Relevant Workflow Examples for a Dissertation Project on Expert-Knowledge-Enhanced Financial Forecasting through Visual Analytics

Recommendations Based on an ATWL Workflow Library

1 Introduction

This document identifies published visual analytics workflows that are relevant to a dissertation project on supporting the use of human expert knowledge for improving financial forecasting. The project centres on the *semantic gap* between a forecasting analyst's causal domain knowledge and the feature-engineering decisions (driver selection and transformation) that a machine-learning model needs. The core experimental task—iterative, visually supported feature engineering for a regression-type forecasting model—has strong parallels in several workflows from a reference library of 17 VA workflows formalised in the ATWL language.

The recommendations are organised by the aspect of the project they address: the core analytical task (§2.1), time-series modelling (§2.4), human–system interaction (§2.6), model exploration (§2.8), feature-effect probing (§2.10), and prescriptive feedback (§2.12). A summary mapping and concrete suggestions for experimental-condition design are given in §2.14 and §2.15.

2 Recommendations

2.1 The Feature-Selection and Model-Building Task

2.2 Workflow 1.10 — Partition-Based Regression Modelling

Source: Mühlbacher & Piringer [6].

2.2.1 Why It Is the Single Most Relevant Reference

This workflow describes *exactly* the analytical task the project's participants will perform: iteratively selecting features ("drivers") and their transformations to build a regression model, guided by visual feedback about feature–target relationships.

2.2.2 What to Study in Detail

The relevance-ranking mechanism. Features are ranked by a quantitative goodness-of-fit measure computed over partitioned feature domains. In the experiment, the analogous step is showing participants which candidate drivers (and which transformations—lag, rolling average, etc.) have the strongest statistical association with the target metric. The three experimental conditions can be seen as three different *representational forms* of this same relevance information.

The conditional-distribution visualisation. The workflow uses "ranked small-multiple partition-based plots"—for each feature, a one-dimensional percentile plot shows how the target variable's distribution changes across partitioned regions of that feature's domain. This is a *visual statistical encoding*: it goes beyond a single relevance number and shows the *shape* of the relationship (non-linearity, local effects, interactions). This design directly informs **Condition B** (visual statistical encoding)—it could be adapted to show how the target metric's distribution varies across partitioned regions of each candidate driver.

The residual-based refinement loop. After building an initial model, the analytical target switches to residuals. Features are re-ranked by their relevance to the residuals, exposing effects not yet captured. Previously included features rank lower, while new relevant features emerge. This is the mechanism by which the workflow helps the analyst progressively close the gap between what the model captures and what remains unexplained. For the experiment, this suggests a key design element: after a participant selects initial drivers and builds a model, the system could show residual-based feedback revealing which additional drivers or transformations would most reduce the remaining error.

The feature-pair interaction displays. The workflow includes two-dimensional colour-coded plots showing conditional target statistics over pairs of features, revealing interaction effects. In the forecasting context, this could correspond to showing how combinations of drivers (or a driver with a specific transformation) jointly relate to the target—information that could be especially valuable for **Condition C** (semantic visual encoding) if the interactions are presented with domain-meaningful labels and causal framing.

2.2.3 How to Use It

Study this workflow as the *structural template* for the analytical task participants will perform. The three experimental conditions can be understood as providing the same underlying information that this workflow computes (feature relevance, conditional distributions, residual patterns) but at three levels of representational richness.

2.3 Workflow 1.12 — Feature Engineering for Behaviour Pattern Recognition

Source: Andrienko et al. [2].

2.3.1 Why It Matters

Apart from the general human-in-the-loop philosophy emphasised in the paper, the *workflow structure* it describes contains specific design patterns directly applicable to the experimental interface.

2.3.2 What to Study in Detail

The feature-engineering loop. The workflow has an explicit iterative loop: compute features → cluster by feature similarity → project and visualise → interpret patterns → assess feature adequacy → *refine features* (add new ones, apply transformations such as logarithmic scaling) → repeat. This is structurally identical to what participants will do: select drivers and transformations → build model → receive visual feedback → refine selections → repeat. The key insight is that **feature refinement itself is inside the loop**, not a one-shot decision. The experimental interface should support this iterative cycle.

