# The Sustainable Living Advisor (SLA): A Technical Analysis of an AI-Powered Recommender System for Personal Carbon Footprint Management

## 1.0 Executive Summary

This report provides a comprehensive technical analysis of the "Sustainable Living Advisor" (SLA), a machine learning project designed to address a critical socio-technical challenge in climate mitigation.1 The project's core objective is to move beyond static data presentation and actively drive behavioral change by closing the well-documented "Awareness-Action Gap" in personal climate accountability.1

**The Problem:** While public awareness of climate change is high, this awareness rarely translates into effective, individual action. The project posits this is due to a "lack of personalized, clear guidance," a failure of existing static carbon calculators which are effective at quantification but poor at prescribing solutions.1

**The Solution:** The SLA is architected as an "ML-powered, dynamic recommender" system.1 It fundamentally reframes the problem by transforming a static calculator into an interactive system that provides "tailored, actionable sustainability insights" designed to empower users and drive "meaningful behavioral change".1

**Methodology:** The project utilized a Kaggle dataset titled "Individual Carbon Footprint Calculation," comprising 10,000 samples and 20 lifestyle predictors.1 A rigorous comparative analysis was performed on three regression algorithms—Support Vector Regression (SVR), Random Forest, and XGBoost—with the goal of predicting annual personal carbon emissions (measured in kg CO\_{2}eq).1 The primary selection metric for model performance was Mean Absolute Error (MAE), a choice that prioritizes average model accuracy for the median user.1

**Key Findings:** The XGBoost algorithm was decisively selected as the champion model, demonstrating superior performance with a coefficient of determination (R^{2}) of 0.974, a Root Mean Square Error (RMSE) of 164.47, and a winning MAE of 118.04.1 This performance significantly outperformed both Random Forest (MAE 249.47) and SVR (MAE 727.45).1 Model interpretability analysis identified "Vehicle Monthly Distance," "Diet Type," and "Frequency of Air Travel" as the top three drivers of emissions, a finding that strongly aligns with established climate science.1

**Deployment & Impact:** The system was deployed as a Flask web application, integrating the predictive model with real-time APIs (Air Quality, Energy Intensity) to provide dynamic, contextual advice.1 This dual-architecture (predictive model + real-time recommender) represents the project's key technical innovation, successfully bridging the gap from abstract awareness to concrete, data-driven action.1

**Future Outlook:** The project roadmap includes a mobile application, integration of further (transport, water) APIs, and—most critically—the implementation of SHAP-based explainability.1 This final point demonstrates a mature understanding of the system's current limitations (e.g., the use of demographic proxies) and a clear, technically-sound path to enhancement.

## 2.0 Introduction: The Socio-Technical Challenge of Personal Carbon Accountability

### 2.1 The Macro-Context: Policy and Individual Responsibility

The contemporary response to anthropogenic climate change is necessarily multifaceted, requiring both large-scale, top-down policy action and widespread, bottom-up individual accountability.1 While international agreements and national regulations establish the framework for decarbonization, these policies are often insufficient to drive the deep, granular behavioral shifts required to meet climate targets. Aggregate individual action—the sum of daily choices in consumption, diet, and mobility—represents a critical and often underestimated component of climate mitigation strategy. The project situates itself squarely at this intersection, focusing on empowering the individual as a key agent of change.

### 2.2 Deconstructing the "Awareness-Action Gap"

The central problem motivating this project is formally identified as the "Awareness-Action Gap".1 This term describes the well-documented psychological and practical disconnect between a person's general *awareness* of climate change and their *translation* of that awareness into tangible, effective action. The project hypothesizes that this gap is not a failure of individual will, but rather a failure of information design.

The root cause is identified as a "lack of personalized, clear guidance".1 Existing tools, such as conventional static carbon calculators, are effective at fulfilling an *informational* function: they quantify a user's impact, often in stark terms. However, they typically fail at the *prescriptive* function. By presenting a single, static number, they are proficient at inducing guilt or anxiety but are poorly equipped to provide a personalized, actionable, and prioritized roadmap for improvement. This information-only approach leaves users with awareness of the *problem* but no clear path to the *solution*, leading to inaction.

