Financial Asset Recommendation:Leveraging Multi-Armed Bandit Techniques

Bharat Khandelwal

Dec 2025

Abstract

Financial asset recommendation (FAR) is a challenging problem due to the highly volatile and non-stationary nature of financial markets. Unlike traditional recommender systems, FAR requires sequential decision-making under uncertainty, making Multi-Armed Bandit (MAB) algorithms a natural fit. In this work, we evaluate contextual and non-contextual bandit methods for asset recommendation using the FAR-Trans dataset [1], which contains real investor transactions, asset metadata, and market prices from 2018–2022. This period captures major disruptions such as the COVID-19 pandemic and the Russia–Ukraine war, resulting in unstable market regimes and weak short-term predictability.We construct a Contextual Multi Armed Bandit(CMAB) ready environment with weekly decision rounds and engineered features including 7-day momentum, 14-day volatility, market metadata, and customer risk profiles. Through five controlled scenarios, we compare UCB1, LinUCB, item–item Collaborative Filtering, and a random baseline using cumulative regret as the primary evaluation metric. Across all settings, regret grows nearly linearly and LinUCB performs similarly to UCB1, reflecting the extremely weak correlation between engineered features and forward-looking returns. The inclusion of customer *riskLevel* did not improve performance, and Collaborative Filtering consistently underperforms, highlighting the difficulty of applying static recommenders in volatile financial environments.Our findings align with prior contextual bandit research [2, 4] showing that contextual models offer limited benefit when features lack predictive power. Nevertheless, the study demonstrates that bandit algorithms remain promising tools for financial recommendation when informative context is available, and that cumulative regret serves as a meaningful metric for evaluating sequential decision policies in finance.

Contents

1 Introduction 3

2 Multi-Armed Bandits 4
- 2.1 Introduction to Multi-Armed Bandits 4
- 2.2 Regret as an Evaluation Metric 4
- 2.3 Non-Contextual Multi-Armed Bandits 5
- 2.4 Contextual Multi-Armed Bandits 5
  - 2.4.1 Linear UCB (LinUCB) 6

3 Methodology 7
- 3.1 Overview 7
- 3.2 Classical Recommendation Models 8
  - 3.2.1 Popularity-Based Recommendation 8
  - 3.2.2 Item–Item Collaborative Filtering 8
- 3.3 Multi-Armed Bandit Framework 8
  - 3.3.1 Non-Contextual Bandit: UCB1 9
  - 3.3.2 Contextual Bandit: LinUCB 9
  - 3.3.3 Random Recommender 9
  - 3.3.4 Item–Item CF as a Bandit Baseline 9
- 3.4 Experimental Scenarios 10
- 3.5 Summary 10

4 Experimentation and results 10
- 4.1 Dataset 10
- 4.2 Preprocessing Pipeline 12
  - 4.2.1 1) Asset Filtering and Stability Selection 12
  - 4.2.2 2) Weekly Snapshot Construction 12
  - 4.2.3 3) Customer Sampling 12
  - 4.2.4 4) Feature Engineering 12
  - 4.2.5 5) Reward Computation 13
  - 4.2.6 6) Final Dataset Construction 13
- 4.3 Feature-Reward Correlation Analysis 13
- 4.4 Results 14
  - 4.4.1 Experimental Scenarios 14
  - 4.4.2 Cumulative Regret for Asset-Only Scenarios (1–4) 14
  - 4.4.3 Impact of Customer Risk Level (Scenario 5) 17
  - 4.4.4 Summary of Cumulative Regret Across Scenarios 18

5 Conclusion 19

6 References 20

1 Introduction

Financial technology (fintech) has rapidly transformed investor engagement with markets, increasing interest in automated financial asset recommendation (FAR) methods. Unlike e-commerce, FAR operates in a dynamic and unpredictable environment that requires recommendations to adapt to changing asset trends and investor needs. Traditional approaches, such as transaction-based collaborative filtering or technical indicators, often struggle to keep pace, particularly during rapid market shifts.

Sanz-Cruzado et al.[1] introduced the FAR-Trans benchmark dataset, containing anonymized retail investor transactions, asset details, and historical prices from a European financial institution. They evaluated eleven algorithms based on profitability and transaction data, but did not include multi-armed bandit (MAB) methods, despite their suitability for decision-making in uncertain and evolving environments.

Recent literature has demonstrated growing interest in applying contextual bandits to financial decision problems such as portfolio optimization[6], contextual combinatorial allocation [4], hedging with contextual k-armed bandits[3], and stock-level trading decisions[5].This trend is evident in works exploring applications in credit risk, asset allocation, and online portfolio selection[10]. These studies demonstrate that bandit algorithms can effectively balance exploration and exploitation in the face of market uncertainty. However, these applications typically focus on trading or portfolio construction, not asset recommendation, and few use real customer transaction data.

In this work, we extend the FAR-Trans benchmark by evaluating LinUCB and UCB1 under a setting based on 253 weekly timestamps (Mondays) and asset-level features such as 7-day momentum and 14-day volatility. Our goal is not to make large-scale recommendations, but to analyze how bandit algorithms behave in a realistic yet moderate-sized sequential environment using publicly available financial data. Because most experiments rely solely on asset-level features, the recommendations are not personalized; we include one experiment incorporating customer risk levels, but observe similar results, suggesting limited sensitivity to user features in this setting. More on this in the methodology section.

