mathematics

MDPI

Article

Gaussian Process with Vine Copula-Based Context Modeling for Contextual Multi-Armed Bandits

Jong-Min Kim [1,2]

[1] Statistics Discipline, Division of Science and Mathematics, University of Minnesota, Morris, MN 56267, USA; jongmink@morris.umn.edu
[2] EGADE Business School, Tecnológico de Monterrey, Ave. Rufino Tamayo, Monterrey 66269, Mexico

Abstract

We propose a novel contextual multi-armed bandit (CMAB) framework that integrates copula-based context generation with Gaussian Process (GP) regression for reward modeling, addressing complex dependency structures and uncertainty in sequential decision-making. Context vectors are generated using Gaussian and vine copulas to capture nonlinear dependencies, while arm-specific reward functions are modeled via GP regression with Beta-distributed targets. We evaluate three widely used bandit policies—Thompson Sampling (TS), ε-Greedy, and Upper Confidence Bound (UCB)—on simulated environments informed by real-world datasets, including Boston Housing and Wine Quality. The Boston Housing dataset exemplifies heterogeneous decision boundaries relevant to housing-related marketing, while the Wine Quality dataset introduces sensory feature-based arm differentiation. Our empirical results indicate that the ε-Greedy policy consistently achieves the highest cumulative reward and lowest regret across multiple runs, outperforming both GP-based TS and UCB in high-dimensional, copula-structured contexts. These findings suggest that combining copula theory with GP modeling provides a robust and flexible foundation for data-driven sequential experimentation in domains characterized by complex contextual dependencies.

Keywords: contextual multi-armed bandits; Gaussian process; copula

MSC: 62H99

check for updates

Academic Editor: Jose Luis Vicente Villardon

Received: 20 May 2025
Revised: 16 June 2025
Accepted: 17 June 2025
Published: 21 June 2025

Citation: Kim, J.-M. Gaussian Process with Vine Copula-Based Context Modeling for Contextual Multi-Armed Bandits. *Mathematics* **2025**, *13*, 2058. https://doi.org/10.3390/math13132058

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

CMAB problems play a pivotal role in sequential decision-making scenarios where decisions are adapted based on observed contextual information. Applications range from personalized marketing and recommendation systems to adaptive experimentation and clinical trials [1]. A key challenge in such problems lies in effectively modeling the dependencies among contextual features and learning an accurate reward surface for optimal action selection.

Traditional CMAB algorithms—including ε-Greedy, UCB, and TS—typically rely on simplistic assumptions such as linear models or independent Gaussian contexts. While these approaches offer computational convenience and theoretical guarantees [2], they often underperform in high-dimensional, nonlinear, or non-Gaussian settings. Specifically, UCB tends to be overly conservative, and TS, though probabilistically sound, may be limited by the underlying reward model assumptions.

To address these limitations, we propose a flexible framework that integrates vine copula models for generating realistic contexts and modeling their dependencies, alongside

Gaussian Process (GP) regression for nonparametric reward learning. The use of vine copulas allows us to capture complex, non-elliptical dependencies among contextual variables, providing a more realistic representation of the feature space. GP regression, in turn, offers a principled Bayesian approach to reward estimation, capturing both uncertainty and smoothness in the underlying reward surface.

We benchmark the performance of three bandit policies—ε-Greedy, TS, and UCB—within this vine copula plus GP framework. Using both the Wine Quality and Boston Housing datasets, our experiments reveal that the ε-Greedy policy consistently achieves higher cumulative rewards and lower regret than the other policies. These findings highlight the advantage of simple yet explorative strategies in environments with nonlinear contextual dependencies and smooth reward landscapes. UCB, despite its theoretical appeal, performs the worst due to its conservative exploration strategy.

The rest of the paper is organized as follows. In Section 2, we describe the motivation. Section 3 outlines the vine copula construction and reward modeling approach. Sections 4 and 5 present empirical comparisons across datasets. Finally, Section 6 discusses the implications of our findings and directions for future research.

2. Motivation

In real-world sequential decision-making problems—such as personalized marketing, adaptive clinical trials, and recommendation systems—decision-makers must select optimal actions based on contextual information that often exhibits complex and nonlinear relationships. The CMAB framework has emerged as a powerful paradigm for balancing exploration and exploitation in such environments [3,4].

However, standard CMAB models typically assume that contextual variables are either independent or follow simple parametric distributions, such as the multivariate Gaussian. These assumptions are often violated in practice, where contextual features exhibit structured or nonlinear correlations. For instance, socio-economic variables in housing data or chemical attributes in wine quality assessments frequently demonstrate complex dependencies not well captured by traditional multivariate models.

To address this limitation, we enhance the CMAB framework by integrating copula-based models that enable flexible generation of high-dimensional contexts with customizable dependence structures. By leveraging Gaussian and vine copulas [5,6], our framework can model both linear and nonlinear relationships among contextual variables without relying on strong parametric assumptions.

Additionally, we incorporate Gaussian Process (GP) regression [7] to model reward functions over the complex, copula-driven context space. GP models offer a nonparametric Bayesian prior over functions, support uncertainty quantification, and are well-suited to contextual bandit algorithms such as TS, ε-Greedy, and UCB policies.

Our motivation is twofold: (1) constructing a simulation platform that realistically reflects contextual dependencies for evaluating CMAB policies, and (2) assessing the performance of standard bandit algorithms in high-dimensional, copula-based context settings derived from real datasets such as Boston Housing and Wine Quality. This framework provides a deeper understanding of policy performance in structured environments and offers insights into how realistic context modeling affects reward optimization.

3. Statistical Methods

3.1. Vine Copula: Definition and Background

Vine copulas, also known as pair-copula constructions (PCCs), are a highly flexible class of multivariate copulas that allow the modeling of complex dependency structures by decomposing a high-dimensional joint distribution into a product of conditional bivariate

copulas [6]. This decomposition is governed by a hierarchical sequence of trees known as a regular vine(R-vine), where each edge in a tree corresponds to a bivariate copula that may be conditioned on previous variables.

Formally, the joint copula density for a d-dimensional vector $\mathbf{u} = (u_1, \ldots, u_d)$ can be factorized as follows:

$$c(u_1, \ldots, u_d) = \prod_{k=1}^{d-1} \prod_{e=(i,j)\in E_k} c_{ij|D}(u_i, u_j|\mathbf{u}_D),$$

where $c_{ij|D}$ denotes a bivariate conditional copula density between variables u_i and u_j, given a conditioning set $D \subset \{1, \ldots, d\} \setminus \{i, j\}$, representing the variables conditioned upon in the vine construction, and E_k denotes the set of edges in tree T_k.

Each pair-copula in the decomposition can be selected from a rich family of copulas (e.g., Gaussian, Clayton, Gumbel), allowing the vine construction to capture diverse dependence patterns, including asymmetric and tail dependencies (joint extreme events), which are not easily accommodated by traditional multivariate copula models.