### 2.3 The Personal Carbon Footprint as a Key Metric

To bridge this gap, a tangible, measurable, and optimizable metric is required. The project selects the "personal carbon footprint" as the key metric for measuring daily environmental impact.1 This metric, quantified in kilograms of carbon dioxide equivalent (kg $CO\_{2}eq$), serves a dual purpose:

1. **As a Technical Target:** It is the *target variable* for the machine learning model, the value the system is trained to predict.1
2. **As a Behavioral Feedback Mechanism:** It is the *primary feedback mechanism* for the end-user. This metric successfully translates the abstract, global concept of "climate change" into a personal, tangible number that can be understood, benchmarked, and, most importantly, managed.

## 3.0 The Sustainable Living Advisor (SLA): Conceptual Framework and System Objectives

### 3.1 A Paradigm Shift: From Static Calculator to Dynamic Recommender

The Sustainable Living Advisor (SLA) is explicitly designed as the *solution* to the Awareness-Action Gap.1 Its core conceptual contribution, and the project's primary value proposition, is that it "Reframes static carbon calculators into a dynamic recommender system".1

This reframing represents a fundamental paradigm shift.

* A **static calculator** is a *one-way, terminal* function: A user inputs their data, and the system returns a single, static quantification of their footprint. The interaction ends there.
* A **dynamic recommender**, as architected in the SLA, is a *two-way, interactive loop*. A user's inputs generate not only a prediction but also a *prioritized set of personalized recommendations*. These recommendations are designed to "drive meaningful behavioral change" 1, which in turn would lead to new user inputs (e.g., "I have now switched to a vegan diet"). This creates a continuous feedback loop of action, measurement, and new recommendations, a concept further supported by the planned "progress tracking" enhancement.1

### 3.2 The Dual-Component Architecture: Prediction and Context

The project's objectives 1 and system workflow 1 imply a sophisticated dual-component architecture, which this analysis will deconstruct:

1. **The Predictive Engine:** This component consists of a machine learning model (XGBoost) trained on a *static, historical* dataset (the 10,000-sample Kaggle dataset).1 Its purpose is to fulfill Objectives 1 and 2: to *predict* a user's *annual baseline* carbon footprint based on their lifestyle inputs (diet, travel, etc.). This provides the user with a strategic, long-term overview of their impact.
2. **The Contextual Recommender:** This component consists of the Flask web application which integrates the model's prediction with *dynamic, live* data streams from external APIs, specifically the Air Quality and Energy Intensity APIs.1 This component fulfills Objective 3.

This dual architecture is the project's true technical innovation. It allows the SLA to provide two distinct, yet complementary, types of advice. The **Predictive Engine** offers *strategic* guidance (e.g., "Your annual footprint is high primarily due to your omnivorous diet and driving habits"). The **Contextual Recommender** offers *tactical*, real-time guidance (e.g., "Do not charge your electric vehicle *now*, as the grid's carbon intensity is high; wait three hours," or "Local air quality is poor today, making it a good day to avoid driving").

### 3.3 The Four Pillars: A Review of Project Objectives

The project's design is supported by four comprehensive objectives that collectively form a robust framework for developing a data-driven, user-centric product 1:

1. **Predict Emissions:** This is the core data science task—to "Develop an ML model to predict annual personal carbon emissions (kg $CO\_{2}eq$)".1
2. **Compare Algorithms:** This is the methodological rigor task—to "Evaluate SVR, Random Forest and XGBoost to choose the best regressor" 1, ensuring the selected model is demonstrably the most effective.
3. **Public Platform:** This is the software engineering and deployment task—to "Build a Flask web app for public access, integrating real-time APIs for contextual advice".1 This objective moves the project from a theoretical model (e.g., a Jupyter notebook) to a tangible, usable tool.
4. **Interpretability:** This is the human-computer interaction (HCI) and behavioral science task—to "Deliver interpretable, data-driven insights that encourage behavioural change".1 This final objective is the most critical; it is the explicit bridge across the Awareness-Action Gap. The project's ultimate success is not measured by its $R^{2}$ score, but by its ability to use its interpretable insights to change user behavior.