A key property of contextual multi-armed bandits is that performance strongly depends on the correlation between contextual features and expected rewards. As established in the LinUCB framework[7] and reinforced by empirical evaluations in contextual bandit surveys, the algorithm benefits only when features contain predictive information about reward outcomes. When features are weakly correlated with returns, as can occur with short-horizon momentum or volatility signals, the advantage of contextual bandits over non-contextual methods becomes minimal, which aligns with our findings.

Further, the classical stochastic MAB formulation underlying algorithms such as UCB1 presumes that each arm's reward distribution is independent and stationary across rounds. Financial time-series data, by contrast, exhibit strong temporal non-stationarity driven by market regimes, volatility cycles, and temporal dependencies. Prior work has shown that under such conditions, non-contextual bandits can suffer substantial performance degradation because they cannot adapt to time-varying reward structures[8, 9].

Our contributions are twofold.

- First, we provide the evaluation of contextual and non-contextual bandit algorithms on the FAR-Trans dataset, filling a gap in the existing benchmark.
- Second, we introduce cumulative regret as a complementary metric to assess sequential learning behavior. Regret reveals how algorithm performance evolves over time and exposes the persistent non-stationarity of financial markets, which is not captured by static accuracy metrics.

Across experiments, we find that contextual and non-contextual bandits exhibit comparable performance when asset features have limited predictive power, while regret analysis highlights the challenge of learning stable decision policies in a highly time-varying environment.

2 Multi-Armed Bandits

2.1 Introduction to Multi-Armed Bandits

Multi-Armed Bandits (MABs) provide a mathematical framework for sequential decision-making under uncertainty. Inspired by the classical "one-armed bandit" slot machine analogy, a decision-maker (the *agent*) chooses among a set of K actions (arms) over multiple rounds. After selecting an arm, the agent observes a stochastic reward, but does not directly observe the reward distribution of the unchosen arms. The objective is to learn a strategy that balances *exploration* (trying different arms to gather information) and *exploitation* (choosing arms that currently appear most profitable) in order to maximize cumulative reward over time.

Formally, let $\mathcal{A} = \{1, \ldots, K\}$ denote the set of arms. At each round $t = 1, \ldots, T$, the agent selects an arm $a_t \in \mathcal{A}$ and receives a random reward $r_t \in \mathbb{R}$ drawn from an unknown distribution associated with arm a_t. Each arm a has an unknown expected reward $\mu_a = \mathbb{E}[r_t \mid a_t = a]$, and the optimal arm is $a^\star = \arg\max_{a \in \mathcal{A}} \mu_a$. Because the agent does not know the μ_a in advance, it must learn them while simultaneously making profitable choices.

2.2 Regret as an Evaluation Metric

Performance in the MAB setting is typically measured using *cumulative regret*. Regret quantifies how much reward is lost by following a learning policy instead of always playing the optimal arm $a^\star$ with full information. The cumulative regret after T rounds is defined as

$$R_T = T\mu_{a^\star} - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right], \tag{1}$$

where $\mu_{a^\star}$ is the expected reward of the best arm and $\sum_{t=1}^{T} r_t$ is the total reward obtained by the algorithm up to time T. A well-designed bandit algorithm aims to achieve *sublinear* regret, i.e., $R_T = o(T)$, so that the average regret per round $R_T/T \to 0$ as T grows. Intuitively, low regret means the agent quickly learns to behave almost as well as if it had known the optimal arm from the start.

2.3 Non-Contextual Multi-Armed Bandits

In the classical or *non-contextual* MAB setting, the environment is assumed to be stationary and the reward distribution of each arm does not depend on any external information beyond the arm identity. The agent's decision at time t is based solely on the history of past actions and observed rewards:

$$\mathcal{H}_t = \{(a_1, r_1), \ldots, (a_{t-1}, r_{t-1})\}.$$

The key challenge is to estimate the expected reward μ_a for each arm while minimizing the regret defined above.

Several classic algorithms have been proposed to address this trade-off, including ϵ-greedy, Upper Confidence Bound (UCB) methods, and Thompson Sampling. For example, the UCB1 algorithm maintains an empirical mean reward $\hat{\mu}_a(t)$ and a confidence term for each arm a. At each round t, it selects the arm

$$a_t = \arg\max_{a \in \mathcal{A}} \left(\hat{\mu}_a(t) + c\sqrt{\frac{2\log t}{N_a(t)}}\right), \tag{2}$$

where $N_a(t)$ is the number of times arm a has been pulled up to round t and $c > 0$ is a parameter controlling the degree of exploration. The confidence term encourages the agent to occasionally try under-explored arms, ensuring that all arms are tested sufficiently often.

2.4 Contextual Multi-Armed Bandits

While non-contextual bandits assume that rewards depend only on the arm identity, many real-world applications, including financial recommendation and personalized decision-making, involve additional side information or *context.* In *contextual* Multi-Armed Bandits, the expected reward depends on both the chosen arm and an observed context describing the current state of the environment or user.

At each round t, the agent observes a context vector $x_t \in \mathbb{R}^d$ (e.g., market conditions, asset features, or user attributes), then selects an arm $a_t \in \mathcal{A}$ and receives a reward r_t whose distribution depends on (x_t, a_t). The goal is to learn a policy π that maps contexts to arms,

$$\pi : \mathbb{R}^d \to \mathcal{A},$$

so as to maximize the expected cumulative reward or, equivalently, minimize contextual regret.

