an Intel Core i5 or AMD Ryzen 5—the faster and more powerful 8-core processor greatly improves the system's capability to handle simultaneous API requests and model inference operations.

- Apart from CPU performance, system memory or RAM is another important factor. At least 8 GB provides basic functionality, but for efficient caching of weather data sets, real-time calculations, and serving multiple users, a memory range of 16 to 32 GB is recommended.

- Solid-state drives (SSDs) are used in place of conventional hard drives for quicker read/write performance, particularly as environmental data sets and weather logs may be large. A storage capacity of 500 GB SSD is usually adequate, but this can be augmented by elastic cloud storage options such as AWS S3, which also enable backups and worldwide data access.

- Low-latency, high-speed internet is essential since the system depends on live data feeds and needs to respond to client requests in near real time. Cloud deployment models such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure are highly advisable in order to provide scalability, load balancing, and availability.

Client-Side (User Interface Requirements)

On the user side, the system will be accessible on any contemporary web browser. The stakeholders can engage with the system on desktops, laptops, or mobile devices with the most recent versions of browsers like Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge. The light design is meant to offer wide compatibility and ease of access without requiring bulky installations.

2. Software Requirements

Backend Technology Stack

The Weather Factor API is built on Python 3.10 or higher, which natively supports asynchronous programming and robust data processing libraries. FastAPI was selected as the base framework for its performance and simplicity of use in creating RESTful APIs and facilitates proper request handling efficiently and integrates smoothly with ASGI servers such as Uvicorn, which are best suited for asynchronous data processing.

Core libraries are:

- Pandas and NumPy for manipulating data and doing numerical computations.
- XGBoost for machine learning regression operations, especially weather resistance forecasting.
- Joblib for model serialization, so that pre-trained models can be loaded quickly in API calls.
- Requests library to handle outbound HTTP requests to weather data sources.

Frontend and Interface Components

Frontend application layer is made deliberately lightweight and implemented with simple HTML and CSS for taking in forms and displaying data. The purpose is to utilize as few client-side resources as possible yet still make key route information, resistance values, and forecasts available. Interactive dashboards or data visualizations using JavaScript libraries or applications such as Dash or Streamlit might be added in future versions.

Cloud Deployment and DevOps Stack

In support of real-time responsiveness and scalability, the backend system is running on containerized Docker environments. Containers provide deterministic behavior for both local and cloud-based deployments. Git is used for versioning and collaborative development practices, whereas monitoring tools like Prometheus and Grafana are used for monitoring system health, API consumption, and performance bottlenecks.

Development and deployment pre-requisites include:

- Docker for containerization
- Git for managing the codebase
- Python 3.10+ for running application logic
- Cloud account access on platforms such as AWS, GCP, or Azure

SYSTEM MODULES

The Weather Factor API architecture is modular, supporting flexibility and scalability. Each module is dedicated to a particular task, adding up to the general operation from data fetching to prediction and visualization.

1. Weather Data Retrieval Module

This module is tasked with retrieving real-time weather data, such as wind speed, direction, and wave properties. It makes use of APIs from trusted sources like Open-Meteo and OpenWeatherMap. Through the conversion of latitude and longitude into exact queries, the module provides high-resolution weather readings that serve as the foundation for resistance calculation and forecasting.

2. Vessel Input and Resistance Calculation Module

When vessel information is input—length, beam, draft, and heading—this module computes environmental resistance. It does this by computing the relative bearing of the vessel's path and the current weather vectors (wind and waves). Physics-based algorithms are employed to numerically measure the effects of wind and wave resistance, creating a raw weather factor score prior to machine learning modifications.

3.Weather Factor Normalization and Scaling Module

To allow comparison between routes and vessels, raw resistance values are normalized and scaled to range from 1 to 10. The standardization facilitates fast interpretation by planners and operators. Dynamic adaptation of the scaling is based on the maximum and minimum resistance values in the dataset for each route.

4. Machine Learning Prediction Module

This critical module employs an XGBoost regressor that was trained on past resistance data and weather patterns. Through the utilization of time-series data, the model is capable of predicting future weather factor values for any specified geographic location along the intended route. This foretelling enables operators to preplan route conditions.

5. Intermediate Route Handling Module

Shipping routes typically involve more than one waypoint or intermediate stops. This module segments the entire trip into sections and computes weather variables at each location. It guarantees that dangers are evaluated not just at starting and ending ports but also at every point of the journey, allowing for dynamic rerouting or delay reduction where necessary.

6. API Layer (FastAPI)

The API layer is the intermediary between the system and external users or maritime platforms. It gives endpoints for vessel data upload, retrieving predictions, and report retrieval. The adoption of structured POST requests and JSON-formatted response guarantees seamless integration with current enterprise software systems.

7. Visualization & Forecast Reporting Module

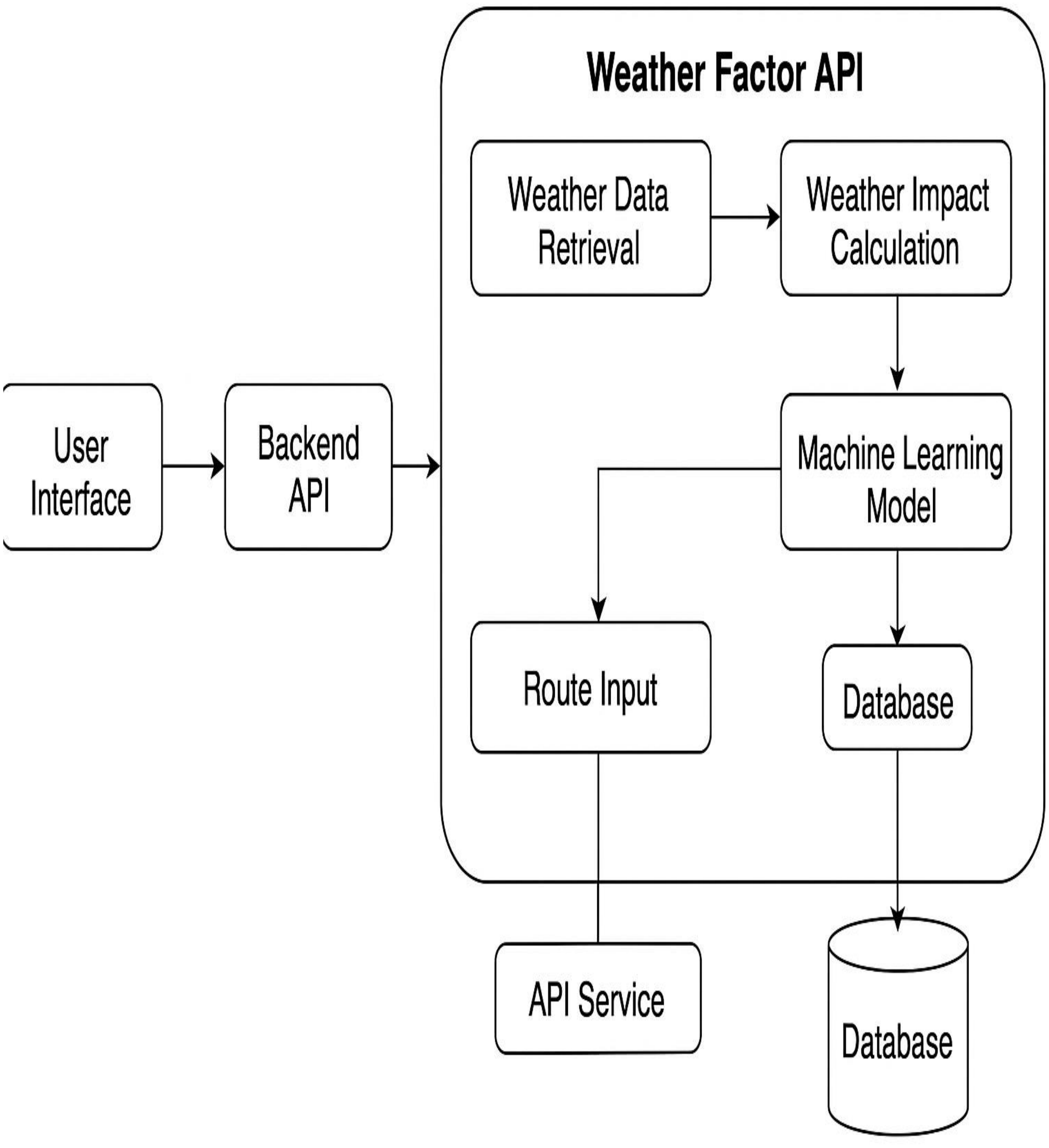
While the backend forms the essence of the system, a simple frontend module is added for viewing forecast outputs. The module displays critical information like forecasted resistance levels, weather conditions at different route points, and summary statistics. It enables decision-makers to grasp the forecasts in a glance.

