

**ENHANCING MODEL EVALUATION AND MODEL SELECTION DECISIONS USING PROPER SCORING RULES**

**Progress Report**

by Juan Jaya Jauanda

School of Information Technology and Electrical Engineering,

the University of Queensland.

Submitted for Master of Data Science Capstone 2

8th April 2025

**SUMMARY**

Write the summary of all the report

**Table of Contents**

[1. Introduction 5](#_Toc195108016)

[1.1 Description of the problem 6](#_Toc195108017)

[1.1.1 Diverse data formats and inconsistent tools 6](#_Toc195108018)

[1.1.2 Lack of standardized evaluation methods 6](#_Toc195108019)

[1.2 Project aims 7](#_Toc195108020)

[1.2.1 Prediction Standardization 8](#_Toc195108021)

[1.2.2 Scoring and Evaluation Framework 8](#_Toc195108022)

[1.2.3 Visualization and Interpretation 8](#_Toc195108023)

[1.2.4 Performance Optimization and Scalability 9](#_Toc195108024)

[1.2.5 Modularity and Extensibility 9](#_Toc195108025)

[1.3 Structure of progress report 9](#_Toc195108026)

[2. Background 11](#_Toc195108027)

[2.1 The importance of prediction standardization in modelling 11](#_Toc195108028)

[2.2 Understanding uncertainty in model predictions 12](#_Toc195108029)

[2.3 Review of related packages 14](#_Toc195108030)

[2.3.1 marginaleffects 15](#_Toc195108031)

[2.3.2 scoringRules 15](#_Toc195108032)

[2.3.3 stats 15](#_Toc195108033)

[2.3.4 ggplot2 16](#_Toc195108034)

[2.4 Identified gaps 17](#_Toc195108035)

[3. Architectural Design and Core Modules 18](#_Toc195108036)

[3.1 High-Level Overview of the Architecture 18](#_Toc195108037)

[3.1.1 scoriverse\_main.R 18](#_Toc195108038)

[3.1.2 prediction\_extraction.R 18](#_Toc195108039)

[3.1.3 scoring\_functions.R 18](#_Toc195108040)

[3.1.4 visualizations.R 19](#_Toc195108041)

[3.1.5 scoriverse-package.R 19](#_Toc195108042)

[3.1.6 wrappers.R 19](#_Toc195108043)

[3.2 Core Modules of the Package 19](#_Toc195108044)

[3.2.1 scoriverse\_main.R 19](#_Toc195108045)

[3.2.2 prediction\_extraction.R 20](#_Toc195108046)

[3.2.3 scoring\_functions.R 20](#_Toc195108047)

[3.2.4 visualization.R 21](#_Toc195108048)

[3.2.5 scoriverse-package.R 21](#_Toc195108049)

[3.2.6 wrappers.R 22](#_Toc195108050)

[3.3 Design Decisions for Extensibility and Consistency 23](#_Toc195108051)

[Reference 24](#_Toc195108052)

# **1. Introduction**

This project aims to develop an R package named Scoriverse, which provides a unified and automated framework for generating probabilistic predictions and evaluating predictive models using proper scoring rules. In the rapidly evolving fields of data science and machine learning, model evaluation plays a critical role in understanding and improving predictive models (Hastie et al., 2009). Traditional evaluation methods, such as accuracy and RMSE, focus on point estimates, providing a snapshot of how well a model is performing. While useful, these methods often fail to account for the inherent uncertainty in real-world data and predictions. This limitation is particularly apparent in applications like forecasting, risk analysis, and decision-making, where uncertainty is a significant factor (Kuhn et al., 2016). Evaluating a model’s performance is not only about assessing how well it predicts but also understanding the reliability of those predictions. Proper evaluation methods must consider the probabilistic nature of predictions and provide insights into their uncertainty (Gneiting & Katzfuss, 2014).

However, current approaches for handling uncertainty—such as manually implementing proper scoring rules—can be complex, error-prone, and inconsistent across different model types. This presents a significant challenge for data scientists and decision-makers who need robust, reliable evaluation metrics. Scoriverse aims to simplify and standardize this process. The package abstracts away the complexities of probabilistic evaluation, offering a model-agnostic, plug-and-play solution. By integrating uncertainty into model evaluation, Scoriverse enables users to compare models more effectively and make better-informed decisions. Designed to be flexible and accessible, the package supports various model types, from traditional statistical models to machine learning algorithms, while ensuring a consistent, transparent evaluation process.