A common and interpretable formulation is the *linear contextual bandit* model, which assumes that the expected reward of each arm is a linear function of the context:

$$\mathbb{E}[r_t \mid x_t, a_t] = x_t^\top \theta^\star_{a_t}, \tag{3}$$

where $\theta^\star_a \in \mathbb{R}^d$ is an unknown parameter vector associated with arm a. In many implementations, including LinUCB, contexts may be arm-specific (i.e., $x_{t,a}$ for each arm a at round t), and the expected reward is modeled as

$$\mathbb{E}[r_t \mid x_{t,a}, a] = x_{t,a}^\top \theta^\star, \tag{4}$$

with a shared parameter vector $\theta^\star$ across arms. In both cases, learning an effective policy crucially depends on the relationship between context features and rewards: if the features are informative and correlated with the reward, the agent can rapidly identify high-performing actions in different states.

2.4.1 Linear UCB (LinUCB)

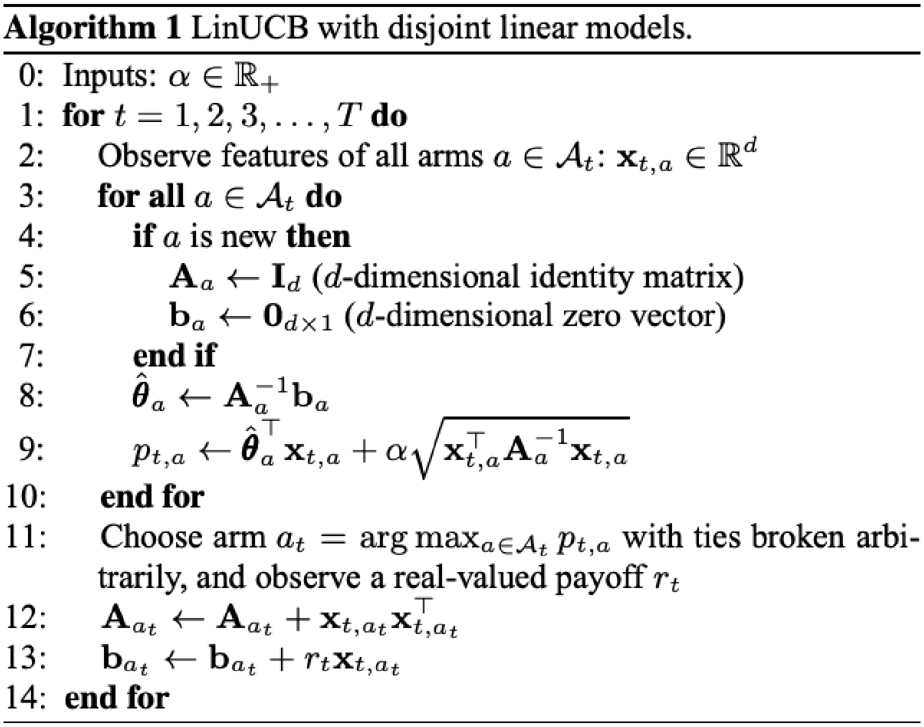
The LinUCB algorithm is a prominent example of a contextual bandit method that combines linear reward modeling with the optimism-in-the-face-of-uncertainty principle. It maintains an estimate $\hat{\theta}_t$ of the unknown parameter vector $\theta^\star$ and a covariance matrix A_t summarizing how much has been learned about the feature space. At each round t, for each arm a with context $x_{t,a}$, LinUCB computes an *upper confidence bound*

$$p_{t,a} = x_{t,a}^\top \hat{\theta}_t + \alpha \sqrt{x_{t,a}^\top A_t^{-1} x_{t,a}}, \tag{5}$$

where $\alpha > 0$ controls the exploration-exploitation balance. The arm selected at round t is

$$a_t = \arg\max_{a \in \mathcal{A}} p_{t,a}.$$

The first term $x_{t,a}^\top \hat{\theta}_t$ represents the current estimate of expected reward (exploitation), while the second term reflects the uncertainty around that estimate (exploration). After observing the reward r_t, LinUCB updates A_t and $\hat{\theta}_t$ using standard ridge regression-style updates, incorporating both the new context and reward signal.



Algorithm 1 LinUCB with disjoint linear models.

```
Inputs: $\alpha \in \mathbb{R}_+$
for $t = 1, 2, 3, \ldots, T$ do
  Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
  for all $a \in \mathcal{A}_t$ do
    if $a$ is new then
      $\mathbf{A}_a \leftarrow \mathbf{I}_d$ ($d$-dimensional identity matrix)
      $\mathbf{b}_a \leftarrow \mathbf{0}_{d \times 1}$ ($d$-dimensional zero vector)
    end if
    $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
    $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}}$
  end for
  Choose arm $a_t = \arg\max_{a \in \mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff $r_t$
  $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^\top$
  $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t \mathbf{x}_{t,a_t}$
end for
```

Figure 1: LinUCB pseudo code[7]

By explicitly modeling the dependence of rewards on features, contextual bandit methods like LinUCB can better adapt to non-stationary or heterogeneous environments where different contexts favor different actions. This makes them particularly well-suited for tasks such as financial asset recommendation, where the correlation between contextual features (e.g., momentum, volatility, or risk level) and reward plays a central role in learning effective policies.

3 Methodology

3.1 Overview

The objective of this study is to evaluate the effectiveness of different recommendation strategies for financial asset selection, ranging from classical collaborative filtering approaches to contextual and non-contextual Multi-Armed Bandit (MAB) algorithms. The methodology consists of two major phases: (1) benchmark recommendation using popularity-based and item–item collaborative filtering, and (2) sequential decision-making using MAB algorithms measured through cumulative regret. All experiments are performed on subsets of the FAR-Trans dataset, including multiple scenarios varying in asset count, customer count, and feature configurations.