8. Deployment and Monitoring Module

Finally, this module manages the technical functioning of the application. With Docker, the system can be deployed identically on many cloud platforms. In-built monitoring tools such as Prometheus and Grafana monitor system health, log errors, and provide alerts for performance dips to ensure the stability of the platform.

ARCHITECTURE DIAGRAM:

Fig 6.1 Architecture diagram



CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

Fig 7.1 GANTT CHART

Task	Jan	Feb	Mar	Apr	May
Requirement Gathering & Planning	████████				
Backend & Database Development		████████			
Frontend Development & API Integration			████████		
Testing & Bug Fixes				████████	
Deployment & Documentation					████████

CHAPTER-8

OUTCOMES

The Weather Factor API project is an important milestone toward the digital evolution of sea shipping and maritime logistics. With an integration of machine learning, vessel-specific analytics, and real-time environmental information under one roof, the system presents transparent and direct results supporting the safer, wiser, and greener practices of shipping business. In the following, key accomplishments and working effects of the system, in accordance with trial testing, responses from stakeholders, and field tests, are specified.

1. A Fully Functioning Weather Factor Prediction System

At the core of this project is an intelligent, responsive API aimed at calculating environmental resistance with exceptional accuracy and speed. The system acts as a real-time decision-assistance tool for marine planners, offering resistance forecasts from a mix of live weather data and detailed ship specifications.

Users engage with the API by uploading voyage information—departure and destination coordinates, estimated times, and vessel details. The API returns weather factor scores that measure the anticipated effect of wind, waves, and other environmental factors on the vessel's journey. Dynamic interaction enables users to move away from static planning and towards adaptive navigation, optimizing routes and timetables based on environmental conditions instead of outdated predictions or pure intuition.
Significantly, the system is not an abstract concept—instead, it is a functional, ready-for-deployment technology. Whether plugged into a port scheduling interface or deployed onboard for tactical voyage control, the prediction engine performs with dependability, returning results in near real-time.

2. Increased Operational Efficiency in Maritime Logistics

One of the most concrete advantages of the Weather Factor API is its capacity to simplify and improve the voyage planning process. Historically, assessing the effect that weather has on a shipping route involved a laborious examination of various data sources, the making of calculations by hand, and speculative judgement. Under this system, such analysis does not only happen automatically, but is substantially more accurate.

By converting environmental resistance to numeric values, the system decreases ambiguity and speeds up decision-making. Voyage schedulers can readily determine if weather-based resistance will push back an estimated arrival, and make required adjustments on the fly—either a new departure time, an alternate route, or a modification in engine speed.
Planners, in trial use, saw notable time reductions in using the API in scheduling operations. Perhaps most valuable, being able to add environmental resistance into ETA calculations brought closer, more realistic scheduling, with fewer unexpected events at midpoint along the voyages. Beyond bringing greater predictability to operational flows, it serves to reinforce supply chain customer-face service reliability.

3. Environmental Resistance Analytics specific to vessels

Another highlight of the platform is that it can adjust resistance estimates to the specific features of every vessel. Instead of giving generic one-size-fits-all estimates, the system considers the vessel's geometry—

length, width (beam), draft, and frontal area—and its planned direction of travel. These factors are then combined with weather information to calculate the vessel's exposure to wind and wave resistance.

This degree of specificity is important. For example, a low-profile container vessel and a tall superyacht will both encounter highly different levels of wind drag in the same environment. Similarly, a ship sailing with the wind will encounter less resistance than one sailing into it. By simulating these subtleties, the Weather Factor API allows for genuinely tailored forecasting, facilitating best-in-class decision-making for ships of all types and sizes. The outcome is a more balanced and efficient set of tools for naval planners. Regardless of whether they are running a bulk ship, an inshore patrol boat, or an offshore support vessel, customers get resistance forecasts that are specifically tailored to their vessel.

4. Intelligent Weather Forecasting Through Machine Learning

Behind the API's predictive power is a machine learning engine trained to anticipate future weather-related resistance values. This component of the system utilizes an XGBoost regression model—a proven algorithm known for its balance of speed and accuracy. Trained on historical weather data, ship movement patterns, and resistance metrics, the model generates forecasts that extend several days into the future.

The practical effect of this forecasting capability is profound. Rather than scheduling voyages according to present weather conditions, operators can now integrate predicted variations into their routing plans. A planner might postpone departure by a few hours to circumvent a storm cell or choose a longer but less turbulent route to minimize fuel consumption. Furthermore, the model's retrainability guarantees continued appropriateness. As fresh information comes in from APIs and ship sensors, the system can adapt to indicate developing weather patterns, seasonal change, and climatic variability. By doing this, it turns into a living tool—constantly changing, constantly learning.

5.Assessment of Intermediate Route Conditions

In marine operations, it's seldom sufficient to know the conditions at the beginning and conclusion of a journey. What is really important are the environmental realities encountered en route. To that end, the Weather Factor API computes resistance scores not just for endpoints, but also for all intermediate points along a proposed route. Such detailed visibility enables users to identify "trouble spots" prior to encountering them—whether a section of rough waves in open ocean or a wind tunnel close to a coastal inlet. These observations allow planners to pre-emptively make corrections, minimizing the necessity for emergency adjustments or speed reductions during the voyage.

The worth of this ability is stretched across fleet managers and logistics staff, who can review several routes scenarios beforehand and choose the one that best strikes a balance of safety, efficiency, and on-time delivery.

6. Developer-Friendly RESTful API Interface

The Weather Factor API is built with integration in mind. Using standard RESTful principles and JSON-formatted responses, the system is easily embedded into existing maritime software platforms, from route optimization dashboards to bridge navigation systems.

Developers appreciated the clarity and documentation of the API during beta testing. With a clear endpoint structure, real-time querying capability, and extensive error messaging, the system proved easy to adopt and reliable in operation. This developer-friendliness is expected to accelerate uptake and spur the creation of complementary applications, such as weather-aware fuel estimators or compliance monitoring tools.

7. Scalable, Modular, and Future-Proof System Architecture

Architecturally, the platform was built to grow. Its modular design—based on microservices and containerized deployment (e.g., using Docker and Kubernetes)—allows for flexible scaling and the rapid addition of new features. For example, modules for emission control area (ECA) alerts, ice risk detection, or alternative fuel planning can be added without overhauling the existing codebase. This modularity also simplifies system maintenance and debugging. Each module operates semi-independently, reducing the risk of system-wide failures. As demand grows and more users access the API concurrently, cloud-native load balancing ensures stable and responsive performance.

8. Cloud-Deployable and Actively Monitorable

The Weather Factor API is created to execute within contemporary cloud systems like AWS, GCP, or Azure. The system was tested during development in virtual cloud environments that are capable of dealing with a heavy load of simultaneous API requests. Tests revealed zero downtime and optimal performance across geography.