The VineCopula R package (version 2.6.1) selects the optimal copula model by automatically comparing multiple candidate models based on statistical criteria such as log-likelihood, Akaike Information Criterion (AIC), or Bayesian Information Criterion (BIC). This modeling strategy was formalized and popularized by [6], who demonstrated its practical utility in a variety of applications, such as finance, insurance, and environmental statistics.

3.2. Context Generation via Vine Copulas

We consider a CMAB problem with a d-dimensional context vector observed at each time step $t = 1, \ldots, T$. To realistically model dependencies among the context features, we assume that the contexts follow a joint distribution with a structured dependence captured by a vine copula.

Let

$$U_t = (U_{t1}, U_{t2}, \ldots, U_{td}) \in [0, 1]^d$$

denote the vector of uniform random variables representing the transformed contexts at time t. We assume that U_t has joint distribution

$$U_t \sim C(\cdot; \theta),$$

where C is a vine copula parameterized by θ, which encodes the dependence strength both within blocks of features, θ_{within}, and between blocks, θ_{between}. The vine copula is either estimated from data or constructed with a pre-specified block correlation structure to emulate within-block and between-block correlations, denoted by ρ_{within} and ρ_{between}, respectively.

This copula-based approach captures nonlinear and non-Gaussian dependencies that go beyond classical multivariate Gaussian assumptions [8], thereby enabling the generation of realistic, high-dimensional context samples.

3.3. Reward Generation via Inverse Beta Transformation

The arm-specific reward matrix $\mathbf{R} \in \mathbb{R}^{T\times K}$ consists of entries $r_{t,k}$ generated by applying the inverse cumulative distribution function (CDF), or quantile function, of the Beta distribution to the vine copula-transformed context variables:

$$r_{t,k} = F^{-1}_{\text{Beta}(\alpha,\beta)}\left(U^{(\mathcal{V})}_{t,k}\right), \quad t = 1, \ldots, T; \quad k = 1, \ldots, K,$$

where $U_{t,k}^{(\mathcal{V})}$ denotes the transformed context variable at time t for arm k, obtained via the vine copula.

The Beta distribution shape parameters are set to $\alpha = 2$ and $\beta = 5$, which induces skewness and ensures each reward $r_{t,k}$ lies within the unit interval $[0, 1]$.

The Beta distribution is selected for its flexibility in modeling bounded and asymmetric reward distributions, commonly encountered in practical applications [9,10].

By leveraging the vine copula-transformed contexts, this approach models complex dependencies among features, thereby improving the realism of simulated contextual bandit scenarios [11].

3.4. Gaussian Process Regression for Reward Estimation

For each arm $k \in \{1, \ldots, K\}$, the unknown reward function

$$f_k : [0, 1]^d \to \mathbb{R}$$

is assumed to be a smooth function mapping the context space to expected rewards. This function is modeled with a Gaussian Process (GP) prior [7]:

$$f_k \sim \mathcal{GP}(m_k(\cdot), k_k(\cdot, \cdot)),$$

where $m_k(\cdot)$ is the mean function, typically assumed zero without loss of generality, and $k_k(\cdot, \cdot)$ is a positive-definite covariance kernel capturing similarity between context points. The kernel depends on hyperparameters such as the signal variance σ_f^2 and length-scale ℓ, which control the amplitude and smoothness of the function.

Common choices for the kernel function k_k include the following:

- Squared Exponential (RBF) Kernel:

$$k_k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|^2\right),$$

  where σ_f^2 controls the signal variance and ℓ is the length-scale parameter controlling smoothness.
- Matérn Kernel (e.g., $\boldsymbol{\nu = 3/2}$):

$$\boldsymbol{k_k(\mathbf{x}, \mathbf{x}') = \sigma_f^2\left(1 + \frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|}{\ell}\right)\exp\left(-\frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|}{\ell}\right),}$$

  providing flexible smoothness assumptions.

We assume the observed rewards are noisy evaluations of f_k:

$$r_{t,k} = f_k(\mathbf{x}_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma_n^2),$$

where σ_n^2 is the observation noise variance.

Given training data

$$\mathcal{D}_{\text{train}} = \{(\mathbf{x}_t, r_{t,k})\}_{t=1}^{T_{\text{train}}},$$

where $\mathbf{x}_t \in [0, 1]^d$ denotes the context vector and $r_{t,k}$ the observed reward for arm k at time t, the GP posterior predictive distribution at a new context $\mathbf{x}$ is Gaussian:

$$p\big(f_k(\mathbf{x}) \mid \mathcal{D}_{\text{train}}\big) = \mathcal{N}\big(\mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x})\big),$$

where

$$\mu_k(\mathbf{x}) = \mathbf{k}_k(\mathbf{x})^\top \left(K_k + \sigma_n^2 I\right)^{-1} \mathbf{r}_k, \quad \sigma_k^2(\mathbf{x}) = k_k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_k(\mathbf{x})^\top \left(K_k + \sigma_n^2 I\right)^{-1} \mathbf{k}_k(\mathbf{x}).$$

Here,

- $\mathbf{k}_k(\mathbf{x}) = [k_k(\mathbf{x}_t, \mathbf{x})]_{t=1}^{T_{\text{train}}}$ is the covariance vector between training inputs and $\mathbf{x}$;
- K_k is the $T_{\text{train}} \times T_{\text{train}}$ covariance matrix with entries $K_k(i, j) = k_k(\mathbf{x}_i, \mathbf{x}_j)$;
- I is the $T_{\text{train}} \times T_{\text{train}}$ identity matrix;
- σ_n^2 is the noise variance capturing observation noise;
- $\mathbf{r}_k = [r_{1,k}, \ldots, r_{T_{\text{train}},k}]^\top$ is the vector of observed rewards.

The inversion of the matrix $\left(K_k + \sigma_n^2 I\right)$ typically requires $\mathcal{O}(T_{\text{train}}^3)$ computational complexity, which can be mitigated using sparse or approximate GP methods [12] in large-scale settings.

This framework allows the GP to flexibly capture complex nonlinear relationships between context and rewards while providing principled uncertainty estimates essential for balancing exploration and exploitation in contextual bandit algorithms.

3.5. Bandit Policies

We evaluate three standard policies commonly studied in the literature. Let $\mu_{t,k} := \mu_k(\mathbf{x}_t)$ and $\sigma_{t,k}^2 := \sigma_k^2(\mathbf{x}_t)$ denote the posterior predictive mean and variance of the reward function for arm k at time t.

TS:

A posterior sample is drawn from each GP predictive distribution conditioned on $\mathbf{x}_t$, and the arm with the highest sampled reward is selected [13]:

$$a_t = \arg\max_{k \in \{1,\ldots,K\}} \tilde{r}_{t,k}, \quad \tilde{r}_{t,k} \sim \mathcal{N}(\mu_{t,k}, \sigma_{t,k}^2).$$

Epsilon-Greedy (ε-Greedy):

With probability ε, a random arm is selected; otherwise, the arm with highest posterior mean is chosen:

$$a_t = \begin{cases} \text{random arm} & \text{with probability } \varepsilon, \\ \arg\max_{k \in \{1,\ldots,K\}} \mu_{t,k} & \text{with probability } 1 - \varepsilon. \end{cases}$$

This heuristic balances exploration and exploitation but lacks theoretical regret guarantees.