## 4.0 Data Science Methodology I: Data Acquisition, Curation, and Preprocessing

### 4.1 Dataset Specification

The foundation of the predictive engine is the "Individual Carbon Footprint Calculation" dataset sourced from the Kaggle platform.1 The dataset's parameters are as follows:

* **Size:** 10,000 samples
* **Features:** A comprehensive set of 20 lifestyle predictors
* **Target Variable:** Annual carbon emission (measured in kg $CO\_{2}eq$) 1

A sample size of 10,000 is generally considered robust for training tree-based models on a 20-predictor feature space and should be sufficient to develop a generalizable model while mitigating significant risks of overfitting, especially when using regularized algorithms like XGBoost.

### 4.2 Feature Engineering and Preprocessing Pipeline

The 20 predictors are described as a "comprehensive set" of lifestyle attributes, including features such as Dietary Habits (e.g., vegan, omnivore), Travel Patterns (air and ground), Household Energy Use, and Consumption Habits (e.g., shopping).1

The preprocessing pipeline was designed to be "simple, efficient, and interpretable".1 This involved a model-aware encoding strategy 1:

* **One-Hot Encoding:** Applied to standard, single-choice categorical features (e.g., a user has *one* primary Diet Type).
* **Multi-Hot Encoding:** Applied to features "allowing multiple selections".1

The choice of Multi-Hot Encoding is a non-trivial and important methodological decision. It suggests the dataset captures complex, overlapping behaviors (e.g., a user who regularly utilizes both a [private\_vehicle] and [public\_transit]). This richer feature representation allows the model to learn from these nuanced combinations, likely yielding a more accurate prediction than a simplified pipeline that forces single-choice categorization.

### 4.3 A Critical Methodological Choice: Scaling

A defining characteristic of the preprocessing pipeline was the decision to *minimize* feature scaling. The presentation notes that "Minimized the need for feature scaling by prioritizing tree-based models (Random Forest, XGBoost), which are less sensitive to the scale of input variables".1

This is a critical, pragmatic engineering decision that has profound consequences for the "Compare Algorithms" objective.1 This "simple, efficient... pipeline" 1 is optimized for the tree-based models (Random Forest, XGBoost) but *not* for the non-tree-based competitor, SVR. Support Vector Machines are explicitly noted as "scaling-sensitive".1

By design, the team created a biased experimental framework. They traded methodological purity (i.e., building a separate, optimized, scaled pipeline for SVR) for pipeline efficiency. This decision is entirely defensible for an *engineering* project (Objective 3) focused on deploying the *best* model, but it effectively guarantees the failure of SVR in this "bake-off." As will be seen in the following section, this decision fully explains the catastrophic performance of the SVR algorithm.

## 5.0 Data Science Methodology II: Comparative Model Evaluation and Selection

### 5.1 The Contenders: Algorithm Rationale

The project performed a comparative analysis of three distinct regression algorithms, each chosen for a specific reason 1:

* **SVR (Support Vector Regression):** Selected as a representative of high-complexity, kernel-based models. Its known "scaling-sensitive" nature 1 makes it a poor fit for the chosen pipeline, but a useful point of comparison.
* **Random Forest:** Selected as the "stable, interpretable baseline".1 As a robust, non-linear ensemble model that is *not* sensitive to feature scaling, it represents the standard, high-performance "control" group for this experiment.
* **XGBoost (Extreme Gradient Boosting):** Selected as the likely champion, described as "efficient, high accuracy, [requires] minimal preprocessing".1 Its inclusion is justified by its consistent state-of-the-art performance on tabular datasets, as cited in its foundational paper.1

### 5.2 The Selection Criterion: MAE over RMSE

The models were evaluated using a standard suite of regression metrics: MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and R^{2} (Coefficient of Determination).1

The "Algorithm Selection Logic" 1 explicitly states that the system selects the model with the "lowest MAE," which is noted as the "most important for regression" in this context. This choice is subtle but significant.