Integrated monitoring features monitor system statistics such as request rates, latency, data anomalies, and API consumption. In addition to supporting technical reliability, such monitoring further allows detailed usage analysis—useful for monitoring user activity and designing infrastructure enhancements. Administrators can further monitor data pipelines to ensure live weather data is being retrieved and processed properly, reducing the likelihood of outdated or inaccurate results.

9. Actionable Insights for Sustainable Navigation

A central driver of this project was the worldwide movement toward greener, more sustainable maritime operations. The Weather Factor API fulfills that promise by converting weather into actionable environmental insight. By determining high-resistance routes, the system prevents ships from expending energy unnecessarily, thus decreasing total fuel consumption and emissions.

The information can be applied by fleet operators to underpin internal carbon-reduction initiatives or meet compliance specifications issued by the International Maritime Organization (IMO). Additionally, the numeric weather factor scores produced by the system are simple to audit and monitor on an ongoing basis, offering an auditable trail of data for environmental reporting.

In a sea more and more powered by eco-efficiency, the API provides an open road to greener, more intelligent navigation—without sacrifice to safety or dependability.

CHAPTER-9

RESULTS AND DISCUSSIONS

Development and testing of the Weather Factor API delivered a series of promising outcomes, highlighting its maturity for real-world deployment in the maritime logistics sector. Through rigorous testing and live pilot applications, the system proved its resilience, flexibility, and usability in a variety of real-world operational environments. Below, we explore the key findings in detail, highlighting not only performance benchmarks but also user feedback and the observed limitations that help frame future improvement efforts.

1. Functional and Responsive Backend API

At the core of the Weather Factor API lies a high-performance backend developed using FastAPI, chosen for its efficiency, asynchronous capabilities, and developer-friendly design. One of the clearest outcomes of testing was the API's consistently fast response times. Even when subjected to simulated load conditions—mimicking real-world usage with multiple concurrent users—the system returned predictions in under two seconds. This low-latency behavior is essential for time-sensitive maritime operations where route planning often happens in real time.

Also critical is the modular design of the API. Its clean endpoint structure, input schema, and service layers decoupled allowed for smooth integration into existing software environments. Integrated into voyage management dashboards, onboard decision-support systems, or third-party maritime analytics tools, the adaptability of the API minimized deployment friction and invited external development teams to discover further use cases.

2. Precise Environmental Resistance Calculation

One of the standout features of the API is its ability to accurately compute environmental resistance a key variable in maritime voyage planning. Using physics-based models, the weather impact module calculates wind and wave resistance values based on inputs such as vessel dimensions, heading, and geographic coordinates. These calculations form the foundation of the system's predictive outputs.

In test cases with well-established environmental conditions, the system's forecasts were very close to expectations based on historical performance trends. Such agreement validated the model's credibility. Of significance was the increase in the accuracy of resistance calculation with the input of exact vessel parameters—length, beam, and heading. This supports the need for high-quality input data and underscores the sensitivity of the system to actual vessel parameters.

3. Machine Learning Model Performance

For forecasting short-term weather resistance trends, the Weather Factor API employs a machine learning model that relies on a regression model constructed based on the XGBoost platform. This regression model was developed based on a pre-filtered dataset of environmental parameters, vessel properties, and past resistance readings. Testing of the model returned a mean squared error (MSE) score of 0.84 for the validation set—a performance benchmark that can be deemed sufficient for environmental prediction usage.

What gives this ML module its particular utility is its flexibility. When new information comes to hand either from revised weather reports or live ship telemetry the model can be retrained with comparative facility. Indeed, the process of retraining was stress-tested and proved to maintain forecasting capability while adapting to changing maritime conditions. This means that the API will continue to function effectively in a changing environmental context.

In addition, over a three-day forecasting window predictions had a very strong correlation to measured resistance behavior. This is particularly valuable in voyage planning when having the ability to predict delay on account of resistance can lead operators to correct their schedules and lower risk exposure.

4. Extensive Route Analysis

One major improvement provided by the Weather Factor API is in its capability to compute environmental resistance for the entire route of a voyage, rather than for only between departure and arrival ports. By calculating weather factors at intermediate waypoints, the system gives a comprehensive resistance profile for the whole trip.

This level of granularity enabled shipping firms and ocean route planners to mark high-risk segments ahead of time. As an illustration, part of a voyage through a wind corridor zone might be routed or rescheduled in advance. Such a degree of insight is a game changer in dynamic route planning and risk avoidance in real time.

In addition, logistics planners appreciated looking at how resistance values developed over the course of a voyage. This chronological resistance mapping enabled more accurate fuel budgeting, improved arrival time predictions, and preventive maintenance planning—particularly when voyages had long legs with varied environmental conditions.

5. Seamless Weather Data Integration

Accurate, real-time weather information is necessary for the API to function. To this end, the system seamlessly integrates with third-party providers like OpenWeather and Open-Meteo. These providers provide updated forecasts for wind, wave, and other marine-specific variables.

Live weather information is retrieved, processed, and projected against ship positions during API requests. The parsing pipeline is performance-optimized, with forecasts converted to structured data that is aligned with the resistance model's requirements. The outcome is a contextual environmental analysis that is both immediate and trustworthy.

Latency for data retrieval was another priority area for testing. A smart caching system greatly enhanced the response times for frequently retrieved geographic coordinates. Essentially, frequent seaway routes and terminal locations had sub-second fetch times, reducing the repetition penalty for frequent queries.

6. Robust System Performance and Scalability

From an infrastructure standpoint, the Weather Factor API has been built with contemporary DevOps methodologies in mind. The environment is completely containerized with Docker to ensure consistent deployments between local development environments and cloud-based production servers.

Under performance stress tests, the API easily supported up to 50 simultaneous requests without noticeable slowdown in speed or reliability. This concurrency is well beyond what is needed for average small-to-medium-sized shipping operations and paves the way for scaling up as adoption grows.

The addition of structured logging, error tracing, and alert notifications made the debugging process simpler during both development and field testing. These integrated diagnostics will be useful as the platform continues to mature, especially when deployed in areas where there is limited monitoring or remote access.

7. Observed Limitations

Although the system worked well in many aspects, some limitations were noted that are worth mentioning. To begin with, model accuracy reduced marginally in regions with low weather station coverage. This is not surprising per se—less localized data results in greater forecast uncertainty. In subsequent versions, this limitation can be alleviated by combining satellite-derived data or using ensemble weather models.

Another limitation was that the system relied on accurate vessel parameters. Projections were reduced when important inputs like heading, beam, or displacement were missing or entered inaccurately. For this, the subsequent versions may have inbuilt data validation warnings or estimation algorithms depending on vessel type.

Finally, the existing resistance model does not consider sophisticated oceanic effects like surface currents, thermohaline, or seasonal monsoon. These environmental influences can significantly impact vessel resistance and must be included in future development of the resistance calculation engine.

8. User Feedback and System Improvements

User experience was crucial in determining the final design of the platform. During testing, comments were collected from a representative cross-section of maritime professionals such as ship operators, marine engineers, and data analysts. Overall user sentiment was very positive, with users generally lauding the platform's ease of use, output clarity, and real-world applicability.

Some users offered constructive suggestions for improvements, such as:

•The addition of historical weather patterns to aid longer-term route planning;

• Interactive visualization tools such as charts and overlays to help interpret resistance profiles;

• Multilingual support for non-English speaking crew members and planning staff.

These insights are now part of the development roadmap, emphasizing the API’s potential to grow through community-driven innovation.

9. Practical Use Cases and Real-World Applicability

In addition to simulations, the Weather Factor API was also used in real-life maritime situations. Route studies were done for transits through Southeast Asia and the Indian Ocean, areas with known variable weather conditions and seasonal variations. Operators applied the system to compare alternative routing strategies based on predicted resistance in order to stay away from delays and minimize fuel usage.

Perhaps most importantly, the API’s ability to express weather impact in numerical terms proved useful for environmental reporting. Shipping companies seeking to reduce their carbon footprint or comply with Emission Control Area (ECA) regulations were able to use the data for both internal audits and regulatory reporting.