This chapter will be divided into 3 sections. Section 1.1 discusses the challenges faced by traditional evaluation methods, section 1.2 outlines the specific objectives of the Scoriverse project, while section 1.3 provides an overview of the structure of this progress report, guiding the reader through the content that follows.

## **1.1 Description of the problem**

Despite the growing sophistication of predictive modelling, several persistent challenges complicate model evaluation for data scientists, analysts, and decision-makers. These challenges hinder reliable, consistent, and insightful model comparisons—particularly in contexts where uncertainty is a key factor. This section outlines four major problems that motivate the development of Scoriverse.

### **1.1.1 Diverse data formats and inconsistent tools**

A major hurdle in model evaluation stems from the diversity of modeling tools, data structures, and output formats. Different frameworks—whether statistical or machine learning—often have their own conventions for data handling, prediction output (e.g., point estimates vs. probability distributions), and evaluation workflows. As a result, users are frequently forced to manually adapt their data and predictions to fit the unique requirements of each tool (Gundersen & Kjensmo, 2018). This process is time-consuming, error-prone, and inhibits scalability. The absence of a unified interface for evaluation across tools and platforms leads to inefficiencies and inconsistent practices in the field.

### **1.1.2 Lack of standardized evaluation methods**

Model evaluation is further complicated by the dominance of point-based metrics such as accuracy, RMSE, or R-squared. While these metrics are widely used, they offer only a narrow perspective—focusing on single-value predictions while ignoring uncertainty. In domains like forecasting, healthcare, or risk analysis, understanding the spread or distribution of possible outcomes is often more critical than pinpointing a single value (Slingo & Palmer, 2011). Without standardized methods that incorporate probabilistic reasoning, evaluations remain shallow, leading to suboptimal model selection and limited real-world applicability.

**1.1.3 Misleading guidance from traditional metrics**

Exclusive reliance on traditional metrics can lead to misleading conclusions, particularly when evaluating models’ generalization performance. Metrics like accuracy or RMSE do not capture the distributional characteristics of prediction errors or express uncertainty. This shortfall is particularly problematic when models are used in decision-making scenarios, where overconfident or under-quantified predictions can have significant consequences (Hüllermeier & Waegeman, 2021). A model that performs well on point estimates but poorly represents uncertainty can easily misguide stakeholders, especially in high-risk environments.

**1.1.4 Difficulty in Comparing Diverse Models**

Finally, comparing models from different families (e.g., decision trees vs. Bayesian models) remains a substantial challenge. Each model may use different evaluation metrics or output formats, making fair comparisons difficult. Without a standardized framework that accommodates the probabilistic nature and structure of various models, practitioners are left with ad hoc solutions that lack rigor and consistency. This fragmentation hampers effective model selection and the broader goal of reproducible, transparent science.

## **1.2 Project aims**

This project aims to develop an R package, **Scoriverse**, that provides a unified, standardized, and extensible framework for generating, extracting, and evaluating predictions with robust uncertainty quantification across a wide range of regression and machine learning models. The core objectives are structured around the following guiding goals:

* **Unified prediction interface** across model types, streamlining prediction workflows.
* **Comprehensive uncertainty quantification**, ensuring users can assess both accuracy and confidence.
* **Standardized output format** for point estimates, standard errors, and prediction intervals.
* **Tight integration with popular R modeling ecosystems**, including lm(), glm(), mgcv::gam(), lme4::lmer(), and others.
* **Extensible modular design** to support future contributions and community-driven enhancements.

To achieve these aims, the development process will be decomposed into the following core components:

### **1.2.1 Prediction Standardization**

This module will focus on ensuring consistent prediction outputs across diverse model classes. It will:

* Generate standardized point estimates, standard errors, and confidence or prediction intervals.
* Provide predictions on both the link and response scales, with support for transformation functions.
* Harmonize outputs from linear models, generalized linear models, mixed-effect models, and additive models.