* **RMSE** squares the errors before averaging, thus "over-weighting" and heavily penalizing large, outlier errors. Optimizing for RMSE would create a model that is perfect for most users but potentially wildly inaccurate for a few.
* **MAE** treats all errors linearly, representing the average magnitude of error. By optimizing for MAE, the team prioritizes a model that is *on average* off by the smallest amount for *all* users. For a public-facing application (Objective 3), optimizing for the *median user experience* and ensuring the prediction is "close enough" for everyone is a more robust and defensible product decision.

### 5.3 Performance Analysis and Data Integrity

An analysis of the performance data on Page 7 of the presentation 1 reveals two conflicting tables.

* A text box at the top of the page lists SVR with R^{2}=0.895 and MAE=265.48.
* A formal table and a series of bar charts at the bottom of the page list SVR with R^{2}=0.0504 and MAE=727.45.

The bar charts for MAE, RMSE, and R^{2} all visually and numerically confirm the data in the *bottom* table. Furthermore, an R^{2} value of ~5% (0.0504) and a very high MAE (727.45) are far more plausible outcomes for a "scaling-sensitive" algorithm (SVR) trained on *unscaled* data, as discussed in Section 4.3. The R^{2}=0.895 value is almost certainly a typographical error. This report will, therefore, proceed using the data from the charts and the bottom table, which are consistent and methodologically sound.

### 5.4 The Results: A Decisive Victory for XGBoost

The comparative performance metrics clearly justify the selection of XGBoost as the champion model.

**Table 1: Comparative Performance of Regression Algorithms**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **R2 (Coefficient of Determination)** | **MAE (Mean Absolute Error, kg CO2​eq)** | **RMSE (Root Mean Square Error, kg CO2​eq)** |
| **SVR** | 0.0504 | 727.45 | 953.63 |
| **Random Forest** | 0.9670 | 249.47 | 327.67 |
| **XGBoost** | **0.9740** | **118.04** | **164.47** |
| Data sourced from 1 (Performance Metrics Summary table and charts). |  |  |  |

The analysis of this data is unequivocal:

* **SVR** failed completely, as predicted. Its $R^{2}$ of 0.0504 indicates its performance is no better than a simple mean-based guess.
* **Random Forest** provided a very strong, stable baseline with an $R^{2}$ of 0.967, proving the viability of the dataset and non-linear approach.
* **XGBoost** achieved the highest $R^{2}$ (0.974) and, most critically, decimated the error metrics. It cut the target metric, MAE, by more than 50% compared to Random Forest (118.04 vs 249.47) and cut the RMSE by nearly 50% (164.47 vs 327.67). This decisive victory demonstrates the clear superiority of the gradient boosting algorithm for this specific tabular dataset.

## 6.0 Champion Model Analysis: XGBoost Performance Diagnostics

### 6.1 Goodness-of-Fit: Actual vs. Predicted

The "Actual vs Predicted XGBoost" scatter plot 1 provides a visual confirmation of the model's high performance. The plot demonstrates a strong, tight linear relationship between the actual carbon emissions and the values predicted by the model. The data points cluster closely around the ideal 45-degree line, visually corroborating the high $R^{2}$ value of 0.974. There are no obvious, systematic deviations, suggesting the model is well-calibrated across the primary range of emission values.

### 6.2 Error Analysis: The Residual Plot

A deeper diagnostic is provided by the "Residual Plot - XGBoost".1 A residual plot charts the prediction errors (residuals) against the predicted values. In an ideal model, this plot should show a completely random, "star-like" scatter of points centered on the 0-error line (a state known as *homoscedasticity*), indicating the model has no systematic bias.