3.2 Classical Recommendation Models

Before evaluating bandit-based approaches, I implemented traditional top-k recommendation methods as baselines. These models operate purely on historical transactions (restricted to *buy* actions) and do not incorporate sequential decision-making or reward estimation.

3.2.1 Popularity-Based Recommendation

The popularity-based recommender ranks assets by their global purchase frequency. Formally, let $\mathcal{A}$ denote the set of assets, and $c(a)$ the count of buy transactions for an asset $a \in \mathcal{A}$. The score for each asset is:

$$s(a) = c(a) \tag{6}$$

Assets are recommended based on descending values of $s(a)$. This method assumes that frequently purchased assets are likely to be desirable to new customers.

3.2.2 Item–Item Collaborative Filtering

Item–item CF computes similarity between assets based on customer co-purchase behavior. For assets i and j, let $U(i)$ be the set of customers who purchased asset i. Similarity is computed using cosine similarity:

$$\mathrm{sim}(i, j) = \frac{|U(i) \cap U(j)|}{\sqrt{|U(i)|\,|U(j)|}} \tag{7}$$

For a customer u with purchase history H_u, the score for an asset a is:

$$s_u(a) = \sum_{h \in H_u} \mathrm{sim}(a, h) \tag{8}$$

Ranking is performed using $s_u(a)$, and recommendations are evaluated using HitRate@k and Recall@k.

3.3 Multi-Armed Bandit Framework

After establishing classical baselines, I employed Multi-Armed Bandit algorithms to model financial asset recommendation as a sequential decision-making problem under uncertainty. Unlike static CF-based models, MAB algorithms adaptively learn from rewards generated over time.

Let $\mathcal{A} = \{1, \ldots, K\}$ be the set of assets (arms). At each round t, the algorithm selects an arm a_t and observes a reward r_t. The goal is to minimize cumulative regret:

$$R_T = T\mu_{a^\star} - \sum_{t=1}^{T} r_t, \tag{9}$$

where $\mu_{a^\star}$ is the expected reward of the optimal arm.

3.3.1 Non-Contextual Bandit: UCB1

The UCB1 algorithm assumes stationary arm rewards and selects the arm maximizing:

$$a_t = \arg\max_{a \in \mathcal{A}} \left(\hat{\mu}_a(t) + \sqrt{\frac{2 \log t}{N_a(t)}} \right) \tag{10}$$

where $\hat{\mu}_a(t)$ is the empirical mean reward and $N_a(t)$ is the number of times arm a has been selected. UCB1 serves as a strong baseline for non-contextual asset selection.

3.3.2 Contextual Bandit: LinUCB

To incorporate asset-level features and customer attributes, I used the linear contextual bandit model (LinUCB). For each arm a at round t, with feature vector $x_{t,a}$, LinUCB computes:

$$p_{t,a} = x_{t,a}^\top \hat{\theta}_t + \alpha \sqrt{x_{t,a}^\top A_t^{-1} x_{t,a}} \tag{11}$$

where:

- A_t is the feature covariance matrix,
- $\hat{\theta}_t$ is the estimated parameter vector,
- α controls exploration.

The selected action is $a_t = \arg\max_a p_{t,a}$. Rewards update the parameters via ridge regression-style updates. LinUCB is well suited to financial assets because reward patterns often correlate with features such as momentum, volatility, and risk level.

3.3.3 Random Recommender

A random baseline selects arms uniformly:

$$a_t \sim \text{Uniform}(\mathcal{A}) \tag{12}$$

This provides a lower-bound reference for regret comparison.

3.3.4 Item–Item CF as a Bandit Baseline

For completeness, the previously implemented item–item CF recommender is included in the regret-based evaluation. At each round, CF recommends the top-ranked asset based on similarity to past purchases.

3.4 Experimental Scenarios

To investigate how the number of assets, number of customers, and availability of context features affect model performance, I conducted regret-based comparison across five controlled scenarios:

1. **68 assets, 500 customers, asset features only**
2. **68 assets, 3000 customers, asset features only**
3. **253 assets, 500 customers, asset features only**
4. **253 assets, 1000 customers, asset features only**
5. **253 assets, 1000 customers, asset features + customer risk level**

For each scenario, the same transaction sampling strategy, reward computation, and sequential evaluation setup is applied to ensure consistent comparison. Contextual bandits (LinUCB) use only asset features in Scenarios 1–4 and incorporate customer risk level in Scenario 5.

3.5 Summary

This methodology enables a systematic comparison between static recommenders (popularity and item–item CF) and adaptive, reward-driven recommenders (UCB1, LinUCB, and random). By evaluating these models through cumulative regret across multiple experimental settings, the study reveals how feature availability, customer volume, and asset universe size influence the effectiveness of sequential recommendation strategies in financial markets.

4 Experimentation and results

4.1 Dataset

The experiments in this project are based on the FAR-Trans dataset, a real-world investment dataset released by a European financial institution and documented by Sanz-Cruzado et al. (2024)[1],which contains real investor transactions, asset metadata, and market prices from 2018–2022. These market phases were heavily influenced by volatility spikes during COVID-19 and geopolitical tensions[12]. The dataset provides a broad view of retail investment activity across multiple asset classes and includes customer profiles, asset metadata, market information, and historical close prices. While the dataset is not specifically designed for bandit algorithms, it offers sufficient structure to support the development of an initial financial asset recommendation model by enabling the construction of decision rounds, asset features, and reward signals.