### **1.2.2 Scoring and Evaluation Framework**

To enable model comparison and performance assessment, this module will implement a suite of robust scoring rules, tailored for both probabilistic and classification outputs:

* **Probabilistic Metrics:** Logarithmic Score, Continuous Ranked Probability Score (CRPS), Dawid–Sebastiani Score, and Energy Score.
* **Classification Metrics**: Brier Score, Multiclass Brier Score, and Zero-One Loss.
* **Forecast Quality Metrics**: Ranked Probability Score, Spherical Score, and Interval Score.
* Each scoring method will include uncertainty quantification for the evaluation metrics themselves.
* Comparative evaluation tools will be provided to benchmark multiple models on equal footing.

### **1.2.3 Visualization and Interpretation**

Recognizing the importance of interpretability, Scoriverse will include tools to visualize prediction uncertainty and performance:

* Prediction interval plots, model comparison charts, and calibration curves.
* Seamless integration with ggplot2 for high-quality static plots.

### **1.2.4 Performance Optimization and Scalability**

Given the increasing size and complexity of modern datasets, performance is a key consideration:

* Efficient handling of large-scale predictions and scoring tasks.
* Support for parallel processing to accelerate computation.
* Memory-aware design principles to maintain usability on limited-resource systems.

### **1.2.5 Modularity and Extensibility**

Scoriverse will follow a modular architecture to facilitate long-term maintainability and community engagement:

* Well-defined modules for prediction, scoring, visualization, and data handling.
* Robust error-handling, input validation, and clear API boundaries.
* Plugin-ready structure allowing developers to contribute support for new model types, scoring rules, or visualization methods.

## **1.3 Structure of progress report**

This report consists of 7 chapters, reflecting the development stages of the Scoriverse R package.

* Chapter One provides an overview of the Scoriverse project, introducing the motivation behind the development, existing challenges in prediction workflows, and the core goals of the package.
* Chapter Two explores the technical background of prediction standardization and uncertainty quantification in statistical modeling. It also reviews related R packages and highlights the gaps Scoriverse aims to address.
* Chapter Three outlines the architectural design of the package, presenting the high-level structure, core modules, and the design decisions that enable extensibility and consistency across models.
* Chapter Four documents the API blueprint, detailing the user-facing interface, function design, input/output specifications, and integration with popular modeling workflows in R.
* Chapter Five focuses on the development of core components, including the prediction extraction engine, scoring framework, and visualization tools. It also describes the unit testing strategies employed to ensure robustness and accuracy.
* Chapter Six discusses ongoing and upcoming work, including performance optimization, enhanced uncertainty quantification, community feedback mechanisms, and contribution guidelines.
* Chapter Seven concludes the report with reflections on the current progress, the challenges encountered, and the broader impact of Scoriverse on the R modeling ecosystem. It also outlines future directions and plans for community engagement and package maintenance.
* Finally, the appendix includes supplementary technical details, code snippets, testing logs, and configuration settings that support the main chapters but are too detailed for the main text.

# **2. Background**

Standardizing prediction outputs and quantifying uncertainty are foundational aspects of modern statistical modelling and data-driven decision-making (Jordan & Mitchell, 2015). While the R ecosystem offers a wide variety of tools for model estimation and prediction, there remains a notable gap in cohesive infrastructure for extracting, evaluating, comparing, and communicating predictions across diverse modelling approaches.

This chapter introduces the theoretical and practical foundations of the Scoriverse package.

* Section 2.1 discusses the importance of prediction standardization.
* Section 2.2 covers uncertainty quantification in predictive modeling.
* Section 2.3 reviews related packages in the R ecosystem.
* Section 2.4 identifies existing gaps and outlines how Scoriverse addresses them.

## **2.1 The importance of prediction standardization in modelling**

In applied statistical modelling, predictions are often the primary product used to inform decisions, evaluate performance, or communicate findings. Yet, despite their central role, prediction outputs remain surprisingly inconsistent across model classes (Gneiting & Katzfuss, 2014). This inconsistency spans not only the format and structure of the output, but also the scale (e.g., link vs. response), the meaning of uncertainty, and even the interpretation of model-specific prediction types.

Such variation complicates downstream tasks that are otherwise common across modelling workflows: comparing predictions from different models, computing predictive scores, visualizing outputs with uncertainty, and integrating predictions into decision analysis or reporting pipelines (Gundersen & Kjensmo, 2018). Without a common structure or grammar, analysts are forced to write repetitive and fragile custom code to bridge these differences—code that is difficult to reuse, scale, or share.

