ProCogia - TMU

Muzz Abdur-Razak

**Executive Summary (Nov 2023): Beachwater Analysis R Shiny App**

**Overview**

The Beachwater Quality Dashboard is a pivotal tool designed to offer historical and current data, as well as predictive insights on beach water quality in the Niagara region. This project involved significant enhancements to the dashboard, focusing on improving data accuracy, user experience, and predictive capabilities. The upgraded dashboard now features comprehensive historical data visualization, advanced predictive models, and customizable user interfaces, making it an indispensable resource for monitoring and decision-making regarding water safety and quality.

**Project Background**

* Initial State and Challenges: The original Beachwater Quality Dashboard, while functional, faced challenges in terms of data integration, user interface design, and predictive accuracy. The data range was not being utilized to its full potential. Users found the dashboard less interactive and informative than desired, and the predictive models lacked precision.
* Client Needs: The client sought to revamp the dashboard to make it more user-friendly, data-rich, and accurate in its predictions. The goal was to transform the dashboard into a more dynamic and reliable tool for local authorities and researchers, studying water quality trends or making informed decisions.

**Codebase Restructuring**

* Original State: The initial codebase was cluttered and lacked scalability. It posed challenges in maintenance and feature expansion.
* Restructuring Process: The restructuring involved modularizing the code, optimizing data processing algorithms, and enhancing the overall architecture for better performance and scalability.
* Improvements: Post-restructuring, the codebase became more maintainable, allowing for easier updates and feature additions. Performance and scalability were significantly improved, facilitating smoother user experiences and faster data processing.

**Frontend Changes**

* Original Design Limitations: The initial frontend was less interactive and visually appealing, limiting user engagement.
* Changes Made: The frontend underwent a complete overhaul with a focus on UI/UX. New features for data interaction and visualization were introduced, making the dashboard more engaging and informative.
* Impact: These changes significantly enhanced the user experience, making data exploration and interpretation more intuitive and effective.

**Backend Changes**

* Initial Setup Shortcomings: The backend was initially less efficient in data handling and integration, affecting the overall performance of the dashboard.
* Modifications: Backend changes included optimizing data processing, integrating APIs for real-time data fetching, and improving server-side functionalities.
* Enhancements: These modifications led to a more robust and reliable backend, ensuring seamless data integration and faster response times.

**API Integration and Data Fetching**

* Significance: Integrating APIs for automatic data fetching was crucial for providing real-time, accurate environmental data, which is essential for predictive modeling.
* Process: The integration involved setting up API connections, handling data retrieval, processing, and error management efficiently.
* Contribution: This feature significantly contributed to the dashboard's effectiveness, offering up-to-date data, and enhancing the accuracy of predictive models.

**Testing and Validation**

* Methods: The application underwent rigorous testing, including unit tests, integration tests, and user acceptance testing to ensure functionality and reliability.
* Feedback: Feedback from stakeholders and early users was overwhelmingly positive, noting improvements in usability, data accuracy, and overall performance.

**Known Issues**

1. Data Discrepancies: The rain48 data for Niagara is currently hard-coded to reflect Toronto's data due to missing data source. This needs correction to ensure region-specific accuracy.
2. Station ID for Buoy Wave Height: A specific station ID is required for accurately fetching buoy wave height data. This is essential for precise local data representation.
3. Winter Period Data Handling: There is uncertainty about buoy operation during winter. It's crucial to determine if buoys are operational in winter and how to handle data during this period if they are not.
4. Meantemp24 Negative Value Handling: The meantemp24 metric does not account for negative values, whereas the lowest value in the network model is 7.3 °C. This limitation needs addressing for accurate temperature representation.
5. Avgwspd Upper Level Limitation: The upper level for avgwspd is too low, capped at 11, while actual values can be as high as 30. This discrepancy affects the accuracy of the model.
6. Limited Values Sent to cpquery(): Due to the above issues, currently, only maxUV and rain48 values are reliably sent to cpquery(), limiting the model's potential.
7. Handling of Null Values in Rainfall Data: Sending a null value for rainfall to the model consistently returns a 0% prediction, indicating a need for better handling of null or missing data.
8. Geomean24 Value Range Limitation: Sending values in the range of 200.01 - 2072.7 for geomean24 to the model always returns a 0% prediction, suggesting a need for expanding the model's range or reevaluating its parameters.