The XGBoost residual plot shows two distinct characteristics:

1. **For predictions between ~1000 and ~4000 kg:** The residuals are largely *homoscedastic*. They are randomly and evenly scattered around the 0 line, which is an excellent result. The "Distribution of Carbon Emissions" chart 1 confirms that this range contains the vast majority of the 10,000 data samples. This means the model is accurate and unbiased for the most common user profiles.
2. **For predictions greater than ~4000 kg:** The plot shows clear signs of *heteroscedasticity*—the residuals begin to "fan out," with errors becoming larger and more spread out.

The combination of the residual plot and the data distribution chart provides a clear explanation. The dataset has a *long tail* of high-emitters (4000-8000 kg), but relatively few samples in that tail. The model is *less certain* about its predictions for these high-emitters because it has seen fewer examples of them during training. This is a *key limitation* of the current system: the SLA's advice for a user with a predicted 6,000 kg footprint is *statistically less reliable* than its advice for a 2,000 kg user. This is a critical finding that must be transparently communicated to the user (e.g., via confidence intervals) and addressed in future model iterations, perhaps by oversampling high-emitters or collecting more data.

## 7.0 Model Interpretability: Deconstructing the Drivers of Personal Emissions

### 7.1 The Most Impactful Factors: Top 5 Drivers

Fulfilling Objective 4 (Interpretability) 1 is essential for bridging the Awareness-Action Gap. A prediction is useless if the user does not understand *why* it was generated. The model identified the following "Top 5 Predictive Features" as the most significant drivers of an individual's carbon footprint 1:

1. **Vehicle Monthly Distance:** "The most impactful factor is how much we drive."
2. **Diet Type:** "An omnivorous diet has a significantly higher carbon footprint compared to plant-based alternatives."
3. **Frequency of Air Travel:** "Long-haul flights and frequent flying contribute massively to emissions."
4. **Heating Energy Source:** The type of energy (e.g., natural gas, electric) used for home heating.
5. **New Clothes Purchased Monthly:** An indicator of consumption, "Fast fashion... have a notable environmental cost."

This output is highly valuable because it provides a clear, prioritized list of *actionable* behaviors. The project's conclusion that "Transportation and diet are the strongest predictors" 1 is not just a statistical finding but one that "strongly aligns with established real-world climate science".1 This alignment grounds the model's outputs in reality and builds user trust.

### 7.2 A Deeper Look: Top 10 Feature Importance

A more detailed, and more problematic, analysis comes from the full "Top 10 Feature Importance - XGBoost" bar chart.1 This chart provides a ranked list of the features that contribute most to the model's predictive power.

**Table 2: Top 10 Feature Importance Ranking (XGBoost)**

|  |  |  |
| --- | --- | --- |
| **Rank** | **Feature Name** | **Inferred Category** |
| 1 | Vehicle Monthly Distance Km | Transportation (Action) |
| 2 | Frequency of Traveling by Air | Transportation (Action) |
| 3 | Vehicle Type | Transportation (Asset) |
| 4 | Transport | Transportation (Action) |
| 5 | Sex | Demographics (Proxy) |
| 6 | Heating Energy Source | Energy (Asset) |
| 7 | Body Type | Demographics (Proxy) |
| 8 | Waste Bag Size | Consumption (Action) |
| 9 | How Many New Clothes Monthly | Consumption (Action) |
| 10 | Waste Bag Weekly Count | Consumption (Action) |

The top of the list confirms the dominance of transportation-related behaviors, which occupy the top four slots. This provides an unambiguous signal to the user about where the most significant gains can be made.

### 7.3 A Critical Flaw: Proxy Variables and Ethical Implications

The presence of "Sex" (Rank 5) and "Body Type" (Rank 7) in the Top 10 1 is a *significant and problematic* finding. This represents a critical flaw in the current model's interpretability.

These features are *not actionable behaviors*. A user cannot "change" their sex or body type to reduce their carbon footprint. These are *demographic proxy variables*, not *causal drivers*.

* **"Sex"** may be acting as a proxy for complex, unmeasured factors such as income, job type (e.g., a high-travel sales role), or even average vehicle choice (which is already a feature, suggesting a complex interaction).
* **"Body Type"** is almost certainly acting as a proxy for *consumption*—either dietary (e.g., higher caloric or protein intake, which correlates with meat consumption) or material consumption.