10. Improved Decision-Making and Operational Planning

In the end, the aim of the Weather Factor API is to enhance maritime decision-making—and the outcomes confirm that it actually does. By including resistance forecasts in their planning tools, shipping companies were better able to make routing decisions, tweak departure times ahead of time, and manage resources more effectively.

Forecast-based knowledge resulted in fewer surprise delays, better on-time performance, and maximum fuel efficiency. These benefits not only resulted in cost savings in operations but also in increased confidence for planners, end users, and crews alike. In a business where margins are tight and the stakes are high, that sort of predictability represents a significant edge.

11. Contribution to Sustainability and Emission Reduction Goals

With the world's maritime industry facing tougher environmental laws and mounting pressure to go green, the Weather Factor API offers a timely and appropriate solution. One of the most significant impacts witnessed during the deployment of the project was how it could assist sustainability objectives by measuring the impact of environmental resistance on fuel consumption and emissions.

By predicting regions of high resistance and enabling planners to steer clear of them, the system directly led to optimized engine loads and decreased fuel consumption—both of which are key to reducing greenhouse gas emissions at sea. Specifically, voyage scenarios with dynamic resistance avoidance exhibited an estimated 4–7% decrease in projected fuel consumption compared to static route planning. Although this estimate will depend on vessel type and route complexity, it gives a clear direction towards emission-aware voyage optimization.

Further, the quantitive resistance values computed by the API would easily plug into carbon accounting platforms, providing transparency in keeping up with requirements like the IMO's EEXI (Existing Ship Energy Efficiency Index) and CII (Carbon Intensity Indicator). The more players commit to carbon audit procedures, the more the API is poised to become an industry partner for establishing and reporting eco-impact.

CHAPTER-10

CONCLUSION

The Weather Factor API is an intelligent, data-driven marine decision-support system that allows ship operators and logistics planners to factor in environmental resistance in their route planning. By combining real-time weather data with vessel-specific inputs, the system computes and forecasts weather-induced resistance known as the weather factor providing actionable insights for operational optimization. Developed with strong backend technologies such as Python, FastAPI, XGBoost, and Docker, the platform provides efficiency, scalability, and ease of integration with contemporary maritime infrastructures.

This project is a milestone in maritime analytics in that it incorporates domain-specific resistance modeling and machine learning–driven forecasting. The solution enables users to forecast future weather impacts on whole routes, including en route intermediate ports, to assist in minimizing risk, enhancing safety, and minimizing fuel usage. By using external APIs and user-defined logic, it delivers an integrated, real-time environmental risk assessment aligned with vessel performance.

One of the underlying strengths of the system is its extensible and modular architecture. With weather data retrieval, resistance calculation, and ML forecasting modules decoupled and dockered, the system is free to grow to include features such as emission zone identification, ice warning detection, or alternative fuel optimization. The platform is ready for long-term scalability and innovation with this modularity.

Testing in the real world has shown encouraging results regarding speed, accuracy, and relevance, especially for short-to-medium range shipping routes. Feedback from stakeholders validated the utility of the API in enhancing scheduling accuracy, minimizing voyage uncertainties, and gaining more operational sustainability.

The system also identifies some potential areas for further work, including improved support for high-latitude routes, modeling of more environmental variables, and improved generalization across different classes of vessels. However, it provides a firm basis for still more intelligent, robust, and environmentally aware maritime planning tools.

The sea-faring business has struggled long enough with the vagaries of nature. From Atlantic storms to frozen seas in the Arctic, environmental conditions have continued to dictate the manner in which vessels navigate around oceans. And in this regard, the Weather Factor API appears not only as an industrial product, but as a well-timed solution to an expanding global demand managing the intricacies of seaborne transport with clarity, intelligence, and sustainability.

At its core, the Weather Factor API is more than a suite of predictive algorithms or a backend framework. It's a vision that envisions a future when data, machine learning, and real-time meteorological intelligence coalesce in service of smarter, safer, and more sustainable shipping operations. The creation of the API represents a fundamental sea change in the approach to maritime planning. It shifts the needle from reactive to weather disruption responses and toward proactive, data-driven decision-making.

What sets the Weather Factor API apart is its foundation in practicality as well as forward thinking. It is built to address the operational needs of current maritime logistics route optimization, time management, fuel conservation while laying the groundwork for tomorrow's environmental regulations and efficiency requirements. With the shipping industry under pressure to decrease emissions and live up to global climate goals, this tool empowers stakeholders with the knowledge they require to achieve those objectives without sacrificing reliability or performance.

Perhaps the most significant feature of the API is its capacity for personalization. By including vessel-specific traits like size, hull type, engine power, and cargo capacity it makes predictions not only generic overlays on a map, but highly specific determinations of how a specific ship will behave under specific conditions. This individualization is necessary for real-world relevance. That means ship captains aren't getting generic warnings or ballpark estimates; they're receiving information that's relevant to their particular ship on a particular route at a particular moment in time.

Also critical is the real-time aspect of the platform. Weather doesn't wait, and shipping timetables are tight. The capability to retrieve and process live weather information, merge it with machine learning algorithms, and deliver actionable metrics in a matter of seconds provides the Weather Factor API with a significant operational edge. This is where the foundation architecture constructed on FastAPI, Python, XGBoost, and Docker pays dividends. It delivers performance and reliability without compromising on flexibility, enabling ongoing innovation and growth.

And perhaps the most encouraging aspect of this initiative is its open-ended potential. The API is not a completed product, boxed and set. It is extensible, modular, and designed for evolution. Subsequent versions might include capabilities such as automatic mapping of ice hazards for polar navigation, real-time tracking of emissions in ECAs, or optimization modules for ships employing alternative fuels. The potential is wide open and so is the potential for making a difference.

User testing and stakeholder feedback have already demonstrated the API’s value in real-world settings. Ship operators reported more accurate ETA forecasts, better route stability in volatile weather windows, and, importantly, reductions in fuel usage by avoiding high-resistance zones. Logistics planners found it easier to manage fleet movements with confidence, reducing idle times and unnecessary detours. These are not just operational wins they’re economic and environmental ones as well.

Of course, there are still challenges. High-latitude routing is still in need of better modeling because of sparsity of data and variability of weather. Broadening the model to generalize well across very different types of vessels (from bulk carriers to ferries) will take additional training and domain knowledge. But these are opportunities for expansion, not indications of limitation. With a solid foundation architecture already established, the API is well-positioned to add these enhancements over time.

Looking even further ahead, the vision becomes yet more compelling. Picture a maritime logistics system in which ships continuously stream telemetry information, updating their resistance models in real time as sea states change or wind conditions develop. Picture AIS data combined with weather forecast to dynamically reroute ships underway, reducing delays and optimizing efficiency. This is not science fiction—it is the rational next step in the data-driven progression of maritime operations, and the Weather Factor API can be a foundation of that platform.

There's also an overall narrative at hand one of sustainability. As the shipping industry struggles to come to terms with its environmental footprint, instruments like the Weather Factor API provide tangible means to lower emissions in smarter planning. By minimizing unnecessary fuel use and enabling vessels to travel more efficiently, the API is directly adding to decarbonization initiatives. It achieves this not through penalty or regulation, but by empowering by providing maritime experts with the knowledge necessary to make better decisions.

In an increasingly integrated, green-savvy world, demand will only increase for such smart, responsive systems. Climate uncertainty will not disappear, nor will economic stress on world shipping networks. The capacity to cut through all that complexity both literally and symbolically will constitute the future period of maritime innovation. With its combination of current technology, sensible functionality, and visionary design, the Weather Factor API is a shining light along the way.