The role of coordinated views for assessment. The workflow uses three coordinated views simultaneously: a UMAP projection showing episode similarity, a geographic map showing trajectory shapes, and histograms showing feature distributions. The projection reveals whether clusters are well-separated (analogous to showing whether chosen drivers create a good model), the map provides *domain-semantic* context (analogous to showing the causal meaning of a driver), and the histograms provide *statistical* detail. This three-view structure maps suggestively onto the three experimental conditions—they might be thought of as progressively adding views from this coordinated set:

- **Condition A:** Only statistical summary (like having only the histograms).
- **Condition B:** Statistical visualisation (like adding the projection—showing the *shape* of statistical relationships).
- **Condition C:** Semantic visualisation (like adding the map—providing domain-meaningful context that helps interpret *why* certain features work).

The feature-assessment artifact. The workflow explicitly models the human's assessment of feature adequacy as an artifact that records whether current features "achieve meaningful behavioural discrimination." In the experiment, this corresponds to the participant's judgment of whether their current driver selection is adequate—a judgment that the VA feedback is meant to support.

2.3.3 How to Use It

Use this workflow's structure as the basis for *designing the interaction flow* of the experimental interface. Regardless of condition, all participants should follow the same loop structure; what changes across conditions is the representational richness of the visual feedback within each iteration.

2.4 Time-Series Modelling and Temporal Relationship Visualisation

2.5 Workflow 1.11 — Spatio-Temporal Analysis and Modelling

Source: Andrienko & Andrienko [1].

2.5.1 Why It Is Relevant

The forecasting scenario involves time-series data with temporal relationships (drivers influencing the target with delays, cyclical patterns). This workflow addresses exactly this: modelling temporal variation in time series, including identifying periodicity, cycle lengths, and trends through visual inspection.

2.5.2 What to Study in Detail

Visual identification of temporal characteristics. The analyst inspects time graphs to identify periodicity, cycle lengths, and trends. In the forecasting context, this corresponds to participants visually recognising the temporal relationship between a driver and the target—e.g., seeing that marketing spend leads subscription sign-ups by two weeks. For **Condition C**, this visual recognition could be supported by time-graph displays that *align* the driver and target series with explicit lag indicators, making the temporal causal relationship visually apparent.

The model-curve-versus-data visualisation. After fitting a model, the model's predicted curve is overlaid on the actual data and the analyst assesses fit visually. This is a powerful feedback mechanism that could be adapted for the experiment: after a participant selects drivers and transformations, show the resulting model's predicted target series overlaid on the actual target, so they can *see* where the model succeeds and fails.

The residual visualisation. Residuals are displayed both temporally (quintile summary bands over time) and spatially. For this project, the temporal residual display is key: it can reveal *when* the model fails, helping participants reason about which drivers or temporal transformations might address the gap. For example, if residuals show a systematic pattern at certain times of year, this might prompt the participant to add a seasonal driver or adjust a lag.

2.5.3 How to Use It

Draw on the time-graph and model-overlay visualisation designs from this workflow when designing the visual feedback in the experimental conditions. The concept of showing model curve vs. actual data, and residual patterns over time, gives participants temporal-semantic context that a single MAPE number cannot.

2.6 Human-Guided Refinement with Computational Feedback

2.7 Workflow 1.8 — Human-Steered Topic Modelling (UTOPIAN)

Source: Choo et al. [4].

2.7.1 Why It Is Relevant

This workflow exemplifies a general pattern highly pertinent to the project: the human provides qualitative, semantic guidance (e.g., "these two topics should be merged" or "this keyword is important for this topic"), and the system incorporates this guidance as soft constraints while maintaining fit to the data. The result is a *semi-supervised* loop where human knowledge and computational optimisation work together.

2.7.2 What to Study in Detail

The specification artifact for human guidance. User interactions (keyword adjustments, topic merges, splits) are formalised as reference matrices with supervision weights. The system then solves a constrained optimisation that *regularises toward the user's intent* while still fitting the data. This is conceptually analogous to what happens in the forecasting scenario: the analyst's domain knowledge says "marketing spend should influence subscriptions with a ∼2-week lag," and the model should incorporate this as a soft constraint—giving that driver–transformation combination priority while still optimising overall fit.

The iterative assess–refine–recompute–re-visualise cycle. The user assesses topic quality, specifies desired changes, the system recomputes, and the updated result is visualised with animated transitions showing what changed. This cycle—with explicit visual feedback on *what changed as a result of the user's guidance*—is a design pattern the experimental interface could adopt. After a participant adds or removes a driver, or changes a transformation, the system should show not just the new MAPE but *what changed* in the model's behaviour.