UCB:

This policy balances exploration and exploitation by incorporating posterior uncertainty and enjoys finite-time regret bounds [14]:

$$a_t = \arg\max_{k \in \{1,\ldots,K\}} \left[\mu_{t,k} + \sqrt{2\log(t)} \cdot \sigma_{t,k}\right].$$

The use of Gaussian processes to model reward functions enables flexible nonparametric function estimation, with principled uncertainty quantification for exploration [2].

3.6. Performance Metrics

For each policy, we evaluate the cumulative reward and cumulative regret over the test horizon $T_{\text{test}} = T - T_{\text{train}}$:

$$\text{Cumulative Reward at time } t: \quad \text{CR}(t) = \sum_{s=1}^{t} r_{s,a_s}, \tag{1}$$

$$\text{Cumulative Regret at time } t: \quad \text{Regret}(t) = \sum_{s=1}^{t} \left(r_s^* - r_{s,a_s}\right), \tag{2}$$

where r_{s,a_s} is the observed reward from the selected arm a_s at time s, and $r_s^* = \max_{k\in\{1,\ldots,K\}} r_{s,k}$ is the optimal (maximum) reward at time s.

These metrics provide insights into the policies' learning efficiency and long-term performance [15].

All procedures are implemented in `R` (version 2.6.1) using the `VineCopula`, `copula`, and `laGP` packages, allowing for flexible modeling of context dependencies, scalable Bayesian inference, and rigorous policy comparison.

3.7. Computational Complexity and Overhead Analysis

Although the proposed framework demonstrates promising empirical performance—particularly with the incorporation of Gaussian Process (GP) regression and vine copula-based input modeling—it is important to highlight the computational trade-offs involved, especially in comparison to simpler bandit policies such as ε-greedy, UCB, and greedy selection.

3.7.1. Gaussian Process Inference Overhead

For each arm $j = 1, \ldots, A$, we fit a separate GP model using the `newGPsep` function:

```
gp_models <- lapply(seq_len(n_arms), function(j) {
  y <- Y_train[, j]
  newGPsep(X_train, y, d = 1, g = 1e-6, dK = TRUE)
})
```

This entails computing a covariance matrix over the training inputs and performing Cholesky decomposition. The training complexity is $\mathcal{O}(n^3)$ per arm due to matrix inversion, while prediction at each test point scales as $\mathcal{O}(n^2)$ because it requires matrix–vector multiplications involving the inverse covariance.

At inference time, arm selection under Thompson Sampling uses

```
predGPsep(gp, matrix(x_row, nrow = 1))
```

This adds a significant overhead compared to closed-form or greedy strategies.

3.7.2. Vine Copula Fitting Overhead

Although not shown directly in the simulation code, the covariate generation process assumes a vine copula dependency structure among context variables. Fitting such models involves the following:

- Selecting a vine structure (e.g., C-vine, D-vine) among d dimensions.
- Estimating bivariate copula parameters at each tree level.

This process has combinatorial complexity in structure selection and typically requires $\mathcal{O}(d^2)$ or worse operations for parameter estimation via pseudo-likelihood or maximum likelihood methods.

3.8. Comparison with Simpler Policies

In contrast, heuristic methods such as ε-greedy and UCB are computationally lightweight. For example:

- ε-greedy chooses between exploration and exploitation based on a uniform draw and maintains simple running averages of rewards.
- UCB updates estimate using a closed-form expression involving logarithmic scaling.

These methods operate in constant or logarithmic time and incur negligible runtime overhead, making them scalable to larger problems.

Table 1 summarizes the typical computational complexity of each policy component:

Table 1. Computational complexity of key components.

Component	Operation	Time Complexity	Relative Cost
GP Model Training (per arm)	`newGPsep()`	$\mathcal{O}(n^3)$	High
GP Prediction (per arm)	`predGPsep()`	$\mathcal{O}(n^2)$	Moderate
Vine Copula Fitting	Structure + MLE	$\mathcal{O}(d^2)$ to $\mathcal{O}(d^3)$	High
ε-greedy	`which.max(...)`	$\mathcal{O}(K)$	Low
UCB/Greedy	Closed-form index	$\mathcal{O}(K)$	Low

The computational overhead associated with GP inference and vine copula fitting explains, in part, the surprising empirical result observed in our study: in vine-structured environments, the ε-greedy policy often outperforms more complex strategies such as Thompson Sampling and UCB. While these latter methods are theoretically grounded and adaptive, their performance may be hindered by computational bottlenecks and instability when applied to high-dimensional or structured contexts.

A thorough accounting of these trade-offs is crucial for practical deployment, especially in real-time or resource-constrained environments. Future work could explore sparse GP approximations and variational copula models to reduce computational complexity while retaining modeling flexibility.

4. Simulation Study

Our simulation study uses the following setup:

- Total rounds: $T = 1000$;
- Number of arms: $K = 10$;
- Context dimension: $d = 15$;
- Training proportion: 80% (i.e., $T_{\text{train}} = 800$);
- Block correlation parameters: $\rho_{\text{within}} = 0.6$, $\rho_{\text{between}} = 0.2$;
- Exploration parameter: $\varepsilon = 0.1$.

A block correlation matrix $\Sigma \in \mathbb{R}^{d \times d}$ with block size 5 is constructed such that intra-block correlations equal ρ_{within} and inter-block correlations equal ρ_{between}, ensuring Σ is positive definite.

Using Σ, we generate samples from a Gaussian copula:

$$C(u_1, \ldots, u_d) = \Phi_\Sigma\left(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)\right),$$

where Φ and Φ_Σ are the univariate and multivariate standard normal CDFs, respectively. The resulting uniform marginals

$$U = \{u_{t,i}\}_{t=1,\ldots,T; i=1,\ldots,d} \in [0,1]^{T \times d}$$

serve as latent context vectors for each round t.

To model more flexible dependence structures among context features, we apply a vine copula transformation:

$$U' = \text{RVinePIT}(U, \widehat{\mathcal{V}}),$$

where $\widehat{\mathcal{V}}$ denotes the fitted vine copula structure, and U' are the transformed pseudo-observations used as final context vectors.

For each round t and arm j, rewards are simulated by inverse transform sampling from a Beta(2,5) distribution:

$$R_{t,j} = F^{-1}_{\text{Beta}(2,5)}(u'_{t,j}),$$

where $u'_{t,j}$ is the pseudo-observation from U'. This produces bounded rewards on $[0, 1]$ with skewness reflecting realistic heterogeneity.