The dataset contains the following key components:

Customer Information:

- `customerID` – Unique customer identifier
- `customerType` – Mass, Premium, Professional, Legal Entity, etc.
- `riskLevel` – Conservative, Income, Balanced, Aggressive, predicted variants
- `investmentCapacity` – Four investment capacity brackets and predicted versions
- `lastQuestionnaireDate` – Timestamp of the most recent risk-profile update
- `timestamp` – Metadata update timestamp

Asset Information:

- `ISIN` – Unique asset identifier
- `assetName`, `assetShortName`
- `assetCategory` – Stock, Bond, Mutual Fund
- `assetSubCategory`, `sector`, `industry`
- `marketID`
- `timestamp`

Market Information:

- `marketID`, `exchangeID`, `name`, `country`
- `tradingDays`, `tradingHours`
- `marketClass`

Close Prices:

- `ISIN`
- `timestamp` – Daily
- `closePrice` – Price in euros

Investment Transactions:

- `customerID`, `ISIN`
- `transactionID`, `transactionType`
- `timestamp`
- `units`, `totalValue`
- `channel` – Online, Phone, Branch
- `marketID`

4.2 Preprocessing Pipeline

The FAR-Trans dataset in its raw form is not directly suitable for contextual or non-contextual multi-armed bandit algorithms. To prepare it for sequential decision-making experiments, a dedicated preprocessing pipeline was implemented to convert the dataset into structured weekly decision rounds with engineered features and reward labels. This pipeline guides the construction of the experimental setup used throughout the project.

4.2.1 1) Asset Filtering and Stability Selection

Assets with incomplete or irregular price histories are removed. Only those with sufficient continuous data for computing momentum and volatility features are retained. Depending on the experimental scenario, a stable universe of 68 or 253 assets is selected.

4.2.2 2) Weekly Snapshot Construction

To define bandit decision opportunities, all dates are aligned to Mondays. Each Monday forms a "decision round" that contains the available price information for all selected assets and is assigned a unique `snapshotID`.

4.2.3 3) Customer Sampling

For each decision round, a fixed number of customers (500, 1000, or 3000 depending on the scenario) is sampled. Most experiments use only asset-side features, while one controlled experiment incorporates customer `riskLevel` as an additional contextual feature. The performance remained similar, reinforcing that contextual bandits primarily leverage features correlated with reward outcomes.

4.2.4 4) Feature Engineering

For each asset in every weekly snapshot, short-term market dynamics are encoded using:

$$\text{momentum}_{7d} = \frac{P_t - P_{t-7}}{P_{t-7}}$$

$$\text{volatility}_{14d} = \sqrt{\frac{1}{14}\sum (r_i - \bar{r})^2}$$

Additional metadata features include `assetCategory` and `marketID`. In one experiment, customer `riskLevel` was included as an optional contextual component.

4.2.5 5) Reward Computation

Each asset's forward-looking reward is computed as the next week's return:

$$\text{reward} = \frac{P_{t+7} - P_t}{P_t}$$

This reward definition supports direct comparison across contextual (LinUCB), non-contextual (UCB1), and baseline recommendation strategies.

4.2.6 6) Final Dataset Construction

The result of the preprocessing pipeline is a structured CMAB-ready dataset where each row corresponds to a *(timestamp, customer, asset)* triplet with associated feature vectors and reward values. This dataset forms the foundation for all experiments conducted in this project.

4.3 Feature-Reward Correlation Analysis

Before evaluating the bandit algorithms, a correlation analysis was conducted to assess the predictive strength of the engineered asset-side features. Figure 2 reports the Pearson correlation matrix between the weekly features and the forward-looking reward (next-week return). The results reveal that all features, including 7-day momentum, 14-day volatility, country encoding, and market identifiers, exhibit extremely weak correlation with the realised reward. The strongest correlation with reward is only 0.04, indicating that short-term return features contain little exploitable predictive signal.

This weak feature-reward association provides a direct explanation for the observed behaviour in later experiments: contextual bandits such as LinUCB cannot leverage context effectively when contextual features do not correlate with reward outcomes. As a result, LinUCB performs similarly to the non-contextual UCB1 algorithm, and all algorithms exhibit nearly linear regret in the subsequent scenarios.