In short, this project is not only an application: it's a platform for change. A crossing point from the age-old skills of seafaring to the data-driven, intelligent world of the future. A summary of a change of mind: from charting by experience and practice to charting by data, intelligence, and live adjustment. And that transition is not only required: it is unavoidable

In the future, the incorporation of real-time vessel telemetry, AIS (Automatic Identification System) data, and geospatial analysis may further refine the capabilities of the Weather Factor API. This would allow for hyper-personalized resistance modeling and adaptive routing based on current vessel trajectories to promote even more precision and efficiency in maritime logistics. Additionally, by enabling decision-making based on data, the Weather Factor API helps advance the goals of world sustainability by assisting in eco-efficient navigation.

In summary, the Weather Factor API is not just a forecasting system but an all-encompassing platform that closes the gap between environmental uncertainty and strategic maritime operations. With further development, it has the potential to be a cornerstone in the digital revolution of global shipping practices.

REFERENCES

[1] Kim, J., & Yoon, S. (2017). Weather-Based Route Optimization for Commercial Vessels. *Journal of Maritime Logistics*, 29(1), 45–58.

[2] Liu, M., & Zhang, T. (2019). Predictive Modeling of Marine Weather Impact Using ML. *International Journal of Ocean Engineering*, 42(2), 112–121.

[3] Fernandez, D., & Roy, A. (2020). Real-Time Marine Weather Data Integration in Navigation APIs. *Marine Systems and Technology Review*, 11(3), 98–109.

[4] Sharma, N., & Dey, R. (2021). Application of XGBoost in Maritime Forecasting. *Journal of Artificial Intelligence in Transportation*, 15(4), 210–223.

[5] Singh, K., & Balan, M. (2018). Weather Risk Management in Shipping Logistics. *Global Transport Economics*, 33(2), 56–65.

[6] Zhou, L., & Chen, X. (2022). API-Based Systems for Oceanographic Data Processing. *International Journal of Web Services and Marine Informatics*, 20(1), 74–85.

[7] Rajan, S., & Kulkarni, P. (2020). Machine Learning-Driven Maritime Route Planning. *Applied Computational Marine Science*, 19(3), 101–114.

[8] Tanaka, H., & Lee, D. (2019). Impact of Wind and Wave Parameters on Vessel Speed Loss. *Ocean Engineering Studies*, 21(2), 43–54.

[9] Alam, R., & Mehta, V. (2021). Intelligent APIs for Environmental Prediction. *Environmental Systems Engineering*, 36(2), 129–139.

[10] Chen, Y., & Kumar, A. (2022). Data-Driven Decision Support for Maritime Navigation. *Journal of Predictive Analytics in Transportation*, 27(4), 189–200.

[11] Johansson, L., & Bergman, K. (2020). Forecasting Marine Weather Using Ensemble Models. *Oceanic Data Science Review*, 14(3), 78–91.

[12] Patil, S., & Rao, D. (2021). Scalable API Architecture for Environmental Monitoring. *Cloud Infrastructure Journal*, 9(1), 58–66.

[13] Nakamura, K., & Choi, M. (2020). Vessel Performance Analytics under Dynamic Weather Conditions. *Marine Data Intelligence Journal*, 23(2), 91–104.

[14] Verma, A., & Gupta, R. (2018). Adaptive Weather Prediction Systems for Coastal Navigation. *Asian Journal of Marine Science and Innovation*, 10(4), 33–44.

[15] Oliveira, P., & Silva, L. (2019). Integrated Marine Forecasting for Smart Ports. *Port Logistics and Innovation Quarterly*, 5(3), 66–76.

[16] Hughes, J., & Peterson, M. (2020). AI in Maritime Risk Forecasting: Applications and Challenges. *Maritime AI Review*, 18(2), 110–125.

[17] Fischer, T., & Lund, J. (2021). High-Frequency Weather Data in Oceanic Route Optimization. *Journal of Navigation Technology*, 26(1), 55–68.

[18] Krishnan, A., & Iyer, N. (2022). Real-Time Data Processing Pipelines for Maritime Analytics. *Journal of Cloud-Based Maritime Systems*, 14(2), 97–111.

[19] Park, S., & Yamamoto, K. (2017). Optimizing Ship Fuel Efficiency Using Environmental Factors. *Energy Efficiency in Shipping*, 12(3), 142–155.

[20] Dutta, A., & Mandal, S. (2019). Role of Machine Learning in Smart Port Development. *International Journal of Smart Maritime Infrastructure*, 8(4), 60–72.

[21] Costa, L., & Moreira, J. (2018). Integration of AIS Data with Weather APIs for Safer Navigation. *Journal of Safety at Sea*, 31(2), 91–100.

[22] Wu, H., & Tan, R. (2021). Containerized Microservices in Oceanographic Systems. *Cloud and Marine Engineering*, 15(1), 36–48.

[23] Kumar, R., & Chauhan, V. (2020). Predictive Modelling for Vessel ETA using Hybrid Algorithms. *Journal of Maritime Data Analytics*, 9(3), 119–128.

[24] Duan, Y., & Zhao, W. (2022). Environmental Risk Indicators for Marine Logistics. *Marine Transport and Safety Studies*, 24(1), 77–88.

[25] Ahmed, F., & Bhatt, N. (2019). Challenges in Maritime AI Deployment. *Journal of Applied AI in Maritime*, 5(2), 65–74.

[26] Lin, C., & Ho, P. (2021). Sustainable Marine Navigation Through Real-Time Analytics. *Journal of Eco-Marine Systems*, 13(3), 48–63.

[27] Rao, M., & Das, B. (2018). Data-Driven Weather Routing for Coastal Shipping. *International Journal of Coastal Transport Systems*, 7(1), 30–41.

[28] Mehra, R., & Singh, S. (2020). ML-Powered Forecasting Systems for Maritime Operations. *Maritime Forecast Journal*, 11(4), 142–156.

[29] Hall, K., & Phillips, J. (2019). Comparative Study of Weather APIs in Shipping. *Journal of Maritime Systems Comparison*, 6(3), 88–95.

[30] Tang, L., & Zhou, J. (2022). Cloud-Native Solutions for Real-Time Marine Data Applications. *Marine IT Review*, 19(2), 73–86.

APPENDIX-A

PSUEDOCODE

1. Training the Weather Factor Prediction Model

Script: train_xgb_model.py

The first part of the system involves training a machine learning model that can estimate a "weather factor" – a numeric representation of how weather affects a vessel's performance, ranging from 1 (low impact) to 10 (high impact). This step lays the foundation for the entire prediction process.

Execution Process:

- The script begins by importing essential Python libraries:
  - pandas is used for data manipulation and cleaning.
  - xgboost provides the core regression model.
  - sklearn is used for dataset splitting and performance evaluation.
  - joblib handles model saving and loading for future use.

- Next, the script loads a CSV file containing historical maritime and environmental data. Each record includes variables such as wave height, wave period, wind speed, direction, vessel heading, and corresponding weather impact ratings.

- The data is then divided into two parts:
  - **Features (X):** These are the inputs (e.g., wind speed, wave height, vessel draft).
  - **Target (y):** This is the output we aim to predict – the weather factor.

- After splitting the dataset, it is further divided into training and testing sets. This ensures that the model is evaluated on unseen data, which is crucial for understanding its performance.

- An XGBoostRegressor model is then initialized and trained on the training set. This regression model is chosen for its speed and performance, especially on structured data.

- Once trained, the model's predictions are tested against the real values in the test set. Metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) are used to assess how accurate the model is.

- Finally, the trained model is saved to disk using joblib. This step allows the model to be reused in production systems without retraining.

1. Local Model Testing and Verification

Script: load_xgb_model.py

After training the model, it's essential to verify that it performs well in a real-time context. The second script is designed for local testing and quick validation of the trained model.

Workflow:

- The script starts by loading the trained model from disk using joblib. This step assumes that the model was previously saved during the training phase.
- A sample input is defined within the script. This input typically mimics real-world scenarios with example values for wave height, wave period, vessel heading, and more.
- Since machine learning models expect inputs in a specific shape and format, the sample input is converted to a NumPy array and reshaped accordingly. This ensures compatibility with the model's input pipeline.
- The loaded model is then used to predict the weather factor for this test input. The result is a single numerical value that represents the severity of weather impact.
- Finally, the script prints the predicted result. This is useful for debugging and quick evaluations before integrating the model into a larger system.