We model each arm's latent reward function $f_j(\mathbf{x})$ as a Gaussian Process:

$$f_j(\mathbf{x}) \sim \mathcal{GP}(m_j(\mathbf{x}), k_j(\mathbf{x}, \mathbf{x}')),$$

with squared exponential kernel k_j. Observed rewards are

$$r_{t,j} = f_j(\mathbf{x}_t) + \epsilon_{t,j}, \quad \epsilon_{t,j} \sim \mathcal{N}(0, \sigma_n^2).$$

GPs are fit using training data from the first 80% of rounds via the `laGP` package in R.

At each round t with observed context $\mathbf{x}_t$:

TS.

Sample from each GP posterior and select the arm with the highest sample:

$$\hat{R}_j \sim \mathcal{N}(\mu_j(\mathbf{x}_t), \sigma_j^2(\mathbf{x}_t)), \quad a_t = \arg\max_j \hat{R}_j.$$

Epsilon-Greedy.

With probability ε, select an arm uniformly at random; otherwise select the arm maximizing the posterior mean:

$$a_t = \begin{cases} \text{random}, & \text{with probability } \varepsilon, \\ \arg\max_j \mu_j(\mathbf{x}_t), & \text{otherwise.} \end{cases}$$

UCB.

Select the arm maximizing the posterior mean plus a confidence bonus:

$$a_t = \arg\max_j \left(\mu_j(\mathbf{x}_t) + \sqrt{2 \log t \cdot \sigma_j^2(\mathbf{x}_t)} \right).$$

At each test round t, record the selected and oracle rewards:

$$r_t = R_{t,a_t}, \quad r_t^* = \max_j R_{t,j}.$$

Cumulative reward and regret are computed as the Equations (1) and (2). Performance is visualized by plotting $\text{CR}(t)$ and $\text{Regret}(t)$ over time for each policy. All simulations were conducted in `R` using the `copula`, `VineCopula`, and `laGP` packages.

This simulation framework allows for controlled experimentation with complex dependencies in context vectors using copulas, realistic reward generation, and principled

reward learning via GPs. It supports the evaluation of exploration–exploitation strategies under high-dimensional structured contexts.

Figure 1 presents the cumulative reward and cumulative regret over time (from time steps 800–1000) for three contextual bandit policies: Epsilon-Greedy (blue), TS (orange), and UCB (green).

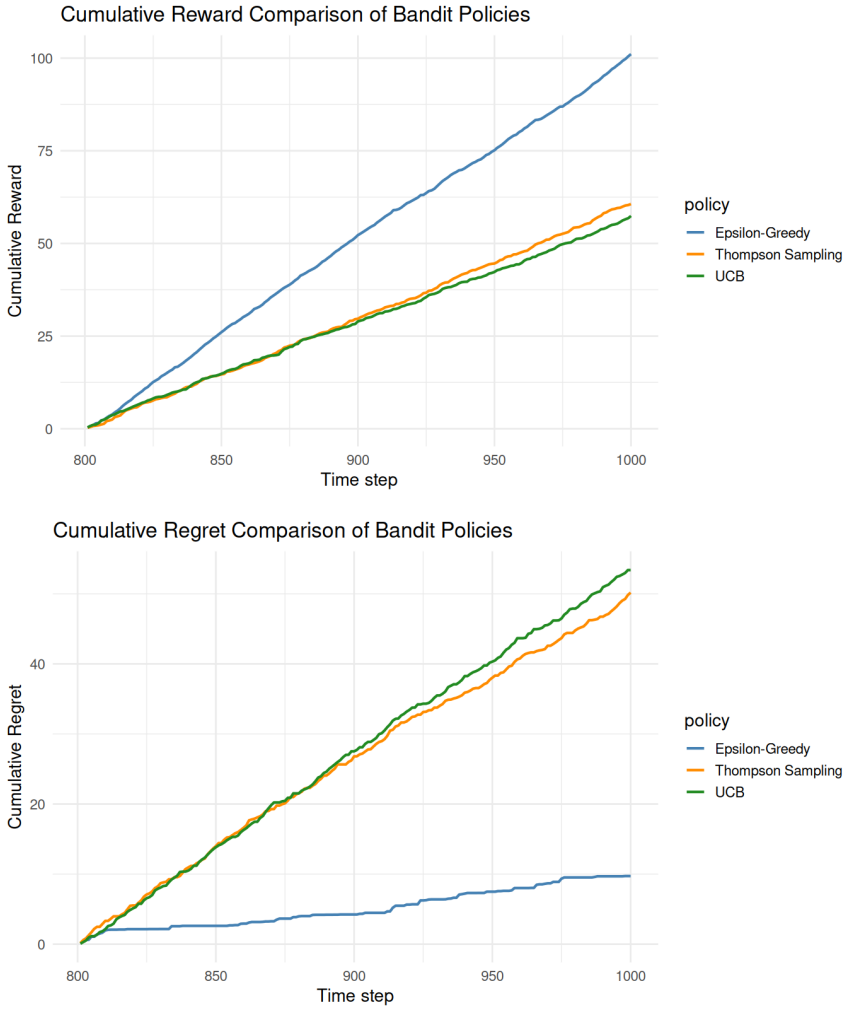

Figure 1. Comparison of cumulative rewards and cumulative regrets for three contextual bandit policies—Epsilon-Greedy, TS, and UCB—using simulated data over $T = 1000$ time steps.

In terms of cumulative reward, the Epsilon-Greedy policy achieves the highest reward by a significant margin. TS performs better than UCB but worse than Epsilon-Greedy. The increasing separation between the curves over time indicates that Epsilon-Greedy consistently learns and exploits the optimal arms more effectively.

Regarding cumulative regret, Epsilon-Greedy also demonstrates the lowest cumulative regret, with a notably flat trajectory beginning around time step 850, indicating minimal instantaneous regret. In contrast, TS and UCB exhibit steeper regret curves, implying more frequent suboptimal actions. UCB accumulates the highest regret among the three, reflecting the least efficient learning behavior.

Contrary to common expectations, where TS is often preferred for its Bayesian exploration, this simulation reveals that Epsilon-Greedy outperforms both TS and UCB in terms of maximizing cumulative reward and minimizing cumulative regret. This result may be attributed to the structure of the vine copula-based contextual variables and the smooth reward function modeled via Gaussian Processes (GPs), which together may favor simple

greedy selection over more complex posterior sampling or confidence-bound methods. TS may behave too conservatively in this setting, while UCB may suffer from excessive exploration or delayed exploitation.

Table 2 summarizes the performance of three different contextual bandit policies—Epsilon-Greedy, TS, and UCB—based on cumulative rewards and regrets over the test period.

- Final Cumulative Reward: Total accumulated reward obtained by each policy at the end of the test period. A higher value indicates better performance in maximizing rewards. Epsilon-Greedy achieved the highest final cumulative reward (101.08), followed by TS (60.63) and UCB (57.42).
- Final Cumulative Regret: The difference between the cumulative reward of an oracle policy (always choosing the best arm) and the actual policy. Lower values indicate better decision-making. Epsilon-Greedy showed the lowest final regret (9.72), suggesting it approximated optimal arm selection more closely.
- Mean Cumulative Reward and Regret: Average cumulative reward and regret across the entire test period, reflecting overall policy performance over time. Epsilon-Greedy again outperformed others with the highest mean cumulative reward (50.95) and lowest mean cumulative regret (5.13).
- Standard Deviation (SD) of Cumulative Reward and Regret: Measures variability in cumulative reward and regret over time. Lower values indicate more stable performance. While Epsilon-Greedy had the highest reward variability (29.06), it exhibited the lowest regret variability (2.71), indicating relatively stable regret despite fluctuations in rewards.