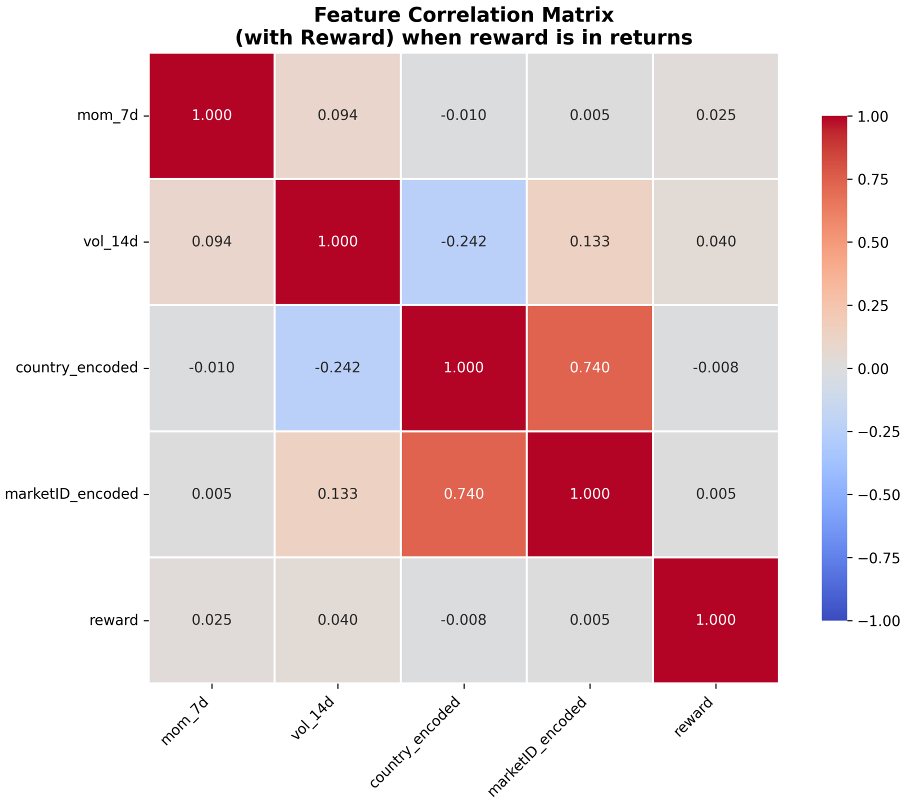


Figure 2: Feature–reward correlation matrix for weekly asset features. All asset-side features show extremely weak correlation with forward-looking reward, limiting the effectiveness of contextual bandits.

4.4 Results

4.4.1 Experimental Scenarios

Table 1 summarizes the five experimental configurations used to evaluate the recommendation algorithms. Scenarios vary the number of assets, the number of customers per weekly decision round, and whether customer risk level is included as an additional contextual feature.

4.4.2 Cumulative Regret for Asset-Only Scenarios (1–4)

Figures 3–6 show the cumulative regret trajectories for Random, item–item Collaborative Filtering (CF), UCB1, and LinUCB under the four asset-only scenarios. Across all configurations, the regret curves grow approximately linearly and remain close to each other throughout the horizon. This behaviour indicates that weekly asset returns in FAR-Trans are highly noisy and that the

Table 1: Summary of Experimental Scenarios

Scenario	Assets	Customers	Features Used
1	68	500	Asset-only (momentum, volatility, marketID, country)
2	68	3000	Asset-only (momentum, volatility, marketID, country)
3	253	500	Asset-only (momentum, volatility, marketID, country)
4	253	1000	Asset-only (momentum, volatility, marketID, country)
5	253	1000	Asset + customer risk level

short-term features used (7-day momentum and 14-day volatility) provide limited predictive structure. As a result, none of the algorithms is able to exploit a stable "best" asset to achieve sublinear regret.

In all four scenarios, LinUCB and UCB1 exhibit very similar cumulative regret, with only small fluctuations in relative ordering at different points in time. This suggests that the contextual information provided by momentum and volatility is not sufficiently correlated with future rewards to give LinUCB a consistent advantage over the non-contextual UCB1 algorithm. CF and Random baselines generally incur higher regret and do not close the gap to the bandit methods, reflecting the difficulty of applying static, interaction-based recommenders in a non-stationary financial setting.

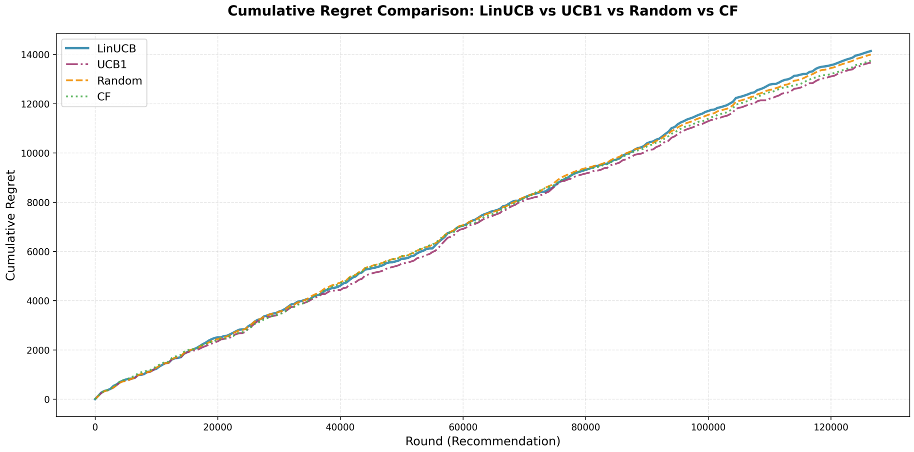


Figure 3: Cumulative regret comparison for 68 assets and 500 customers (Scenario 1). All algorithms exhibit nearly linear regret, with UCB1 performing slightly better than LinUCB

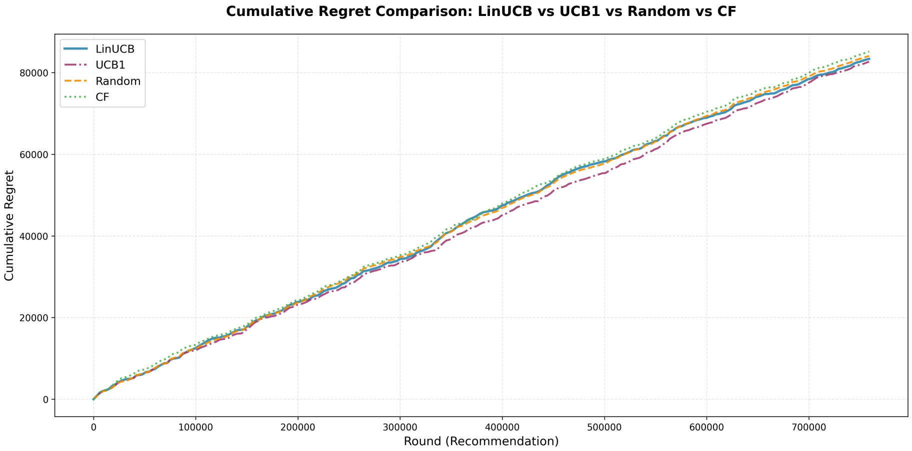


Figure 4: Cumulative regret comparison for 68 assets and 3000 customers (Scenario 2). Increasing the number of customers smooths the regret curves but does not change the relative ordering of algorithms.