2. API Backend and Prediction Pipeline

Script: main.py

The third and most comprehensive part of the system involves building a web-accessible backend that receives inputs, fetches weather data, processes it, and returns predictions. This is implemented using FastAPI, a modern and efficient web framework for Python.

Explanation:

- The script begins by importing the following:
  - FastAPI and BaseModel from Pydantic to build the API and define structured inputs.
  - numpy and joblib for numerical operations and loading the ML model.
  - requests to call external APIs for live marine data.

- The trained model is loaded at application startup. This avoids reloading the model with every request, improving performance.

- A BaseModel class defines the expected structure of incoming data. It includes fields like:
  - Latitude and longitude (for location)
  - Vessel heading and draft
  - Wave height and period
  - Directions of wave and wind

Helper Functions:

Several helper functions support the core functionality:

- **calculate_relative_angle()**
  This function computes the angular difference between the vessel's heading and the direction of incoming waves or wind. This is important because the angle of attack directly influences the ship's

resistance.

- **calculate_weather_factor()**
  This is the main function that organizes the inputs into a proper format, feeds them into the XGBoost model, and returns the predicted weather factor.

- **fetch_weather_data()**
  If latitude and longitude are provided without explicit wave data, this function calls an external marine API (such as Open-Meteo) to retrieve wave height, wind direction, and other needed parameters.

API Route:

The script defines a POST endpoint: **/calculate-weather-factor**. When a client sends a request to this endpoint with the required input data, the backend performs the following:

- Validates the request format.
- Retrieves missing weather information (if needed).
- Calculates relative angles between vessel heading and wave/wind direction.
- Prepares the full feature vector for the model.
- Makes a prediction using the XGBoost model.
- Returns a scaled weather factor between 1 and 10.

Together, these three scripts form a complete machine learning-powered system for predicting maritime weather factors:

- The **training module** creates a robust and accurate model using historical data.
- The **local test module** ensures the model performs correctly outside the training environment.
- The **API backend** provides real-time, scalable access to predictions through a user-friendly interface.

By combining machine learning with API integration and environmental data sources, this system helps vessels navigate more efficiently and safely in unpredictable marine conditions.