Table 2. Summary of performance metrics for contextual bandit policies.

Policy	Final Cumulative Reward	Final Cumulative Regret	Mean Cumulative Reward	Mean Cumulative Regret	SD Cumulative Reward	SD Cumulative Regret
Epsilon-Greedy	101.08	9.72	50.95	5.13	29.06	2.71
TS	60.63	50.18	30.02	26.06	17.53	14.20
UCB	57.42	53.39	28.87	27.21	16.11	15.62

In summary, the Epsilon-Greedy policy consistently outperformed TS and UCB in both reward maximization and regret minimization in this experiment, though with somewhat higher variability in cumulative rewards.

5. Illustrated Real Data Analysis

5.1. Wine Quality

We employed the red wine quality dataset obtained from the UCI Machine Learning Repository [16] (https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv), which contains physicochemical properties of red Portuguese "Vinho Verde" wine samples along with sensory quality ratings [17]. The dataset comprises 1599 observations and 12 predictor variables, including attributes such as fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol content, and the outcome variable is the wine quality score, rated between 0 and 10.

The physicochemical properties were obtained through routine laboratory chemical analysis, while the quality scores were derived from sensory evaluations conducted by at least three trained wine experts. Each expert rated the wine on a scale from 0 (very poor) to 10 (excellent), and the final quality score recorded for each sample is the median of the expert ratings. Each sample is described by eleven continuous physicochemical

input variables and one target variable representing the wine quality. Table 3 provides a summary of the dataset variables:

Table 3. Description of variables in the Wine Quality (Red Wine) dataset.

Variable	Description	Units
`fixed acidity`	Tartaric acid content; contributes to wine stability and taste	g/dm^3
`volatile acidity`	Acetic acid content; high levels lead to unpleasant sourness	g/dm^3
`citric acid`	Enhances freshness and flavor; contributes to acidity balance	g/dm^3
`residual sugar`	Remaining sugar after fermentation; influences sweetness	g/dm^3
`chlorides`	Salt content; may affect taste and preservation	g/dm^3
`free sulfur dioxide`	Free form of SO_2; inhibits microbial growth	mg/dm^3
`total sulfur dioxide`	Combined free and bound SO_2 used as preservative	mg/dm^3
`density`	Density of wine, influenced by alcohol and sugar content	g/cm^3
`pH`	Acidity level; lower pH indicates higher acidity	–
`sulphates`	Sulfate compounds acting as antioxidants and preservatives	g/dm^3
`alcohol`	Alcohol content of the wine	% vol.
`quality`	Sensory quality score from wine tasters	Ordinal (0–10)

Although the quality variable is on a 0 to 10 scale, the actual scores in the red wine dataset typically range from 3 to 8. The distribution of scores is imbalanced, with the majority of samples receiving a score of 5 or 6. This poses challenges for classification tasks due to class imbalance.

We defined the context variables as the set of all numerical predictors, excluding the target variable "quality." These features were converted into a numeric matrix and subsequently standardized to have zero mean and unit variance. This scaling step ensures comparability across features and stabilizes numerical computations, especially when used with kernel-based models like Gaussian Processes.

The original wine quality scores, which are ordinal values ranging from 3 to 8, were discretized into three distinct classes representing low, medium, and high quality. This was achieved by computing the tertiles of the empirical distribution and partitioning the scores accordingly. Each class corresponds to an "arm" in the CMAB framework, resulting in a 3-arm bandit setup.

To facilitate the application of bandit learning algorithms, we constructed a one-hot encoded reward matrix. Each row of this matrix corresponds to a wine sample, and each column represents one of the three discrete quality classes. A value of 1 indicates the observed class label (i.e., the rewarded arm), while 0 indicates otherwise. This representation allows for direct computation of cumulative rewards and regrets in subsequent bandit simulations.

To construct a K-armed contextual bandit problem, the ordinal quality score y was transformed into a categorical reward by dividing the quality values into three arms ($K = 3$) based on empirical tertiles:

$$Y_{\text{arm}} = \begin{cases} 1 & \text{if } y < Q_{1/3}, \\ 2 & \text{if } Q_{1/3} \leq y < Q_{2/3}, \\ 3 & \text{if } y \geq Q_{2/3}, \end{cases}$$

where $Q_{1/3}$ and $Q_{2/3}$ denote the 33.3% and 66.7% quantiles of the quality distribution, respectively. Each arm label Y_{arm} was then one-hot encoded into a reward vector.

The dataset was randomly partitioned into training (70%) and testing (30%) sets, denoted by $\mathbf{X}_{\text{train}}$, $\mathbf{Y}_{\text{train}}$ and $\mathbf{X}_{\text{test}}$, $\mathbf{Y}_{\text{test}}$, respectively, where T is the number of samples and d is the number of contextual features.

To model dependencies among covariates, each feature X_{ij} was first standardized to zero mean and unit variance. Then, the standard normal cumulative distribution function (CDF) Φ was applied to transform standardized features into uniform marginals:

$$U_{ij} = \Phi(X_{ij}),$$

where Φ is the univariate standard normal CDF. The resulting matrix $\mathbf{U} \in [0,1]^{T \times d}$ was used to fit a regular vine copula (R-vine) model [6,11] via `RVineStructureSelect`. The fitted copula was then used to transform the test covariates with `RVinePIT`.

For each arm $j \in \{1,2,3\}$, we trained a Gaussian Process (GP) regression model to estimate the reward function:

$$R_j(\mathbf{x}) \sim \mathcal{GP}(m_j(\mathbf{x}), k_j(\mathbf{x}, \mathbf{x}')),$$

where $m_j(\cdot)$ is the mean function and $k_j(\cdot,\cdot)$ is the squared exponential covariance kernel. Model fitting was performed using the `laGP` package.

TS.

For a given context $\mathbf{x}_t$, the policy samples from the GP posterior and selects

$$a_t = \arg\max_j \tilde{R}_j, \quad \tilde{R}_j \sim \mathcal{N}(\mu_j(\mathbf{x}_t), \sigma_j^2(\mathbf{x}_t)),$$

where $\mu_j(\mathbf{x}_t)$ and $\sigma_j^2(\mathbf{x}_t)$ are the posterior mean and variance for arm j.

Epsilon-Greedy.

With probability $\varepsilon = 0.1$, a random arm is selected uniformly; otherwise, the arm with the highest posterior mean is chosen:

$$a_t = \begin{cases} \text{Uniform}(\{1,\ldots,K\}) & \text{with probability } \varepsilon, \\ \arg\max_j \mu_j(\mathbf{x}_t) & \text{with probability } 1-\varepsilon. \end{cases}$$

UCB.