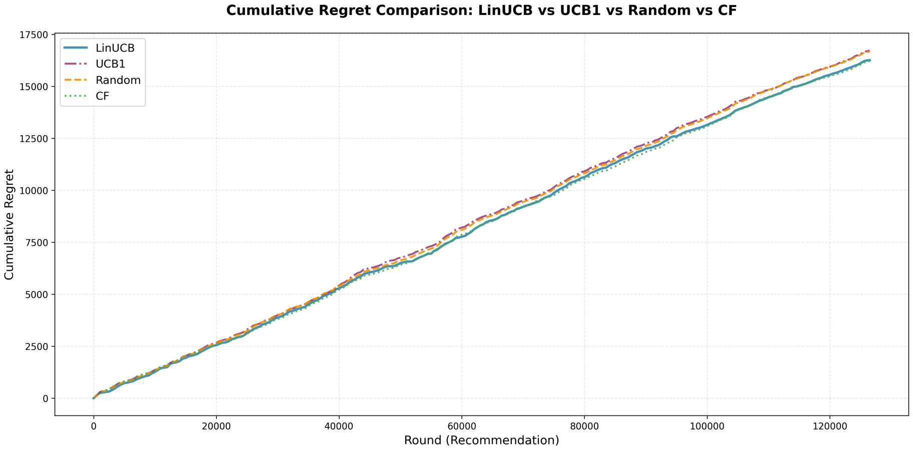


Figure 5: Cumulative regret comparison for 253 assets and 500 customers (Scenario 3). With more assets, all algorithms incur higher regret due to the increased exploration space, with LinUCB performing slighlty better than the others.

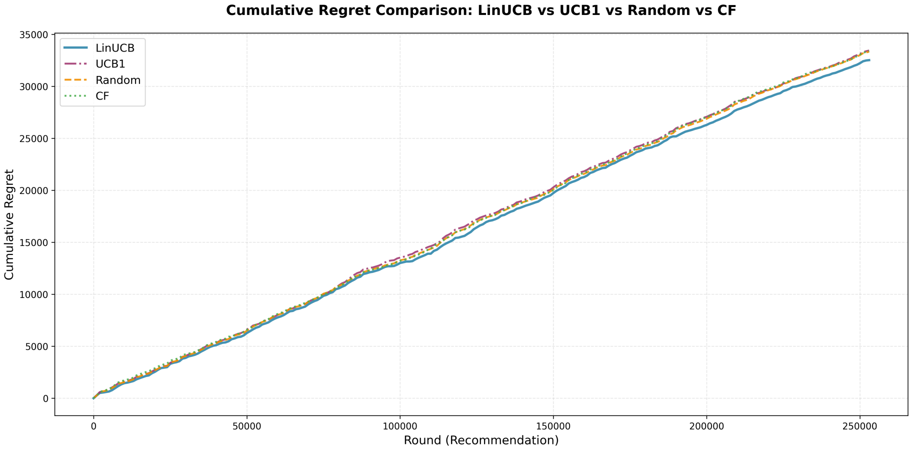


Figure 6: Cumulative regret comparison for 253 assets and 1000 customers (Scenario 4). Regret patterns remain consistent, with UCB1 and LinUCB showing nearly identical performance, although with more assets LinUCB performs better as we can see.

4.4.3 Impact of Customer Risk Level (Scenario 5)

Scenario 5 augments the feature space by including customer *riskLevel* alongside the asset-side features for LinUCB, while UCB1, CF, and Random remain unchanged. The corresponding cumulative regret curves are shown in Fig. 7. The trajectory of LinUCB with risk-based context almost overlaps with its asset-only counterpart and remains close to UCB1 throughout the horizon.

This result indicates that, in this dataset, customer risk level does not provide additional predictive power for short-term weekly returns. Although risk profiles capture differences in investor preferences, the realised rewards are still driven primarily by market movements rather than by customer-level heterogeneity. Consequently, the contextual bandit is unable to exploit this extra feature to reduce regret.

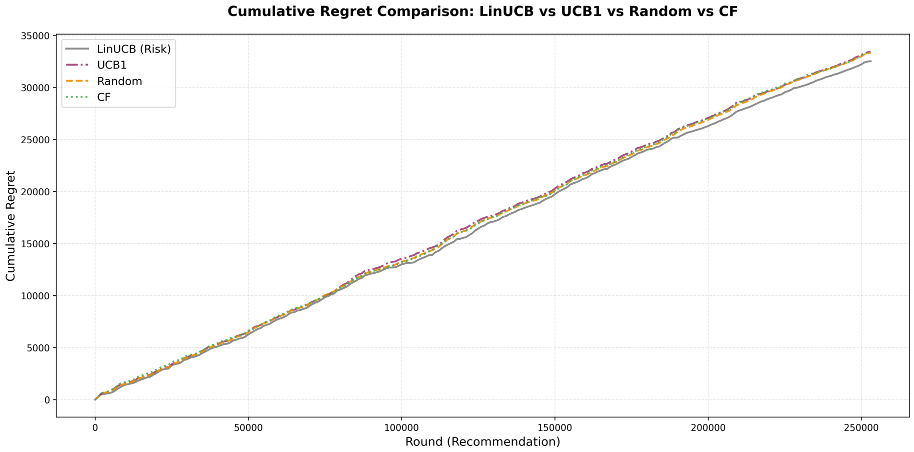


Figure 7: Cumulative regret comparison for 253 assets and 1000 customers with LinUCB using asset features and customer risk level (Scenario 5). The regret of LinUCB with risk-based context closely matches that of UCB1 and the asset-only LinUCB runs, indicating limited predictive value of the risk feature for weekly returns.

