# Smart Applications: My Update
# Davide Marchi
# January 2025

Enhancing Forecasting System with
Explainability and Performance

IN SUPREMÆ DIGNITATIS · 1343 ·
UNIVERSITÀ DI PISA

# Introduction

Smart Factory

I worked on Team A: topic 7, handling all the aspects regarding explainability.

- `ForecastExplainer` → Generates bounds and local explainability
- `RagExplainer` → Connects context to LLM's answers

**Possible improvements**:

- Forecasting system lacked global explainability.
- Response times for predictions and explanations were improvable.

All the updates and the code developed can be found on my GitHub repository.

# Pre-Update State

**Local Explainability**:

- Implemented with **LIME** (perturbation-based).
- Model-agnostic but computationally expensive.

**Uncertainty Bounds**:

- Generated with **bootstrapping** (perturbation-driven).
- Time-consuming approach.

| Environment | Sequence Length | Predictions | Time (seconds) |
|---|---|---|---|
| Local Machine | 15 | 30 | 2.93 |
| Website (with GUI) | 15 | 30 | 4.60 |

# Update 1: Global Explainability

**Approach**:

- Introduced SHAP-based global explainability.
- Computed SHAP values for the dataset using `shap.TreeExplainer`.

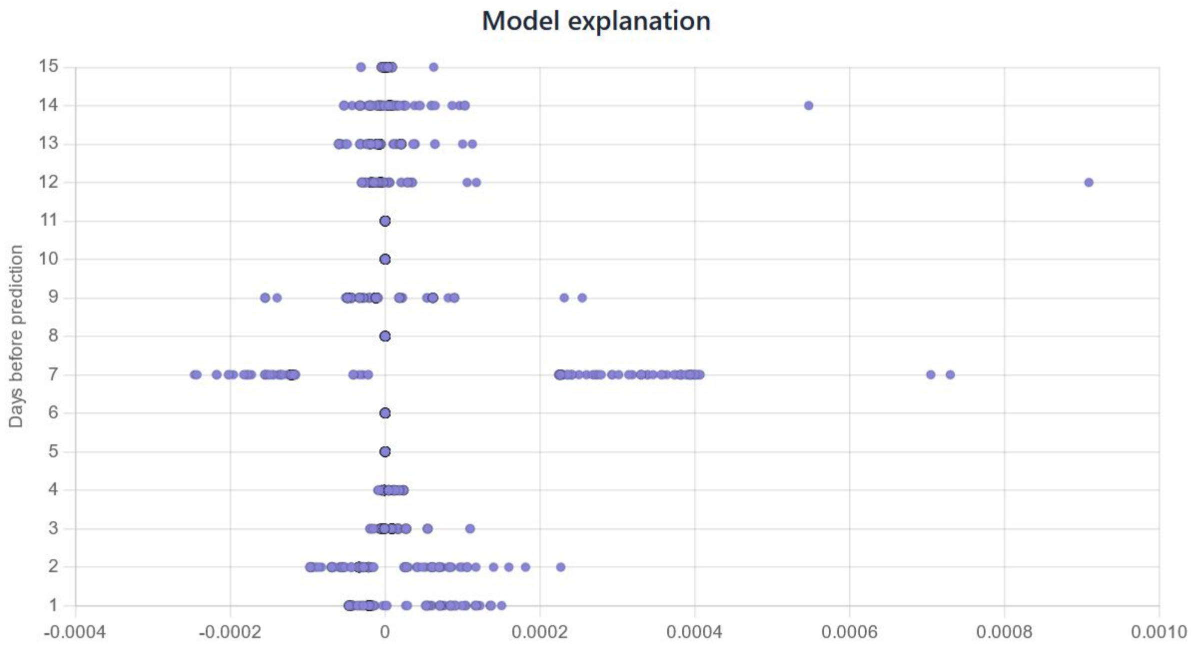
**Advantage**:

- Provides users with insights into overall feature importance.

**Feature Explanation**:

- Visual representation of SHAP values for feature importance (Spread-out points: High feature impact, Points near zero: Minimal impact)

# Bee Swarm Plot for Global Explainability




**This chart illustrates the overall behavior of the model, showing how different days generally affect predictions. Spread-out dots indicate significant impact, while dots near zero show minimal effect.**

Bee swarm plot of SHAP values for feature importance across all the dataset.

# Codebase Changes for Global Explainability

Implementing this feature required understanding and modifying the codebases across multiple topics:

| | |
|---|---|
| **Topic 7** | Updated `ForecastExplainer` to calculate and return global SHAP values alongside predictions. |
| **Topic 4** | Interface classes managing communication between the backend and GUI were updated to include global SHAP values in their structures. |
| **Topic 6** | A new GUI section was added to display the global SHAP values as a bee swarm plot, with clear instructions for interpreting the plot. |

# Update 2: Performance Optimization

**Approach**:

- Replaced LIME with SHAP for local explainability of predictions.
- Used **`shap.TreeExplainer`** for XGBoost, reducing perturbation steps.

**Result**:

- Execution times were significantly reduced in both test and real-world scenarios:
  a. Local: 94.54% reduction.
  b. Website: 82.17% reduction.

# Performance Measurements

Summary of Results (obtained with 10 measures averaged):

- Local Machine: Before: 2.93s → After: 0.16s.
- Website: Before: 4.60s → After: 0.82s.

| Environment | Sequence Length | Predictions | Method | Time (seconds) | Reduction (%) |
|---|---|---|---|---|---|
| Local Machine | 15 | 30 | LIME + Bootstrapping | 2.93 | - |
| Local Machine | 15 | 30 | SHAP + Bootstrapping | 0.16 | 94.54 |
| Website (with GUI) | 15 | 30 | LIME + Bootstrapping | 4.60 | - |
| Website (with GUI) | 15 | 30 | SHAP + Bootstrapping | 0.82 | 82.17 |

# Alternative Approaches

**Residuals-Based Bounds**:

- Residuals-based bounds were explored and developed as an alternative.
- Allows for further executions time reduction.
- Challenges:
    a. Lack of a validation set for proper residual estimation.
    b. Reliance on normally distributed residuals (not guaranteed in this scenario).

**Future Potential**:

- This approach is suitable for scenarios with better-controlled data, where some samples can be reserved exclusively for residual computation.

# Summary of Updates

**Key Achievements**:

- Introduced **global explainability** with SHAP values.
- Enabled deeper study of model behavior.
- Transitioned from LIME to SHAP for **faster local explanations**.
- Execution times were reduced by over 80%, now consistently under 1 second.

**Final Output**:

- Faster, more transparent, and user-friendly forecasting system.
- Provided a foundation for future updates and improvements.