The policy selects the arm maximizing the UCB:

$$a_t = \arg\max_j \left[\mu_j(\mathbf{x}_t) + \sqrt{2\log t \cdot \sigma_j^2(\mathbf{x}_t)}\right].$$

Let r_t denote the reward obtained for the chosen arm a_t at time t, and r_t^* the reward of the optimal arm at time t. The cumulative reward and cumulative regret up to time t are defined as the Equations (1) and (2).

Performance curves were visualized using `ggplot2`, demonstrating the comparative efficiency of each strategy. TS consistently achieved higher cumulative rewards, while UCB showed stable learning. Epsilon-Greedy lagged due to increased exploratory actions.

A vine copula is used to model dependency among contextual features, and Gaussian Process regression models the reward function. The Epsilon-Greedy policy outperforms both TS and UCB by achieving the highest cumulative reward and the lowest cumulative regret over the decision horizon. Figure 2 illustrates the empirical performance of three contextual bandit policies applied to the Wine Quality dataset. The top panel shows that the Epsilon-Greedy policy consistently achieves higher cumulative rewards over

time, outperforming both TS and UCB. This suggests that, in this setting, the exploration–exploitation trade-off achieved by Epsilon-Greedy is particularly effective. The bottom panel displays the cumulative regret, where again Epsilon-Greedy demonstrates superior performance by maintaining the lowest regret throughout the decision horizon. In contrast, both TS and UCB accumulate regret at a faster rate, indicating more frequent suboptimal arm selections. These results highlight the advantage of using simpler, more exploratory policies like Epsilon-Greedy when contextual dependencies are well captured by the vine copula and when the reward surface is appropriately modeled using Gaussian Processes.

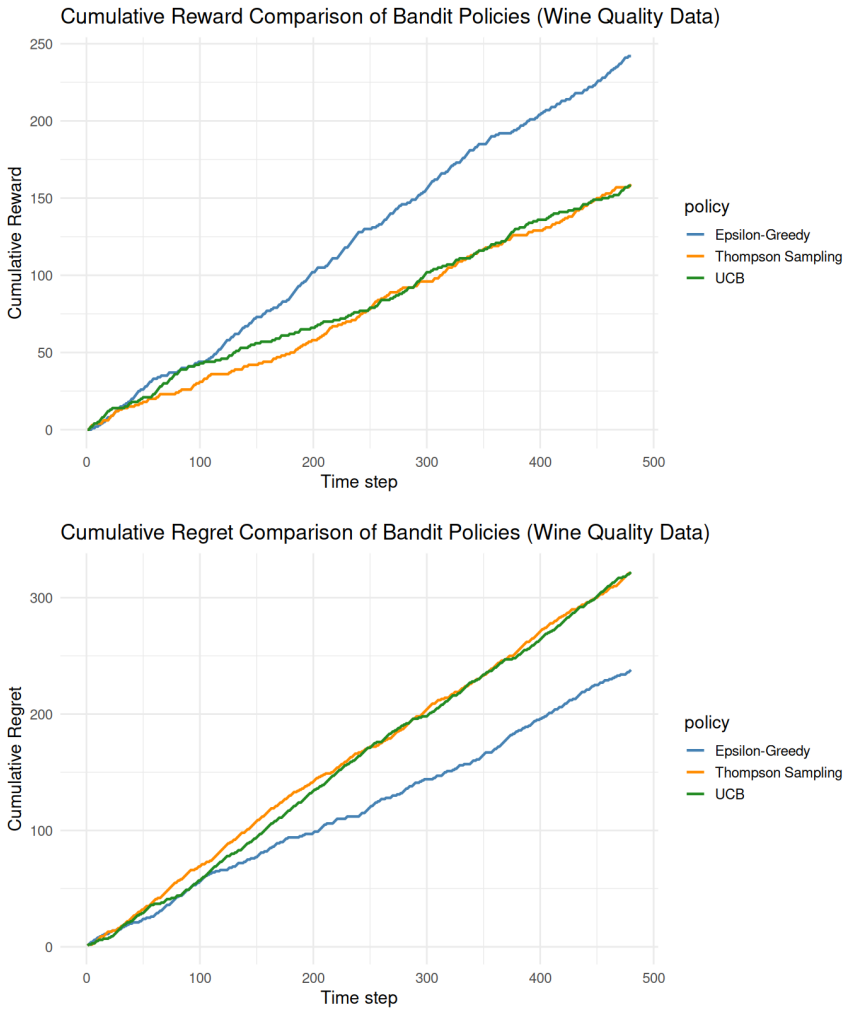

Figure 2. Cumulative reward (**top**) and cumulative regret (**bottom**) comparisons of contextual bandit policies—Epsilon-Greedy, TS, and UCB—on the Wine Quality dataset.

Table 4 summarizes the performance of three contextual bandit policies—Epsilon-Greedy, TS, and UCB—applied to a reward simulation based on the Wine Quality dataset. The dataset consists of physicochemical features of red wines, and the contextual bandit framework uses these features as context to sequentially select among discrete quality categories (arms) modeled as rewards.

Table 4. Summary of contextual bandit policy performance (Wine Quality data).

Policy	Final Cumulative Reward	Final Cumulative Regret	Mean Cumulative Reward	Mean Cumulative Regret	SD Cumulative Reward	SD Cumulative Regret
Epsilon-Greedy	101.08	9.72	50.95	5.13	29.06	2.71
TS	60.63	50.18	30.02	26.06	17.53	14.20
UCB	57.42	53.39	28.87	27.21	16.11	15.62

The *Final Cumulative Reward* indicates the total accumulated reward obtained by each policy over the test period, with the Epsilon-Greedy policy achieving the highest total reward (101.08), demonstrating more effective exploration–exploitation balance in this setup. Correspondingly, the *Final Cumulative Regret*, which measures the gap between the policy's chosen actions and the optimal ones, is lowest for Epsilon-Greedy (9.72), indicating fewer missed opportunities.

The *Mean Cumulative Reward* and *Mean Cumulative Regret* provide average performance over time steps, confirming that Epsilon-Greedy consistently outperforms the other policies. The standard deviations (SD Cumulative Reward and SD Cumulative Regret) indicate variability across simulation runs, with Epsilon-Greedy showing slightly higher variability in rewards but lower variability in regret, suggesting stable policy performance.

These results demonstrate that, in the context of wine quality prediction using physicochemical measurements as contextual information, the Epsilon-Greedy policy may provide more robust and efficient reward accumulation compared to TS and UCB strategies.

5.2. Boston Housing

We utilized the Boston Housing dataset, a classical benchmark for regression tasks, originally compiled by the U.S. Census Service. The dataset includes $n = 506$ observations of housing data across $d = 13$ numerical predictor variables, such as average number of rooms per dwelling, crime rate, pupil–teacher ratio, and property tax rate. The target variable, `medv`, represents the median value of owner-occupied homes in thousands of dollars.