This project considers a wide range of models, each with its own prediction conventions:

* Linear and Generalized Linear Models: lm(), glm()
* Generalized Additive Models: mgcv::gam()
* Mixed Effects Models: lme4::lmer()
* Tree-based Models: randomForest, gbm, xgboost
* Regularized Regression: glmnet

These models differ not only in how they generate predictions but also in their assumptions, the meaning of uncertainty, and how their results are typically evaluated. For example:

* GLMs and GAMs may default to predictions on the link scale, requiring manual transformation to the response scale.
* Tree-based models may provide point estimates but no uncertainty unless bootstrapping or simulation is applied.
* Mixed effects models might require parsing of random vs. fixed effect contributions depending on the use case.

The lack of standardization introduces friction into otherwise routine tasks. Moreover, it becomes a serious limitation when attempting to:

* Evaluate models side-by-side using scoring rules.
* Ensemble or stack models.
* Apply resampling-based workflows (e.g., cross-validation, bootstrapping).
* Communicate model results to non-technical audiences.

A standardized prediction grammar would abstract away these model-specific differences, allowing users to interact with predictions in a unified, predictable way—regardless of the underlying model. It would enable scalable tooling, more reproducible workflows, and easier integration with visualization and scoring libraries.

Standardization is not about hiding the complexity of modelling choices—it’s about encapsulating them in a way that makes pipelines more robust, interpretable, and extensible (Machine Learning Pipelines: From Prototype to Production | Outerbounds, 2022).

## **2.2 Understanding uncertainty in model predictions**

Uncertainty is a fundamental aspect of any predictive model (Ghahramani, 2015). While point estimates offer insight into central tendencies, they often obscure the model’s confidence—potentially misleading decision-makers about the reliability of predictions. Understanding and communicating uncertainty is critical for several reasons:

* Risk-aware decision-making: Decisions such as resource allocation or policy planning depend not just on the most likely outcomes, but on a full understanding of possible scenarios.
* Robust model evaluation: Scoring rules like the Continuous Ranked Probability Score (CRPS) or the log score require full predictive distributions to assess model performance effectively.
* Model comparison and selection: Without standardized uncertainty measures, it becomes difficult to fairly compare models or construct ensembles.
* Transparent scientific communication: Clear communication of uncertainty enhances interpretability and trust in model outputs (Bhatt et al., 2020).

Despite its importance, uncertainty quantification remains fragmented across modeling frameworks. Different model families express uncertainty in distinct ways, often requiring custom logic and non-trivial assumptions to extract usable information:

|  |  |
| --- | --- |
| **Model Type** | **Uncertainty Handling** |
| Bayesian Models (brms) | Naturally produce posterior predictive distributions via functions like posterior\_predict(), providing rich uncertainty information ready for scoring or visualization. |
| Frequentist GLMs and GAMs | Provide standard errors on the link scale; observation-level uncertainty requires back-transformation (e.g., exp(λ) for Poisson) and simulation (e.g., using rpois()), often involving complex assumptions. |
| Tree-Based Models (randomForest, xgboost) | Typically output point predictions without uncertainty; require bootstrapping or quantile regression forests to approximate uncertainty. |
| Mixed Effects Models (lme4::lmer) | Offer standard errors for fixed and random effects but may not yield full predictive distributions suitable for scoring or sampling. |
| Regularized Regression (glmnet) | Prioritizes predictive accuracy and typically lacks built-in uncertainty estimates or intervals. |

This fragmented ecosystem poses challenges for automating evaluation, applying probabilistic scoring, and generalizing workflows across models. To address these issues, a standardized framework for predictive uncertainty should adhere to the following principles:

1. Unified access to uncertainty: All models should expose not only point estimates but also methods for sampling or approximating full predictive distributions.
2. Clear scale handling: Outputs should indicate whether they are on the link or response scale, with tools for seamless transformation between them.
3. Fallback estimation strategies: For models lacking analytic uncertainty (e.g., tree-based methods), standardized techniques like bootstrapping or quantile estimation should be consistently available.
4. Structured prediction objects: Outputs should encapsulate both predictions and uncertainty metadata (e.g., standard errors, quantiles, samples) in formats easily consumed by downstream tools.
5. Scoring compatibility: Uncertainty outputs should integrate directly with scoring functions such as CRPS, log score, and interval score, avoiding the need for model-specific pipelines.
6. Graceful degradation: Where full uncertainty estimates are not feasible, the system should fall back to approximate methods or alert the user to potential limitations.