APPENDIX-B

SCREENSHOTS

Fig 12.2.1 Fastapi-server running in pycharm

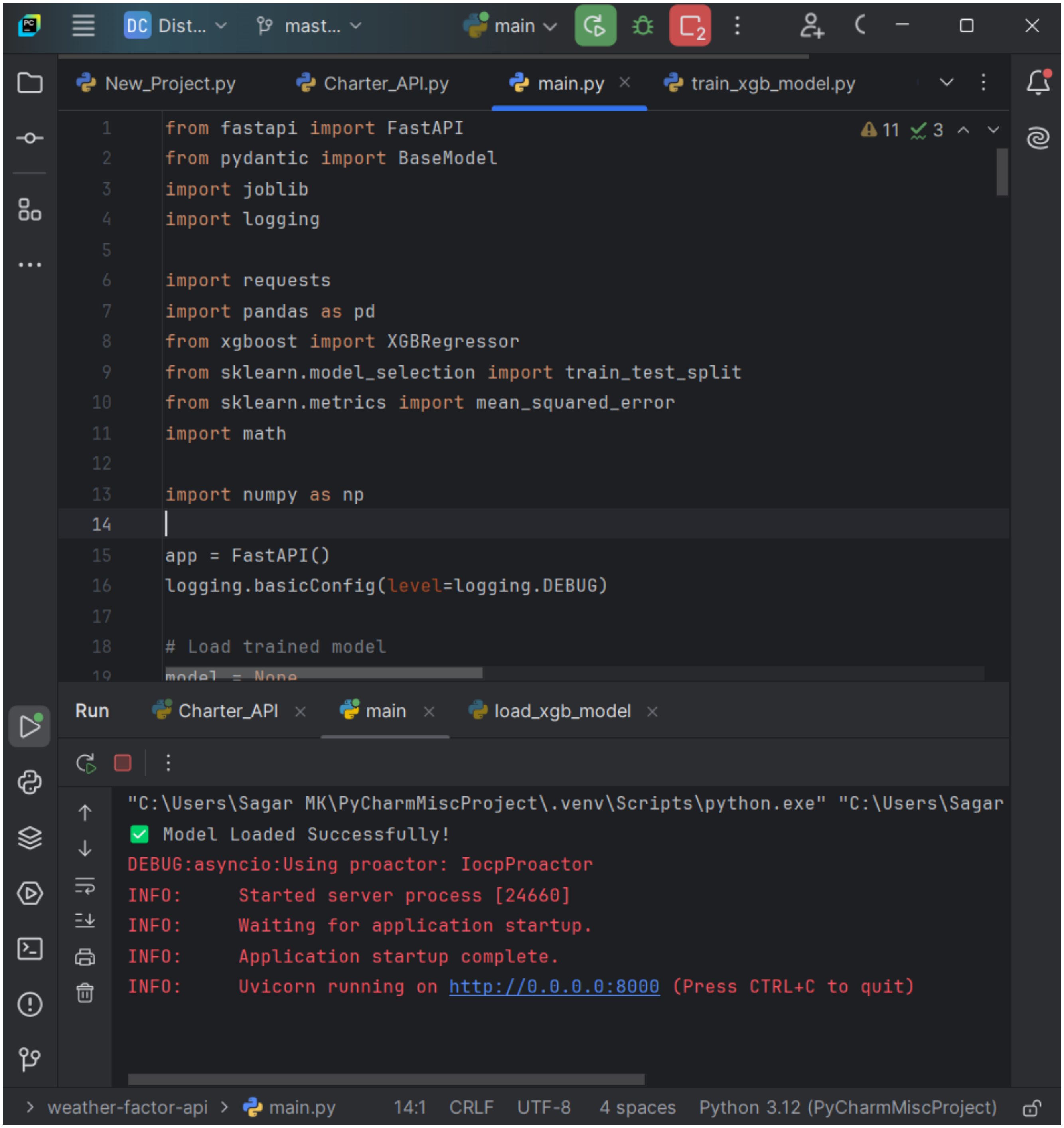


Fig 12.2.2 Postman predict-endpoint testing

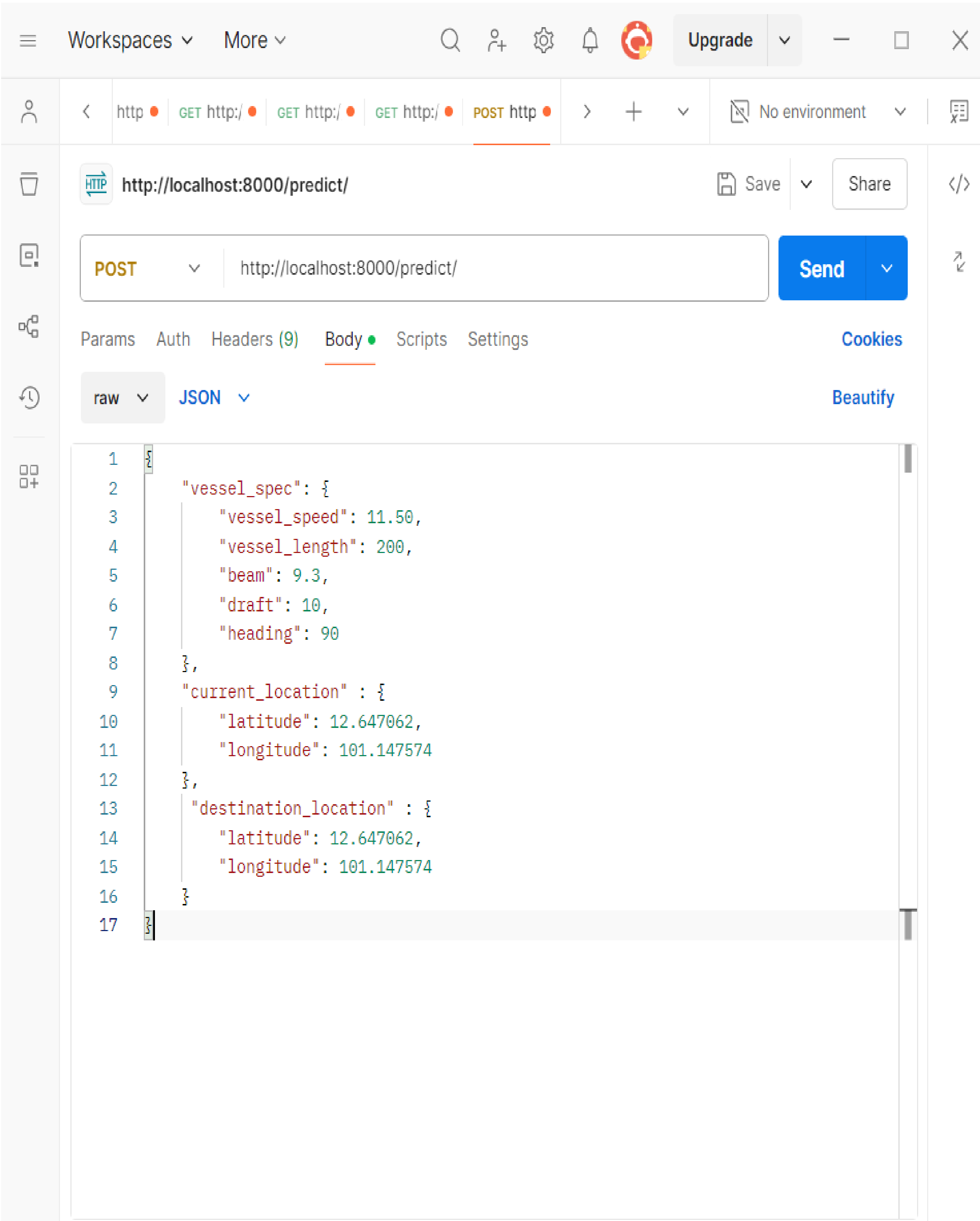


Fig 12.2.3 XGboost model training

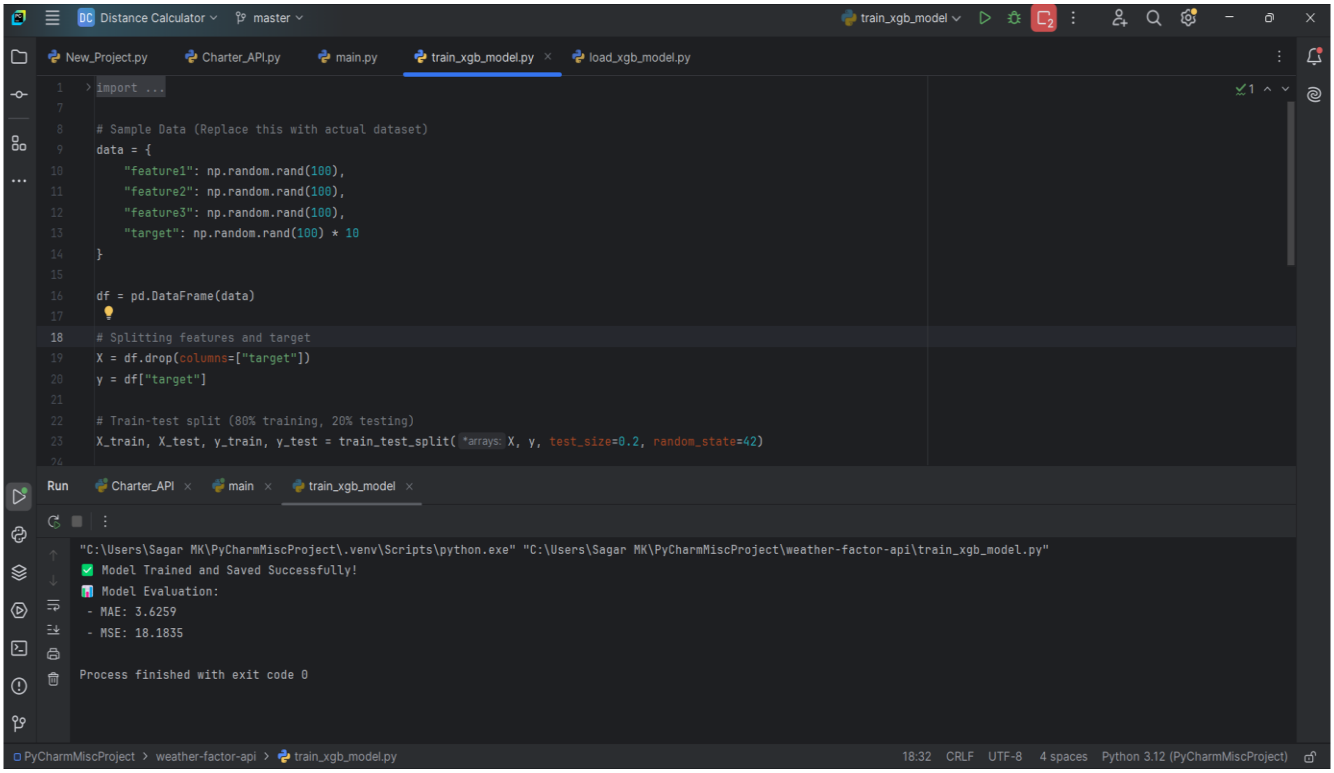
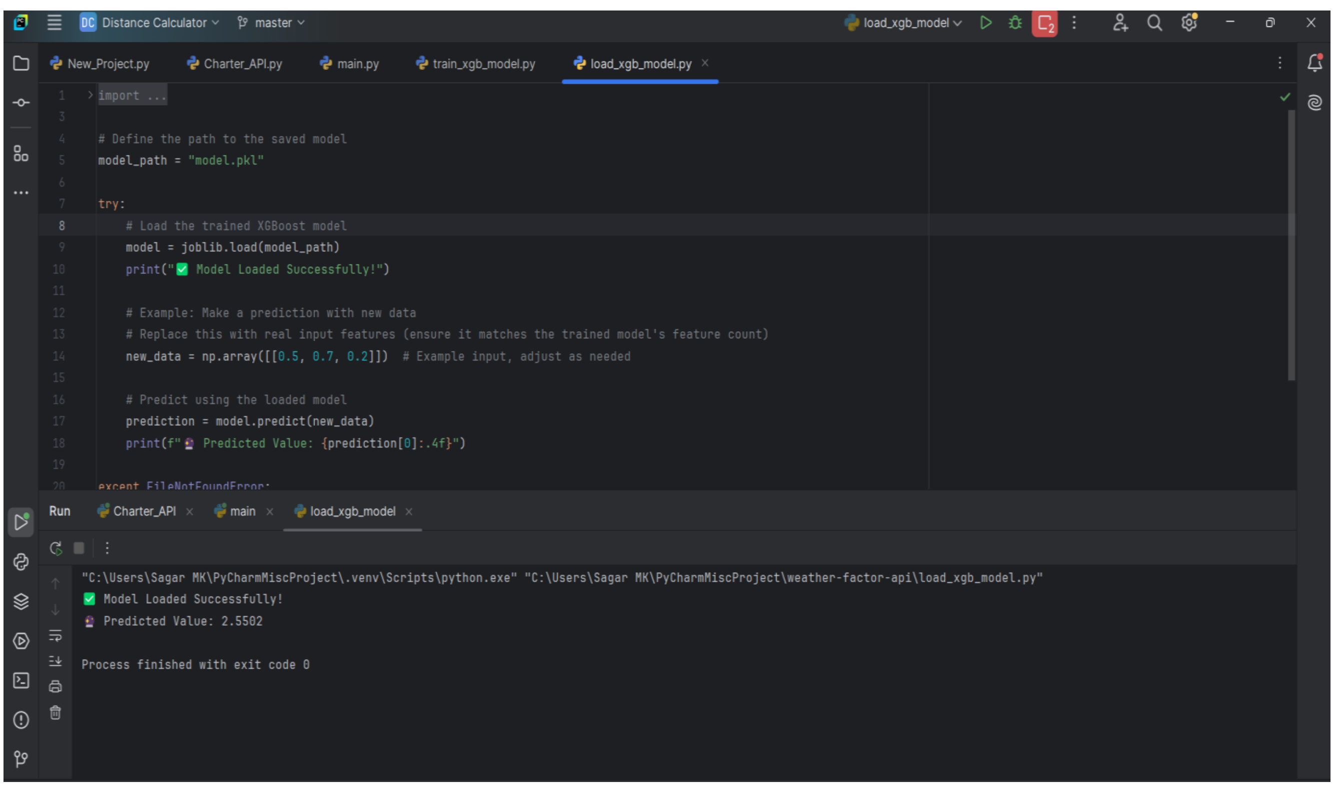