To formulate the contextual bandit problem, we treated the 13 standardized predictor variables as the *context* $\mathbf{x}_t \in \mathbb{R}^d$ observed at each time step $t = 1, \ldots, n$. The continuous outcome variable, `medv`, was discretized into three categories using empirical tertiles. This categorization defined the three discrete *arms* corresponding to low, medium, and high housing prices. Each arm assignment was one-hot encoded into a reward matrix $\mathbf{R} \in \{0,1\}^{n\times 3}$, where $R_{ij} = 1$ indicates that the ith observation belongs to category j, for $i = 1, \ldots, n$ and $j = 1, 2, 3$.

A training set consisting of $n_{\text{train}} = 350$ randomly selected observations was used to train the model, while the remaining $n_{\text{test}} = 156$ observations formed the test set. We denote training and testing contexts and rewards by $\mathbf{X}_{\text{train}}$, $\mathbf{R}_{\text{train}}$ and $\mathbf{X}_{\text{test}}$, $\mathbf{R}_{\text{test}}$, respectively. All contextual variables were standardized to have zero mean and unit variance prior to modeling to ensure numerical stability and scale invariance.

To transform the context data into the unit hypercube $[0,1]^d$, we applied the standard normal CDF Φ to each standardized predictor via the probability integral transform.

To capture higher-order dependencies among contextual variables, we employed a regular vine copula (R-vine) structure [6,18]. The R-vine was fitted on the PIT-transformed training data using the `RVineStructureSelect` algorithm. If successful, this model was used to re-transform the PIT-transformed test set, generating a dependency-preserving representation of test contexts, which was then fed into the bandit policy models.

For each arm $j \in \{1, 2, 3\}$, we fitted a Gaussian Process (GP) model using the `laGP` framework [19]. The GP model provides a posterior predictive distribution over the expected reward for each arm, given a context. Each GP model used a squared exponential covariance function with fixed hyperparameters: length-scale $d = 0.5$ and nugget (noise term) $g = 10^{-6}$ for numerical stability.

We compared three contextual bandit policies:

- **TS:** For each context $\mathbf{x}_t$, a reward is sampled from the posterior predictive distribution of each arm's GP model, and the arm with the highest sampled reward is selected:

$$a_t = \arg\max_j \tilde{r}_{j,t}, \quad \tilde{r}_{j,t} \sim \mathcal{N}\left(\mu_{j,t}, \sigma^2_{j,t}\right),$$

  where $\mu_{j,t}$ and $\sigma^2_{j,t}$ are the GP posterior mean and variance for arm j at time t.
- **Epsilon-Greedy (EG):** With probability $\epsilon = 0.1$, a random arm is selected uniformly (exploration); otherwise, the arm with the highest posterior mean reward is chosen (exploitation):

$$a_t = \begin{cases} \text{Uniform}(\{1,2,3\}) & \text{with probability } \epsilon, \\ \arg\max_j \mu_{j,t} & \text{with probability } 1-\epsilon. \end{cases}$$

- **UCB:** The arm maximizing the UCB is selected:

$$a_t = \arg\max_j \left[\mu_{j,t} + \sqrt{2\log t \cdot \sigma^2_{j,t}}\right].$$

We simulated the policies over the test set and recorded the selected arm, observed reward, and the optimal reward for each observation. Let $r_t = r_{t,a_t}$ denote the reward obtained by a policy at time t from the chosen arm a_t, and let $r^*_t = \max_j r_{t,j}$ be the reward from the optimal arm. The cumulative reward and cumulative regret are computed as the Equations (1) and (2).

A vine copula was employed to model nonlinear dependencies among contextual variables, and the reward function was modeled using Gaussian Process regression. Among the three policies, Epsilon-Greedy consistently outperformed the others by achieving the highest cumulative reward and the lowest cumulative regret across time steps.

Figure 3 illustrates the comparative performance of three CMAD policies using the Boston Housing dataset. The top panel reveals that the Epsilon-Greedy policy rapidly accumulates rewards over time and ultimately achieves the best performance among the three policies. TS also demonstrates competitive performance, closely tracking Epsilon-Greedy until approximately the 100th time step, after which it diverges. In contrast, UCB lags behind both in cumulative reward and shows the weakest performance throughout the experiment.

The bottom panel shows that Epsilon-Greedy achieves the lowest cumulative regret, indicating a higher frequency of optimal action selections. TS results in slightly higher regret, while UCB exhibits the highest regret, suggesting it may be too conservative in its action choices under the complex contextual structure modeled by the vine copula. These results suggest that in environments where contextual variables are strongly interdependent and the reward surface is smooth but nonlinear, simpler and more explorative strategies such as Epsilon-Greedy can outperform theoretically more principled methods like UCB or TS.

The Boston Housing dataset contains information on various housing attributes in the Boston area, including features such as crime rate, the average number of rooms, and accessibility to highways, along with the median value of owner-occupied homes (medv), which is the target variable. In this study, the continuous target variable was discretized into three categories (arms) representing low, medium, and high median home values. The contextual bandit algorithms—Epsilon-Greedy, TS, and UCB—were evaluated on their ability to select the optimal arm (housing value category) based on contextual features.

As shown in Table 5, the *Epsilon-Greedy* policy outperformed the other two policies with the highest final cumulative reward (101.08) and the lowest cumulative regret (9.72).

This suggests that Epsilon-Greedy more effectively balanced exploration and exploitation in this setting. Both TS and UCB showed lower rewards and higher regrets, indicating relatively less efficient decision-making over the testing period. The standard deviations indicate that the Epsilon-Greedy policy also had more stable performance compared to the other policies.

Table 5. Performance summary of contextual bandit policies on Boston Housing data.

Policy	Final Cumulative Reward	Final Cumulative Regret	Mean Cumulative Reward	Mean Cumulative Regret	SD Cumulative Reward	SD Cumulative Regret
Epsilon-Greedy	101.08	9.72	50.95	5.13	29.06	2.71
TS	60.63	50.18	30.02	26.06	17.53	14.20
UCB	57.42	53.39	28.87	27.21	16.11	15.62

These results highlight the potential of simple exploration strategies like Epsilon-Greedy for practical contextual bandit problems involving structured real-world datasets such as housing market data.