This reliance on proxies undermines the project's core objective 1 in two ways:

1. **It is not "interpretable"**: It does not provide a clear, causal reason for a prediction (e.g., "Your footprint is high because of your body type").
2. **It cannot "encourage behavioural change"**: It is disempowering and potentially unethical to base a recommendation on a non-actionable, personal attribute.

This flaw *directly* explains the necessity of the "SHAP-based explainability" enhancement listed in the project's future outlook.1 The project team is clearly aware that this simple feature-importance chart is insufficient. A SHAP (SHapley Additive exPlanations) analysis would be the correct solution, as it would move the system from *global feature importance* (what features matter *on average*) to *local feature contribution* (precisely *how* each feature contributed to *your specific* prediction). This would allow the model to untangle these proxies (e.g., "Your 'Body Type' feature contributes +50kg, because it is correlated with a high-meat diet") and, ideally, allow for a new model to be trained that replaces these proxies with their true, causal, and *actionable* counterparts (e.g., "Caloric Intake" or "Dietary Protein Source").

## 8.0 System Architecture and Deployment: The SLA in Practice

### 8.1 The Flask-Based System Workflow

The project successfully achieved Objective 3 (Public Platform) by deploying the trained model in a Flask web application.1 The system workflow is efficient and logical:

1. **User Input:** The user provides their lifestyle data (diet, travel, etc.) via a web interface.
2. **Flask Web Server:** The input is sent to the Flask server, which "has the loaded, trained... XGBoost model".1
3. **XGBoost Model:** The server passes the user's data to the model, which generates a prediction.
4. **Display Results:** The server returns the output to the user.

Crucially, the "Display Results" step is not just a single number. The workflow diagram 1 shows the output is a *bundle* of information: the "Predicted Value" + "Customized Recommendations."

### 8.2 The "Secret Sauce": Real-Time API Integration

The true innovation in the deployment architecture is the integration of this predictive bundle with "real-time API advice".1 This is the practical implementation of the "Contextual Recommender" component discussed in Section 3.2. The project integrates two specific APIs: the "Air Quality API" and the "Energy Intensity API".1

This integration transforms the SLA from a static tool into a dynamic, tactical advisor:

* **Energy Intensity API:** This API (such as the one from Electricity Maps, which is cited 1) provides real-time data on the carbon intensity of the local power grid. This allows the SLA to provide *time-specific, tactical* advice. For example: "Your baseline footprint is 2,200 kg. A simple way to reduce this is to run your dishwasher and charge your EV at 3 AM, when the grid's carbon intensity is 50 gCO2/kWh, rather than at 6 PM when it is 300 gCO2/kWh." This is profoundly more actionable than generic "use less energy" advice.
* **Air Quality API:** This API (such as AirNow, also cited 1) provides real-time data on local air pollutants. This allows the SLA to provide *health-based co-benefits* for sustainable choices, which is a powerful psychological motivator. For example: "Today's PM2.5 level is unhealthy. By taking public transit instead of driving (your #1 emission source), you will also reduce your personal exposure to harmful pollutants."

### 8.3 The User-Facing Synthesis: The Individual's Report

The sample "Individual's report" 1 serves as the final, user-facing synthesis of the entire system. It is the literal, tangible bridge across the Awareness-Action Gap, presenting the output of all components in a simple, interpretable dashboard.

1. **Component 1: Awareness (Context):** The "Your Carbon Emission vs Average" chart.1 This component anchors the user, contextualizing their personal score (e.g., 1406 kg) against a relevant benchmark (Average: 2209 kg). This immediately answers the user's first question: "Is my score good or bad?"
2. **Component 2: Interpretability (Attribution):** The "Estimated Carbon Footprint Breakdown" pie chart.1 This chart (e.g., Transport 40.0%, Diet 20.0%, Energy 15.0%, Lifestyle 25.0%) is the direct output of the model's interpretability. It tells the user *why* their score is what it is, directly reflecting the feature importances discussed in Section 7.0.
3. **Component 3: Action (Personalized Guidance):** The "Top 5 Recommendation Impacts" and "Emission Reduction Potential" charts.1 This is the *solution*. It shows the user *precisely* what to do (e.g., the top recommendation could save "400 kg") and quantifies the total potential impact of their behavioral changes (e.g., reducing from 1400 kg to 100 kg). This is the "personalized, clear guidance" 1 that the project set out to deliver from the very beginning.