Adhering to these principles can foster a “grammar of uncertainty” that improves automation, evaluation, and communication across modelling tasks—ultimately enhancing both reliability and transparency in predictive analytics.

## **2.3 Review of related packages**

The R ecosystem offers a variety of mature packages that address specific aspects of the predictive modelling workflow, from generating predictions to scoring uncertainty and visualizing results. While these tools are individually powerful, they often function in isolation—making it difficult to build unified, reproducible pipelines for end-to-end model evaluation. Below is a brief overview of key packages and how **Scoriverse** enhances and integrates their functionality.

### **2.3.1 marginaleffects**

The marginaleffects package provides a streamlined interface for computing predictions and marginal effects across a wide range of models, including those created with brms, glm, and others (Arel-Bundock et al., 2024). It offers standardized access to fitted values, supports both average and conditional predictions, and respects model-specific structures. However, its focus remains primarily on expectations and marginal effects—often returning outputs on the link scale. For full probabilistic scoring, users must manually simulate predictive distributions, particularly for discrete or count models. **Scoriverse** extends this functionality by wrapping marginaleffects::predictions() and post-processing its outputs into observation-level simulations suitable for scoring workflows.

### **2.3.2 scoringRules**

The scoringRules package implements a suite of proper scoring metrics such as the Continuous Ranked Probability Score (CRPS), Dawid-Sebastiani Score (DSS), and the logarithmic score—critical tools for evaluating full predictive distributions (Jordan et al., 2017). While statistically robust, the package assumes users provide predictive samples in a specific format. This places the burden of simulation and formatting on the user.  
**Scoriverse** simplifies this process by integrating prediction and scoring workflows, automatically transforming simulation outputs into formats compatible with scoringRules.

### **2.3.3 stats**

Base R’s stats::predict() function supports predictions for a broad set of models, including lm, glm, and others (Venables & Ripley, 2013). It provides fitted values and, in some cases, standard errors or confidence intervals. However, it lacks support for distributional outputs or robust uncertainty quantification—particularly in non-Gaussian or hierarchical models.  
**Scoriverse** addresses this limitation by wrapping predict() in a higher-level abstraction (wrap\_predict()), which standardizes prediction outputs and enables simulation-based uncertainty estimates where applicable.

### **2.3.4 ggplot2**

ggplot2 is the de facto standard for data visualization in R, offering a flexible, grammar-based approach for building custom plots (Wickham, 2016). It excels in plotting predictions, residuals, and diagnostics—but crafting tailored visualizations such as prediction intervals or calibration plots often requires verbose code and deep knowledge of ggplot2 internals.  
**Scoriverse** builds on ggplot2 by providing intuitive, pre-configured plotting functions (visualize\_predictions(), visualize\_residuals()), reducing boilerplate and making uncertainty visualization more accessible.

While each of these packages addresses a specific stage in the predictive modelling pipeline, they lack seamless integration. In real-world workflows, analysts must frequently transform prediction types, simulate observation-level outcomes, apply scoring metrics, and generate visualizations—often across diverse model types. This leads to repetitive code, custom logic, and inconsistent practices.

**Scoriverse** introduces a cohesive framework that unifies these components, offering:

* **Standardized prediction workflows**
  + Functions like prepare\_model\_for\_prediction() and extract\_predictions() enable consistent simulation of uncertainty-aware outputs across model types.
* **Integrated scoring utilities**
  + Tools such as wrap\_scoring(), compute\_crps(), and compute\_log\_score() streamline the application of proper scoring rules.
* **Built-in visualization support**
  + Ready-to-use functions like visualize\_predictions() and visualize\_residuals() generate interpretable plots directly from model objects.
* **End-to-end reproducibility**
  + The run\_scoriverse() function enables complete evaluation workflows with minimal user input, promoting transparency and repeatability.

By abstracting the quirks of individual packages and model classes, Scoriverse delivers a consistent, extensible, and user-friendly solution for uncertainty-aware model evaluation.