**Conclusion and Future Recommendations**

* Achievements: The project successfully transformed the Beachwater Quality Dashboard into a more dynamic, accurate, and user-friendly tool. It now stands as a model for future effective data visualization and predictive modeling in environmental monitoring.
* Future Development: Future recommendations include:
  + Adding other cities of interest and **continuously scale**
  + Developing **user database and user login** for expanded app flexibility and usage
  + Addressing and solving **known issues**:
    - Data Accuracy and Region-Specific Adjustments: Implement corrections to ensure that all environmental data, including rain48 and wave heights, are region-specific and accurate.
    - Enhanced Data Handling for Extreme and Missing Values: Develop strategies to handle extreme values and missing data, especially for metrics like avgwspd, meantemp24, and geomean24.
    - Model Expansion and Algorithm Enhancement: Expand the data range and incorporate additional environmental parameters. Explore advanced algorithms, possibly incorporating machine learning, to enhance predictive accuracy and handle a broader range of scenarios.
    - Winter Period Data Strategy: Establish a clear strategy for handling data during winter periods, especially concerning buoy operation and data availability.
    - Technical Documentation and Continuous Monitoring: Maintain comprehensive technical documentation and establish a system for continuous monitoring and updating of the dashboard to address any emerging issues or discrepancies.
  + Expanding the data range and replacing unreliable APIs using **Sarracenia protocol**. Sarracenia is Environment Canada’s publication/subscription management toolkit designed for the real-time publication of data. It adds a message queueing protocol layer to file and web servers, driving workflows that continuously transfer and transform data in real-time and mission-critical contexts. The toolkit aims to link processes together, avoiding the need to repeatedly poll servers or directories. It offers simple methods for parallelism in file transfers, robustness against failures, load balancing, and handling many failure modes, making it easier for application developers. The benefits of implementing the Sarracenia protocol are as follows:
    - Reduced Server Load and Improved Efficiency: Sarracenia avoids the need for clients to constantly poll the web server to check if new data is available. This can significantly reduce the workload for both the client and server, making the data retrieval process more efficient.
    - Real-Time Data Availability: It enables faster downloading by using true publish/subscribe mechanisms. Clients receive notifications precisely when new files are ready, ensuring that the app has the most current data available.
    - Scalability and Parallel Processing: Sarracenia naturally supports parallel processing. If one process is insufficient, more can be added seamlessly. This feature is particularly beneficial for handling large amounts of data or high traffic periods.
    - Flexibility and Robustness: The toolkit can push and pull data across different network configurations, including private networks and across firewalls. It also provides robust operation, handling server downtimes and transfer failures gracefully.
    - Customization and Workflow Integration: With its extensive plugin API, Sarracenia allows for on-the-fly file renaming, directory structure changes, and data-driven workflow implementation. This flexibility can be leveraged to tailor the data handling process to the specific needs of the beach water quality monitoring project.
  + **Further enhancement of UI** design, brand identity, accessibility, personalized experiences, and custom data stylization
    - Customized UI Components: Develop intuitive and visually appealing UI components tailored to the Beachwater Quality Dashboard. This includes custom buttons, input fields, and navigation elements that enhance user interaction and make the dashboard more engaging.
    - Responsive Design: Ensure the dashboard is fully responsive, providing an optimal viewing experience across various devices and screen sizes, from desktops to smartphones.
    - Theming and Branding: Implement a consistent theme and branding across the dashboard that aligns with the TMU’s visual identity. This includes color schemes, typography, and iconography that resonate with the client's brand.
    - Interactive Charts and Maps: Integrate advanced interactive charts and maps using libraries like plotly, ggplot, and leaflet. These visualizations will allow users to explore data in a more dynamic and informative way.
    - Custom Visualization Styles: Develop custom styles for data visualizations to maintain consistency with the dashboard's overall design. This includes tailored color palettes, chart layouts, and interactive elements like tooltips and legends.
    - Data Filtering and Exploration Tools: Provide tools for users to filter and explore data interactively, such as sliders, dropdown menus, and date range pickers. This will enable users to customize the data they view according to their specific interests or needs.
    - Accessibility Features: Implement accessibility features to ensure the dashboard is usable by people with disabilities. This includes keyboard navigation, screen reader support, and high-contrast modes.
    - User Guidance and Help Resources: Incorporate user guidance features, such as documentation, contact information, help icons, and an FAQ section, to assist users in navigating and utilizing the dashboard effectively.
    - Performance Optimization: Optimize the dashboard's performance to ensure fast loading times and smooth interactions, even with large datasets.
    - Personalized Dashboard Views: Allow users to customize their dashboard view by saving their preferred settings, filters, and visualizations.
    - Notification and Alerts System: Implement a system for sending notifications and alerts to users based on specific criteria or thresholds in the data, keeping them informed about important updates or changes.
    - Modular Design: Adopt a modular design approach, allowing for easy addition and modification of dashboard components as the project evolves.
    - Documentation and Style Guides: Create comprehensive Figma stylebook and style guides to facilitate future development and ensure consistency in design, functionality and brand identity.