4.4.4 Summary of Cumulative Regret Across Scenarios

Table 2 reports the final cumulative regret for each algorithm at the end of the time horizon for all five scenarios. Across configurations, UCB1 and LinUCB consistently achieve the lowest regret and remain close to each other, while CF and Random generally perform worse. Introducing customer risk level in Scenario 5 does not materially change the performance of LinUCB, reinforcing the observation that contextual bandits only benefit when contextual features are strongly correlated with reward outcomes[2]. A Survey on Practical Applications of Bandits.Features such as momentum or volatility provide low predictive power for short-term reward, leading to regret curves that grow almost linearly[4].

Table 2: Final cumulative regret for all algorithms across scenarios.

Scenario	Random	CF	UCB1	LinUCB	LinUCB (Risk)
1	13991.18	13743.74	13664.14	14134.75	–
2	84052.80	85153.74	82649.09	83347.65	–
3	16682.10	16220.62	16726.83	16262.33	–
4	33332.86	33404.51	33453.66	32524.67	–
5	33332.86	33404.59	33453.66	–	32524.67

5 Conclusion

This project evaluated contextual and non-contextual multi-armed bandit algorithms for financial asset recommendation using the FAR-Trans dataset. After constructing a CMAB-ready environment with weekly snapshots and engineered features, five scenarios were tested across varying asset universes, customer volumes, and the inclusion of customer riskLevel.

Across all experiments, cumulative regret grew almost linearly, and LinUCB performed similarly to UCB1. This outcome reflects the extremely weak correlation between the engineered asset-side features and forward-looking returns, limiting the advantage that contextual models can obtain. Adding customer-riskLevel offered no improvement, and both Collaborative Filtering and Random baselines consistently underperformed. These findings are further influenced by the 2018–2022 period covered by FAR-Trans, a time of severe market disruptions due to COVID-19 and the Russia–Ukraine war, which introduced high volatility and reduced short-term predictability in asset behaviour.

Despite these limitations, the study affirms that contextual bandit approaches remain promising when contextual features exhibit meaningful correlation with reward signals, as highlighted in prior work [2, 4].Although deep reinforcement learning is gaining traction in similar tasks, bandits provide a more sample-efficient and interpretable framework in low-signal regimes[11]. In such settings, contextual models can leverage informative features to personalise recommendations and reduce uncertainty over time. Moreover, cumulative regret serves as an effective metric for assessing the quality of recommendation policies in financial environments, offering a transparent measure of sequential decision performance under uncertainty.

References

[1] J. Sanz-Cruzado et al., “FAR-Trans: A Benchmark Dataset for Financial Asset Recommendation,” 2024.
https://arxiv.org/pdf/2407.08692

[2] D. Bouneffouf and I. Rish, “A Survey on Practical Applications of Multi-Armed and Contextual Bandits,” 2019.
https://arxiv.org/pdf/1904.10040

[3] L. Cannelli, “Volatility regimes and non-stationary drifts pose a challenge for contextual bandits, as the reward distribution changes faster than the algorithm can adapt,” *Quantitative Finance*, 2023.
https://www.sciencedirect.com/science/article/pii/S240591882300017X

[4] L. Ni, “Contextual Combinatorial Bandits for Financial Allocation,” *Quantitative Finance*, 2023.
https://arxiv.org/pdf/2407.00567

[5] D. Kar, “Stock-Level Trading Decisions Using Contextual Multi-Armed Bandits,” 2024.
https://dl.acm.org/doi/pdf/10.1145/3638530.3664145

[6] W. Chen and Z. Huang, “Contextual Bandits for Dynamic Portfolio Allocation,” *Finance Research Letters*, 2024.
https://researchportal.hkust.edu.hk/en/publications/enhancing-the-performance-of-bandit-based-hyperparameter-optimiza-2/

[7] L. Li, J. Langford, and R. E. Schapire, “A Contextual-Bandit Approach to Personalized News Article Recommendation, 2012.
https://arxiv.org/pdf/1003.0146

[8] A. Slivkins, “Introduction to Multi-Armed Bandits,” *Foundations and Trends in Machine Learning*, 2019.
https://arxiv.org/pdf/1405.3316

[9] A.Goldenshluger, A.Zeevi “A LINEAR RESPONSE BANDIT PROBLEM,” 2014.
https://pubsonline.informs.org/doi/epdf/10.1287/11-SSY032

[10] de Freitas Fonseca, G., Coelho e Silva, L. and Lima de Castro “P.A. Improving Portfolio Optimization Results with Bandit Networks” 2025.
https://doi.org/10.1007/s10614-025-11090-0

[11] Z.Zhang, S.Zohren, S.Roberts“Deep Learning for Portfolio Optimization” 2021.
https://arxiv.org/pdf/2005.13665

[12] A. Baker and J. Wurgler,”Investor Sentiment and the Cross-Section of Stock Returns During Crise”,2006
https://papers.ssrn.com/sol3/papers.cfm?abstract$_i d = 464843$