## **2.4 Identified gaps**

Despite the breadth of modelling tools in R, key gaps remain:

* Lack of a unified grammar: There is no consistent interface or vocabulary for generating, handling, or comparing predictions across models.
* Fragmented functionality: Core tasks such as scoring, simulation-based evaluation, and uncertainty visualization are scattered across packages, leading to boilerplate code and custom solutions.
* Limited support for multi-model workflows: Ensembling, comparative diagnostics, and batch simulations often require ad-hoc scripting.
* Barriers to extension: Many packages are not designed with community extension in mind, making it difficult for researchers or educators to prototype new evaluation methods or customize workflows.

Scoriverse is designed to address these gaps through a modular and extensible architecture. The package emphasizes:

* A cohesive API for prediction extraction and transformation.
* Plug-and-play scoring methods and uncertainty-aware evaluation tools.
* Seamless integration with the broader R modeling ecosystem.
* A community-oriented design that supports teaching, research, and collaborative development.

These features aim to benefit analysts, researchers, and package developers by reducing boilerplate, improving reproducibility, and enabling richer insight from predictive models.

# **3. Architectural Design and Core Modules**

This chapter outlines the internal structure of the **Scoriverse** package. It explains the modular design approach, key functional components, and the rationale behind choices that balance extensibility, usability, and consistency. The goal is to provide transparency into how different parts of the system interact and support future development.

## **3.1 High-Level Overview of the Architecture**

In this section, we’ll explain the overall organization of the **Scoriverse** package, breaking it down into major components, their roles, and how they interact to provide a cohesive experience for users. This will help to contextualize the different parts of the package and show how they contribute to the larger system.

**Scoriverse** is structured to provide a streamlined workflow for prediction extraction, uncertainty quantification, scoring, and visualization. The core components of the package are organized as follows:

### **3.1.1 scoriverse\_main.R**

This file contains the primary API functions that users interact with. It provides a master function to run the entire Scoriverse workflow, integrating the various components of prediction extraction, scoring, and visualization. Users can call this function to perform all the necessary steps, from loading models to extracting predictions and evaluating them with appropriate scoring metrics.

### **3.1.2 prediction\_extraction.R**

This file is responsible for extracting predictions from different model types. It uses functions from the marginaleffects package and custom wrapper functions to standardize the prediction extraction process. It handles different types of models (e.g., linear models, generalized linear models, random forests, etc.) and ensures that predictions are obtained consistently, regardless of the model type.

### **3.1.3 scoring\_functions.R**

This file contains the scoring metrics used for evaluating model predictions. It implements wrappers around proper scoring rules such as the Continuous Ranked Probability Score (CRPS), Logarithmic Score, and others from the scoringRules package. These scoring functions are applied to probabilistic predictions and are central to assessing the model's performance.

### **3.1.4 visualizations.R**

This file contains functions for visualizing the predictions and residuals from the models. It leverages ggplot2 to create various plots, such as observed vs. predicted values, uncertainty intervals, and model comparison graphics. These visualizations provide insights into how well the model fits the data and help in the evaluation of the predictive performance.

### **3.1.5 scoriverse-package.R**

This file manages the internal package metadata. It handles the necessary imports and global variable declarations that prevent R CMD check warnings. It also imports essential functions from other packages (like ggplot2 and stats) to ensure smooth integration of the different modules. Furthermore, it suppresses warnings related to global variables that are used within the package (e.g., observed, predicted), making the code more compatible with R’s internal checks.

### **3.1.6 wrappers.R**

If required, this file contains custom wrapper functions that harmonize the interfaces of different model-fitting functions. These wrappers ensure that predictions are consistently handled across different model types, allowing the user to work with a unified prediction extraction process without worrying about the specifics of each model.

## **3.2 Core Modules of the Package**

Down below is..

### **3.2.1 scoriverse\_main.R**

At the heart of the **Scoriverse** package is the master function, which is housed in scoriverse\_main.R. This function (run\_scoriverse()) orchestrates the entire process, calling sub-functions across different modules to perform the following:

* **Prediction Extraction**: The function first invokes the appropriate prediction extraction methods based on the model class (using functions from prediction\_extraction.R).
* **Scoring**: Once predictions are extracted, run\_scoriverse() triggers scoring routines (using functions from scoring\_functions.R) to compute evaluation metrics such as CRPS, log score, and interval score.
* **Visualization**: The function also generates plots, which are important for interpreting uncertainty, calibration, and residuals. These visualizations are produced by the visualization.R module.
* **Result Output**: Finally, run\_scoriverse() compiles results and outputs them in a user-friendly format, often as tables or ggplot objects for further analysis.