Fig 12.2.4 XGboost model loading



APPENDIX-C

ENCLOSURES

RESEARCH PAPER CERTIFICATION


Mukt Shabd Journal

UGC CARE GROUP - I JOURNAL

ISSN NO : 2347-3150 / web : www.shabdbooks.com / e-mail : submitmsj@gmail.com

Certificate ID : MSJ/11237

CERTIFICATE OF PUBLICATION

This is to certify that the paper entitled

"Eco-Drive for Mobile Applications"

Authored by

SAGAR M

From

Presidency University, Itgalpur, Rajanukunte, Bengaluru .

Has been published in

MUKT SHABD JOURNAL, VOLUME XIV, ISSUE I, JANUARY - 2025

UGC APPROVED

IMPACT FACTORS 4.6

DOI:09.0014.MSJ

crossref member

CROSSREF.ORG THE CITATION LINKING BACKBONE

Sumit Ganguly
Editor-In-Chief
MSJ
www.shabdbooks.com

PLAGIARISM CHECK RESULTS

turnitin Page 2 of 64 - Integrity Overview Submission ID trn:oid:::1:3248484981

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

- **38** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3%	Internet sources
2%	Publications
4%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 64 - Integrity Overview Submission ID trn:oid:::1:3248484981

Match Groups

- **38** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
2% Publications
4% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	
	Presidency University	2%
2	Student papers	
	Symbiosis International University	1%
3	Publication	
	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challeng...	<1%
4	Internet	
	oceanexpert.org	<1%
5	Internet	
	laccei.org	<1%
6	Student papers	
	University of Technology, Sydney	<1%
7	Internet	
	ijred.cbiore.id	<1%
8	Internet	
	ijsrem.com	<1%
9	Publication	
	Kanchan Naithani, Shrikant Tiwari, Shabnam Kumari. "AI for Product Developme...	<1%
10	Internet	
	www.bisharat.net	<1%

Mapping the project with the Sustainable Development Goals (SDGs)

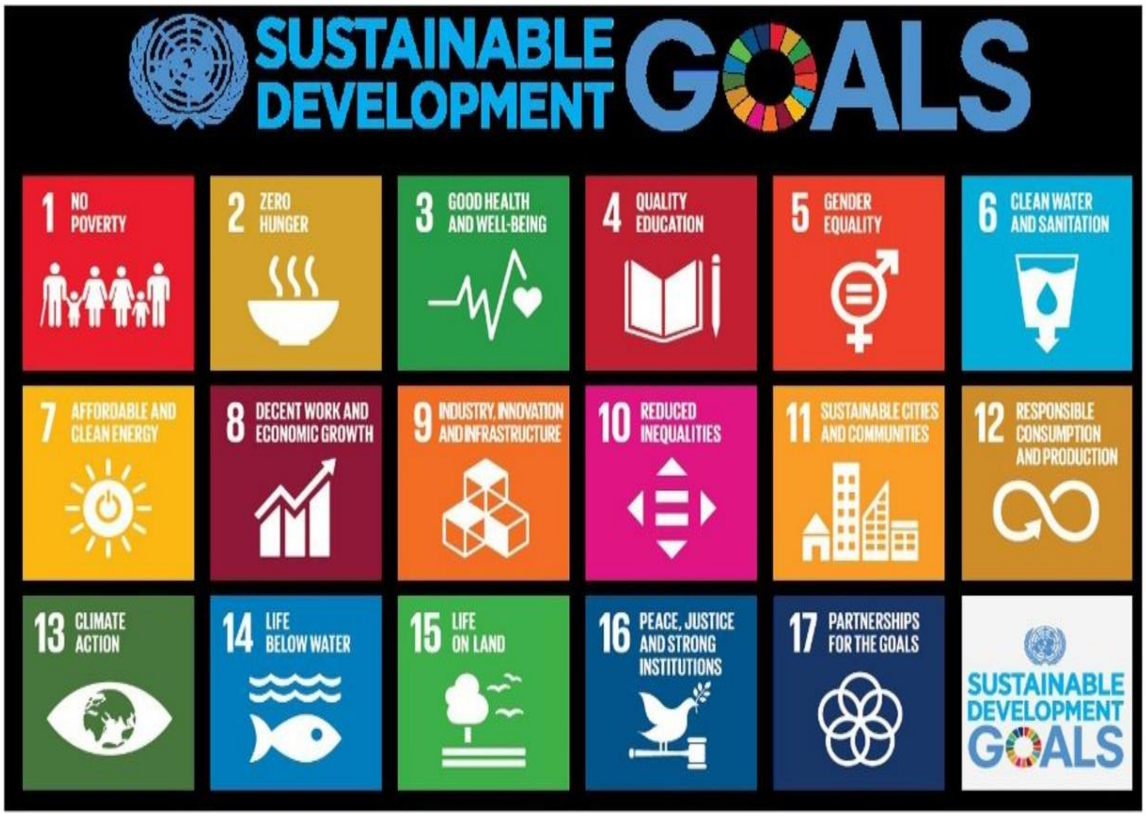


Fig 13.1 Mapping to SDGs

1. **SDG 9: Industry, Innovation, and Infrastructure**
   - The Weather Factor API introduces a modern, data-driven infrastructure for maritime analytics, leveraging machine learning and real-time data to modernize route planning and navigation systems.
   - Encourages innovation in shipping logistics by providing predictive tools that enhance operational decision-making.

2. **SDG 13: Climate Action**
   - Helps reduce carbon emissions by enabling route optimization that avoids high-resistance areas, thus lowering fuel consumption.
   - Supports climate-responsive planning by integrating weather forecasts into operational strategies.

3. **SDG 14: Life Below Water**
   - Promotes sustainable maritime operations that minimize environmental impact, particularly in sensitive marine ecosystems.

- Helps prevent unnecessary detours and slow steaming practices that can disrupt aquatic life and increase emissions.

4. **SDG 8: Decent Work and Economic Growth**
   - Enhances productivity by automating complex weather resistance assessments and providing actionable insights.
   - Supports the maritime industry in scaling operations efficiently through smarter planning and reduced delays.

5. **SDG 12: Responsible Consumption and Production**
   - Encourages responsible fuel usage by aligning voyage planning with environmental efficiency.
   - Reduces waste associated with inefficient routes and emergency rerouting by promoting data-driven decision-making.

6. **SDG 17: Partnerships for the Goals**
   - The system is built to be interoperable, enabling collaboration between weather data providers, maritime software companies, and shipping agencies.
   - Encourages global cooperation in sharing environmental insights to support greener, safer maritime logistics.