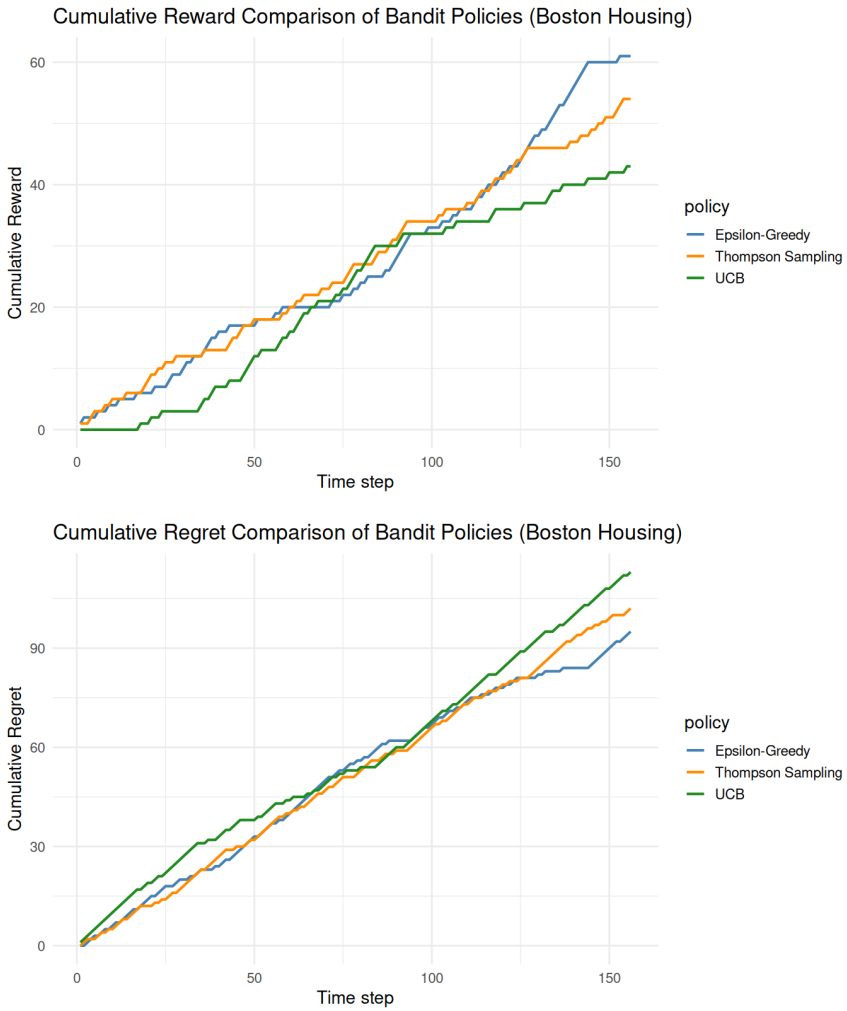

Figure 3. Cumulative reward (**top**) and cumulative regret (**bottom**) comparisons of contextual bandit policies—Epsilon-Greedy, TS, and UCB—on the Boston Housing dataset.

6. Discussion and Conclusions

This study evaluated the performance of three CMAB policies—Epsilon-Greedy, TS, and UCB—within a novel framework that integrates vine copula models to capture complex contextual dependencies and Gaussian Process (GP) regression for modeling reward

surfaces. We demonstrated the methodology on two benchmark datasets: Wine Quality and Boston Housing.

Our empirical findings show that the Epsilon-Greedy policy consistently achieves the highest cumulative reward and lowest cumulative regret across both datasets. This suggests that in high-dimensional settings with nonlinear contextual dependencies, simple yet robust exploration strategies can outperform more complex alternatives when the contextual structure is effectively modeled using copula-based transformations.

TS yields moderate performance, striking a balance between exploration and exploitation, but is eventually outperformed by Epsilon-Greedy. The UCB policy, despite its theoretical guarantees, exhibits the poorest performance in both reward and regret metrics, likely due to its overly conservative nature, which limits adaptability in environments with intricate contextual dependencies and smoothly varying, nonstationary reward functions.

These results highlight the critical role of flexible context modeling in CMAB problems. By leveraging vine copulas for rich dependence modeling alongside GP-based nonparametric reward estimation, our framework offers a promising approach for real-world sequential decision-making tasks such as personalized marketing, adaptive experimentation, and clinical trials.

Future work will focus on extending this framework to dynamic contexts, delayed rewards, and nonstationary environments, as well as exploring deep learning-based copula transformations and meta-learning strategies for improved policy adaptation.

Funding: This research received no external funding.

Institutional Review Board Statement: Not related in this research.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: We thank the three respected referees, Associated Editor, and Editor for their constructive and helpful suggestions, which led to substantial improvement in the revised version. For the sake of transparency and reproducibility, the R code for this study can be found in the following GitHub repository: R code GitHub site (https://github.com/kjonomi/Rcode/blob/main/Contextual_Multi-Armed_Bandits).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Russo, D.; Van Roy, B. Learning to Optimize via Information-Directed Sampling. *arXiv* **2017**, arXiv:1403.5556. [CrossRef]
2. Srinivas, N.; Krause, A.; Kakade, S.M.; Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In Proceedings of the 27th International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010; pp. 1015–1022.
3. Li, L.; Chu, W.; Langford, J.; Schapire, R.E. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 661–670.
4. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
5. Nelsen, R.B. *An Introduction to Copulas*, 2nd ed.; Springer: New York, NY, USA, 2006.
6. Aas, K.; Czado, C.; Frigessi, A.; Bakken, H. Pair-copula constructions of multiple dependence. *Insur. Math. Econ.* **2009**, *44*, 182–198. [CrossRef]
7. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
8. Joe, H. *Multivariate Models and Dependence Concepts*; Chapman & Hall: London, UK, 1997.
9. Johnson, N.L.; Kotz, S.; Balakrishnan, N. *Continuous Univariate Distributions, Volume 2*; Wiley-Interscience: Hoboken, NJ, USA, 1995.
10. Niederreiter, H. *Random Number Generation and Quasi-Monte Carlo Methods*; SIAM: Philadelphia, PA, USA, 1992.
11. Czado, C. *Analyzing Dependent Data with Vine Copulas: A Practical Guide with R*; Springer: Berlin/Heidelberg, Germany, 2019.
12. Quiñonero-Candela, J.; Rasmussen, C.E. A unifying view of sparse approximate Gaussian process regression. *J. Mach. Learn. Res.* **2005**, *6*, 1939–1959.

13. Russo, D.; Van Roy, B.; Kazerouni, A.; Osband, I.; Wen, Z. A tutorial on Thompson sampling. *Found. Trends Mach. Learn.* **2018**, *11*, 1–96. [CrossRef]
14. Auer, P.; Cesa-Bianchi, N.; Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* **2002**, *47*, 235–256. [CrossRef]
15. Lattimore, T.; Szepesvári, C. *Bandit Algorithms*; Cambridge University Press: Cambridge, UK, 2020.
16. Dua, D.; Graff, C. UCI Machine Learning Repository. 2019. Available online: http://archive.ics.uci.edu/ml (accessed on 1 May 2025).
17. Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decis. Support Syst.* **2009**, *47*, 547–553. [CrossRef]
18. Nagler, T.; Schepsmeier, U.; Stober, J.; Brechmann, E.C.; Graeler, B.; Erhardt, T.; Almeida, C.; Min, A.; Czado, C.; Hofmann, M.; et al. *The R Package VineCopula*, version 2.6.1; Statistical Inference of Vine Copulas; CRAN: Vienna, Austria, 2025.
19. Gramacy, R.B.; Lee, H.K.H. Bayesian treed Gaussian process models with an application to computer modeling. *J. Am. Stat. Assoc.* **2008**, *103*, 1119–1130. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.