### **3.2.2 prediction\_extraction.R**

This file contains functions that handle the extraction of predictions from various types of model objects. These models can range from simple linear models (lm) to more complex machine learning models (xgboost, randomForest).

The extraction process often varies by model class. For example:

* **Linear and Generalized Linear Models**: Prediction is straightforward and typically involves calling predict() on the model object.
* **Bayesian Models**: For models like those fitted with brms, the posterior\_predict() function is used to obtain samples from the posterior predictive distribution.

The goal of this module is to normalize the extraction process so that predictions are returned in a standardized format across model types, which simplifies downstream processing.

### **3.2.3 scoring\_functions.R**

The scoring functions module provides wrappers for different **proper scoring rules** that evaluate the quality of probabilistic predictions. Some of the metrics implemented include:

* **CRPS** (Continuous Ranked Probability Score)
* **Log Score**
* **Brier Score**
* **Interval Score**
* Dawid-Sebastiani Score (DSS)

These scoring rules require full predictive distributions, not just point estimates. As part of this module, the scoring functions handle both the extraction of relevant prediction distributions and the computation of these metrics.

For example, the CRPS is computed based on the cumulative distribution function (CDF) of the predicted values. This file ensures that the metrics are computed correctly regardless of the model type and its specific output format.

### **3.2.4 visualization.R**

This file contains functions that generate plots to help users understand and interpret their model's performance. It supports the following visualizations:

* **Prediction vs. Actual Plots**: Scatter plots or line plots to visualize how well predicted values match actual values.
* **Uncertainty Visualizations**: Interval plots to show the uncertainty in predictions.
* **Residuals Plots**: To identify model fit issues or outliers.

These plots are built using ggplot2, leveraging its robust capabilities to create customizable and publication-ready visuals. The user only needs to call high-level functions like visualize\_predictions() to generate insightful plots.

### **3.2.5 scoriverse-package.R**

The **scoriverse-package.R** file plays a crucial role in managing the internal structure and metadata of the Scoriverse package. While not containing model-specific logic or functionality, it ensures that the package runs smoothly by addressing the following key aspects:

* **Package Imports**:
  + This file imports necessary functions from other R packages such as ggplot2, stats, and marginaleffects. These imports are essential for Scoriverse's functionality, providing the core tools for prediction extraction, visualization, and statistical analysis.
* **Global Variable Declarations**:
  + The file suppresses R CMD check warnings regarding global variables like observed and predicted. By using the utils::globalVariables() function, it ensures that R does not flag these as issues during package checking, thus maintaining compatibility with R's internal checks.
* **Error Handling**:
  + While not as complex as in utils.R, scoriverse-package.R ensures that all necessary dependencies are imported and that the package structure adheres to R's standards. This file is designed to minimize issues during package building and testing.
* **Logging and Debugging**:
  + Although scoriverse-package.R does not explicitly handle extensive logging, it helps facilitate a smooth user experience by ensuring the internal components of the package are properly connected and free of metadata issues.

In essence, scoriverse-package.R is integral to the functionality and stability of the package, ensuring that the code runs without warnings or errors during the build process. It manages dependencies, suppresses unnecessary checks, and provides the foundational structure for the rest of the package to operate.

### **3.2.6 wrappers.R**

This module contains custom wrapper functions that standardize interactions with different model types. For instance, it might contain functions to harmonize how predictions are extracted from glm vs. xgboost, or to convert the predictions into a suitable format for scoring or visualization.

This wrapper layer is key to enabling **Scoriverse** to work consistently with a wide variety of models, making it flexible and extensible. Users do not need to worry about the inner workings of each individual model class, as the wrappers ensure that predictions are handled in a consistent manner.