## 9.0 Conclusion: Achievements and Strategic Future Trajectory

### 9.1 Summary of Achievements

The Sustainable Living Advisor project successfully met its primary objectives, delivering a technically robust and conceptually innovative solution to a significant socio-technical problem.

* **Technical Achievement:** A high-accuracy (R² 0.974, MAE 118.04) XGBoost model was successfully built, validated, and selected through a rigorous comparative analysis.1
* **Engineering Achievement:** The model was successfully deployed in an "interactive Flask app" 1, moving the project from a theoretical exercise to a usable public platform.
* **Conceptual Achievement:** The integration of "live environmental APIs" 1 created an innovative "dual-architecture" system. This system provides not only a *strategic* annual baseline prediction but also *tactical*, real-time, contextual advice, which is a significant advancement over static calculators.

The final impact is a system that effectively "bridges the gap between awareness and action" by offering "data-driven, personalised sustainability advice," thereby fulfilling the project's central thesis.1

### 9.2 Critical Analysis of the Future Enhancements Roadmap

The "Future Enhancements" 1 outlined by the project demonstrate a high level of project maturity. They are not simply a "wish list" but a strategic roadmap that directly addresses the key limitations of the current prototype as identified in this analysis.

* **"SHAP-based explainability"**: This is the *most critical* technical enhancement. As discussed in Section 7.3, it is the direct and correct solution to the "proxy variable" problem. SHAP analysis will allow the system to move from *global* importance to *local* contribution, enabling ethical, accurate, and truly causal explanations for users.
* **"Mobile app version"**: This is a *behavioral* enhancement. Climate action is a high-frequency, daily activity. A web app is a low-frequency, "one-time-check" tool. A mobile app is essential for long-term engagement, enabling *push notifications* (e.g., "The grid is clean *now*! Time to run the laundry") and integrating with the "progress tracking" feature.
* **"Progress tracking"**: This is the missing piece of the *behavioral change loop*. The current system provides a *plan* (the individual report), but "progress tracking" provides the *feedback* and *reinforcement* necessary to "encourage behavioural change" 1 over the long term.
* **"Additional APIs (transport, water)"**: This enhancement scales the *contextual recommender* component (Section 8.2). Integrating live public transport APIs or local water-stress APIs would make the system's tactical advice even more powerful and relevant.

### 9.3 Final Concluding Remarks

The Sustainable Living Advisor project is a highly successful and technically proficient prototype. It demonstrates a clear, intelligent, and end-to-end progression from a well-defined socio-technical problem to a data-driven, deployed solution. The project's "dual-architecture" (predictive baseline + contextual recommender) is a significant conceptual innovation. While current limitations exist—chiefly the model's reliance on demographic proxies (Section 7.3) and its reduced accuracy for high-emitters (Section 6.2)—the proposed future roadmap addresses these issues directly and intelligently. The project stands as an exemplary case study in using machine learning not just for *prediction*, but for *understanding* and *driving meaningful behavioral change*.

## 10.0 References

* T. Chen & C. Guestrin, "XGBoost: A Scalable Tree Boosting System," KDD 2016. 1
* L. Grinsztajn et al., "Tree-Based Models vs Deep Learning on Tabular Data," NeurIPS 2022. 1
* Kalra et al., "Carbon Emission Prediction in India using ML," Global NEST Journal, 2025. 1
* AirNow API; Electricity Maps API Documentation, 2024. 1

ARPITA PRIYADARSHINI SAHOO

REGD.NO-2302060030

EEE1