The balance between human guidance and data fit. The system does not blindly follow the user; it balances user specifications with data evidence. This is an important design principle for the VA tool: the system should show participants where their domain intuition aligns with the data and where it conflicts, rather than simply accepting all selections.

2.7.3 How to Use It

Use UTOPIAN as a conceptual model for the human–system interaction design. Participants inject domain knowledge (driver selection, transformation choice); the system provides feedback on how well this knowledge aligns with the data. The richer the representational form of this feedback, the better the participant can calibrate their knowledge—which is precisely the project's experimental hypothesis.

2.8 Exploring and Comparing Multiple Models

2.9 Workflow 1.13 — Exploratory Model Analysis

Source: Cashman et al. [3].

2.9.1 Why It Is Relevant

Participants will effectively explore a space of possible models (defined by different driver selections and transformations). This workflow provides a structured approach to exploring model alternatives.

2.9.2 What to Study in Detail

The data exploration–model specification–model comparison cycle. Participants first explore data to identify potentially predictive variables, then specify a modelling problem, then compare trained models. This three-phase structure maps to the experimental task: participants explore candidate drivers, select and transform them, then evaluate the resulting model.

The concept of "analytical direction." The workflow explicitly models the analyst's evolving understanding of which variables and model types are promising. In the experimental context, this corresponds to the participant's evolving hypotheses about which drivers matter—hypotheses that are shaped by both their domain knowledge and the VA feedback they receive.

The diagnostic loop. When models are unsatisfactory, the analyst *diagnoses why* and returns to specify a different problem. In the experiment, this corresponds to participants recognising that their current driver selection yields poor MAPE and needing to understand *why* in order to improve it. The richness of the diagnostic feedback is precisely what varies across the three conditions.

2.9.3 How to Use It

Use this workflow's phase structure (explore data → specify model → evaluate → diagnose → refine) as a reference for structuring the task flow in the experiment.

2.10 Probing Feature Effects and Sensitivity

2.11 Workflow 1.17 — What-If Tool: Probing ML Models

Source: Wexler et al. [7].

2.11.1 Why It Is Relevant

The "what-if" probing philosophy aligns with a key aspect of the semantic-gap problem: the analyst wonders "what if I include this driver with a 2-week lag?" and needs feedback on the consequence.

2.11.2 What to Study in Detail

Partial dependence analysis. The tool computes how model predictions change as a single feature varies across its range. In the forecasting context, this could show participants how the target metric's predicted value changes as a candidate driver varies—effectively visualising the *marginal effect* of each driver. This is a powerful visual encoding that goes beyond a correlation number.

Feature editing with re-inference. The user edits a feature value and immediately sees the updated prediction. In the experiment, this maps to the participant adjusting a driver transformation (e.g., changing lag from one week to two weeks) and immediately seeing the impact on model predictions. This rapid feedback loop supports the iterative refinement the study aims to investigate.

Counterfactual identification. The tool finds the nearest data point with a different outcome. In the forecasting context, this could translate to showing participants examples of time periods where the target metric behaved differently from their model's prediction, and highlighting which driver values differed—providing a data-driven *explanation* for model errors.

2.11.3 How to Use It

Draw on the partial dependence and feature-editing concepts when designing the interaction mechanisms in the VA conditions, particularly for **Condition C** where semantic encoding could show the causal effect of each driver in domain-meaningful terms.

2.12 Prescriptive Feedback with Iterative Tuning

2.13 Workflow 1.5 — EventAction: Prescriptive Recommendation

Source: Du et al. [5].

2.13.1 Why It Is Relevant at a Structural Level

Although the domain is different (temporal event sequences, not forecasting), the workflow's plan-tuning loop is structurally analogous to what participants do: specify a plan (driver selection + transformations), receive feedback on estimated outcomes (predicted MAPE), assess whether the outcome is satisfactory, and refine the plan.

2.13.2 What to Study in Detail

The plan specification–outcome re-estimation–assessment–refinement cycle. Each time the user modifies the plan, the system immediately recomputes outcome probabilities and updates the display. This tight feedback loop is the core interaction pattern the experimental interface needs to support.

The outcome visualisation design. The system overlays updated outcome probabilities on the original distribution, showing *the impact of the user's choices* as a visual delta. In the experiment, this could correspond to showing the MAPE improvement (or degradation) resulting from each driver selection or transformation change, overlaid on the baseline.

2.13.3 How to Use It

Adopt the tight plan → feedback → refine loop as the core interaction pattern across all three experimental conditions.

2.14 Summary: Mapping Workflows to Project Components

Table 1 summarises the recommended workflows for each component of the project.