## **3.3 Design Decisions for Extensibility and Consistency**

A key design philosophy behind **Scoriverse** is to provide users with a unified, extensible framework that integrates with existing R packages while allowing future model classes to be added with minimal effort. To ensure this, the following design decisions were made:

1. **Modular Architecture**: Each module has a clear and distinct responsibility. This promotes clean code, easy debugging, and allows individual components to be easily extended or modified.
2. **Consistency Across Models**: By creating custom wrappers (wrappers.R) and standardizing the format of predictions, **Scoriverse** ensures that different models can be used interchangeably without requiring users to adjust their workflows.
3. **Extensibility: Scoriverse** was designed with future model types and scoring metrics in mind. New models can be added by creating additional prediction extraction functions, and new scoring rules can be added by implementing new wrappers in the scoring\_functions.R module.
4. **Seamless Integration**: The package seamlessly integrates with other R packages like marginaleffects and scoringRules, reducing the need for users to manually handle data formatting, prediction extraction, or scoring rule implementation.

4.

* Chapter Four documents the API blueprint, detailing the user-facing interface, function design, input/output specifications, and integration with popular modeling workflows in R.

5.

* Chapter Five focuses on the development of core components, including the prediction extraction engine, scoring framework, and visualization tools. It also describes the unit testing strategies employed to ensure robustness and accuracy.

6.

* Chapter Six discusses ongoing and upcoming work, including performance optimization, enhanced uncertainty quantification, community feedback mechanisms, and contribution guidelines.

7.

* Chapter Seven concludes the report with reflections on the current progress, the challenges encountered, and the broader impact of Scoriverse on the R modeling ecosystem. It also outlines future directions and plans for community engagement and package maintenance.

# **Reference**

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction, Second Edition*. Springer New York.

‌ (Hastie et al., 2009)

Kuhn, M., Johnson, K., & Springer Science+Business Media. (2016). Applied predictive modeling. Springer.

(Kuhn et al., 2016)

Gneiting, T., & Katzfuss, M. (2014). Probabilistic Forecasting. Ssrn.com. https://doi.org/10.1146/annurev-statistics-062713-085831

(Gneiting & Katzfuss, 2014)

Gundersen, O. E., & Kjensmo, S. (2018). State of the Art: Reproducibility in Artificial Intelligence. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11503

(Gundersen & Kjensmo, 2018)

Slingo, J., & Palmer, T. (2011). Uncertainty in weather and climate prediction. Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences, 369(1956), 4751–4767. https://doi.org/10.1098/rsta.2011.0161

(Slingo & Palmer, 2011)

Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Machine Learning, 110(3), 457–506. https://doi.org/10.1007/s10994-021-05946-3

(Hüllermeier & Waegeman, 2021)

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255–260. https://doi.org/10.1126/science.aaa8415

(Jordan & Mitchell, 2015)

Machine Learning Pipelines: from Prototype to Production | Outerbounds. (2022). Outerbounds.com. https://outerbounds.com/blog/machine-learning-pipelines-from-prototype-to-production

(Machine Learning Pipelines: From Prototype to Production | Outerbounds, 2022)

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. Nature, 521(7553), 452–459. https://doi.org/10.1038/nature14541

(Ghahramani, 2015)

Bhatt, U., Zhang, Y., Antorán, J., Liao, Q. V., Prasanna Sattigeri, Riccardo Fogliato, Melançon, G. G., Krishnan, R., Stanley, J., Omesh Tickoo, Nachman, L., Rumi Chunara, Weller, A., & Xiang, A. (2020). Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty. ArXiv (Cornell University). https://doi.org/10.48550/arxiv.2011.07586

(Bhatt et al., 2020)

Arel-Bundock, V., Greifer, N., & Heiss, A. (2024). How to Interpret Statistical Models Using marginaleffects for R and Python. Journal of Statistical Software, 111(9). https://doi.org/10.18637/jss.v111.i09

(Arel-Bundock et al., 2024)

Jordan, A., Krüger, F., & Lerch, S. (2017). Evaluating probabilistic forecasts with scoringRules. ArXiv (Cornell University). https://doi.org/10.48550/arxiv.1709.04743

(Jordan et al., 2017)

Venables, W. N., & Ripley, B. D. (2013). *Modern Applied Statistics with S-PLUS*. Springer Science & Business Media.

(Venables & Ripley, 2013)

Wickham, H. (2016). Ggplot2 : Elegant Graphics For Data Analysis. Springer-Verlag New York. https://ggplot2.tidyverse.org

(Wickham, 2016)