Table 1: Mapping from project components to relevant library workflows.

Project Component	**Workflows**	**What to Look For**
Core analytical task (iterative feature selection for regression)	**1.10**, **1.12**	Feature relevance ranking; conditional distribution displays; feature engineering loop; feature assessment [6, 2]
Time-series context (temporal driver–target relationships)	**1.11**	Visual identification of temporal characteristics; model-curve-vs-data overlay; residual displays over time [1]
Human–system interaction design (knowledge injection + computational feedback)	**1.8**, **1.12**	Semi-supervised refinement loop; specification artifacts for human guidance; coordinated multi-view assessment [4, 2]
Model exploration and comparison	**1.13**	Data exploration → model specification → evaluation → diagnosis cycle; analytical direction concept [3]
Feature-effect probing (what-if reasoning about drivers)	**1.17**	Partial dependence analysis; feature editing with re-inference; counterfactual identification [7]
Tight feedback-loop design (plan → outcome → refine)	**1.5**	Plan specification → outcome re-estimation → visual impact display → refinement cycle [5]

2.15 Recommendations for Designing the Three Conditions

Based on the workflows above, the three experimental conditions can be framed as follows in terms of what the library teaches.

Condition A — Tabular/numerical statistical feedback. Participants receive the same information that Workflow 1.10 computes (feature relevance scores, model accuracy metrics) but presented only as numbers and tables. This is the "raw" output of the characterise transforms without visualisation.

Condition B — Visual statistical encoding. Participants see the ranked small-multiple conditional-distribution plots from Workflow 1.10 [6] and the model-curve-versus-data overlays from Workflow 1.11 [1]. These visualisations show the *shape* of relationships and the *temporal structure* of model fit, but without domain-semantic framing. The information is identical to Condition A but the representational form makes patterns (non-linearity, lag effects, interactions) perceptually accessible.

Condition C — Semantic visual encoding. Participants see the Condition B visualisations *plus* domain-meaningful framing inspired by Workflow 1.12's coordinated views [2] and Workflow 1.17's partial-dependence displays [7]: driver–target relationships are shown with explicit causal annotations (e.g., labelled lag arrows, named business-cycle effects), and the partial effect of each driver is displayed in domain terms (e.g., "each additional £1k in marketing spend is associated with $+X$ subscriptions after Y weeks"). This adds the semantic bridge that Workflows 1.8 [4] and 1.12 [2] achieve through domain-contextual views.

This interpretation allows testing the core hypothesis—that the *representational form*, not the *information quantity*, drives the closure of the semantic gap—because all three conditions present the same computed statistics, just in progressively richer representational forms.

2.16 An Additional Suggestion

Consider also studying the **structure of the iterative loops** across the recommended workflows. All of them [6, 1, 2, 4] share a common pattern: *assess* → *diagnose* → *specify refinement* → *recompute* → *re-visualise*. The explicit **assessment** step, where the analyst forms a quality judgment, is the moment where the semantic gap is either bridged or not.

The project's qualitative data collection (inline justifications at the moment of each feature-selection decision) effectively captures this assessment step. The workflow library suggests that the quality of this assessment—and hence the quality of subsequent refinement decisions—depends critically on the representational form of the preceding visualisation. This framing could strengthen the theoretical argument of the dissertation.

References

[1] Natalia Andrienko and Gennady Andrienko. A visual analytics framework for spatio-temporal analysis and modelling. *Data Mining and Knowledge Discovery*, 27(1):55–83, Jul 2013.

[2] Natalia Andrienko, Gennady Andrienko, Alexander Artikis, Periklis Mantenoglou, and Salvatore Rinzivillo. Human-in-the-loop: Visual analytics for building models recognizing behavioral patterns in time series. *IEEE Computer Graphics and Applications*, 44(3):14–29, 2024.

[3] Dylan Cashman, Shah Rukh Humayoun, Florian Heimerl, Kendall Park, Subhajit Das, John Thompson, Bahador Saket, Abigail Mosca, John Stasko, Alex Endert, Michael Gleicher, and Remco Chang. A user-based visual analytics workflow for exploratory model analysis. *Computer Graphics Forum*, 38(3):185–199, 2019.

[4] Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013.

[5] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. Eventaction: Visual analytics for temporal event sequence recommendation. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 61–70, 2016.

[6] Thomas Mühlbacher and Harald Piringer. A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1962–1971, 2013.

[7] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda B. Viégas, and Jimbo Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 2020.