

# Machine Learning in Economic Forecasting: Integrating Traditional Methods with a Tunable LSTM

Yi Luo

Math Department, Lafayette College, Easton, PA, USA  
Email: Luo45802@outlook.com

**How to cite this paper:** Luo, Y. (2026) Machine Learning in Economic Forecasting: Integrating Traditional Methods with a Tunable LSTM. *International Journal of Intelligence Science*, 16, 1-36.  
<https://doi.org/10.4236/ijis.2026.161001>

**Received:** October 8, 2025

**Accepted:** November 23, 2025

**Published:** November 26, 2025

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

The surge of digital data in tourism, finance and consumer markets demands predictive models capable of handling volatility, nonlinear dynamics, and long-term dependencies, where traditional econometric tools often fall short. This study makes two contributions. First, we benchmark four classical machine learning methods-K-Nearest Neighbors (KNN), Reinforcement Learning (RL), K-Means clustering, and Principal Component Analysis (PCA)-to establish their strengths and limitations in economic applications. KNN provides accurate cancellation predictions but lacks sequential awareness; RL adapts dynamically, yet suffers from long-horizon instability; K-Means reveals static consumer clusters but cannot capture temporal shifts; and PCA condenses macroeconomic indicators while discarding dynamic structure. Second, we extend this comparison with an in-depth ablation study of a Long Short-Term Memory (LSTM) framework, systematically varying activation functions, loss functions, and training regimes. This analysis reveals how architectural design governs the accuracy, robustness, and sensitivity of rare events forecasting, and shows that LSTM addresses the shortcomings of classical models by learning temporal dependencies while remaining tunable between tasks. Across baselines, KNN achieves  $AUC = 0.81$  in cancellation classification, while the tuned LSTM (Huber loss + Sigmoid head) achieves  $MAE \approx 11.3$  and  $Directional Accuracy (DA) \approx 0.68$ , outperforming static models in both magnitude error and trend capture. Collectively, our findings provide practical guidelines for choosing between interpretable classical baselines and adaptive deep learning architectures in dynamic economic and tourism environments.

## Keywords

Machine Learning, Economic Modeling, PCA, Deep Learning, Hybrid

## 1. Introduction

The accelerating pace of data generation in the digital economy has fundamentally reshaped how industries approach forecasting, decision-making, and consumer engagement. Volatile demand in tourism, dynamic fluctuations in financial markets, and multidimensional pressures of sustainability pose challenges that traditional econometric approaches often fail to capture. Classical models such as ARIMA or linear regression are constrained by linear assumptions and their limited capacity to model high-dimensional, nonlinear structures. These restrictions make them less effective in environments where complexity, volatility, and uncertainty dominate, ultimately limiting their applicability in modern economic forecasting.

Machine learning (ML) has emerged as a transformative force in addressing these limitations. By uncovering nonlinear dependencies, capturing complex interactions, and adapting to continuously evolving datasets, ML has demonstrated superiority over traditional approaches in a wide range of applications. In finance, ML has outperformed econometric restrictions in stock return predictability [1] and improved credit-risk modeling while maintaining transparency and regulatory compliance [2]. Reinforcement learning (RL) extends this adaptability to portfolio optimization and dynamic decision-making under uncertainty [3], while ML has also been applied to cryptocurrency markets [4] and infrastructure reliability [5]. For structure discovery and dimensionality reduction, PCA provides a natural exploratory tool [6], and ML methods have advanced asset pricing and factor modeling [7]. Beyond finance, ML is increasingly framed as a general-purpose technology reshaping economic analysis and practice [8] [9]. Its broader societal footprint and diffusion are reflected in cross-country indicators and community resources, from global well-being analytics [10] to modern RL curricula and tooling that build on classic foundations [11] [12]. At the intersection of sustainability and finance, large-scale reviews document ML's role in decision support [13], while in tourism analytics neural-network approaches synthesize model choices and performance across contexts [14]. Sustainability applications leverage ML to balance energy production, emissions, and growth [15]; methodologically, ML delivers improvements in general time-series forecasting [16] and in inflation prediction in data-rich environments [17]; environmental-health dynamics provide additional evidence [18]. For clustering and segmentation, accessible tools aid intuition and practice [19], and in travel and mobility, reviews of demand models highlight ML's advantages [20]. Data-intensive implementations extend to tourism demand [21], broader macroeconomic targets such as GDP growth prediction [22], and domain-specific LSTM studies in tourism forecasting with behavioral signals [23]. Collectively, these studies underscore ML not merely as a

---

technical toolkit, but as a general-purpose technology reshaping economic and behavioral analysis across domains.

Despite these advances, the current literature leaves two important gaps. First, systematic comparisons of classical ML models and deep time-series architectures remain scarce, particularly within a unified economic problem framework that encompasses consumer behavior, portfolio optimization, market segmentation, and macroeconomic forecasting. Most existing studies focus narrowly on one application area or evaluate only a single model in isolation, making it difficult to derive generalizable insights. Second, while LSTM models are increasingly applied to economics, few studies rigorously examine how internal design choices—such as activation functions, loss functions, and hyperparameter regimes—directly shape forecasting accuracy, robustness, and sensitivity to rare events. This lack of systematic evaluation limits both theoretical understanding and practical adoption of deep learning methods in applied economic forecasting.

To address these gaps, this paper integrates four foundational ML approaches—K-Nearest Neighbors (KNN), Reinforcement Learning (RL), K-Means clustering, and Principal Component Analysis (PCA)—to evaluate their strengths and limitations across diverse economic contexts. Together, these methods serve as representative benchmarks for predictive accuracy, interpretability, and structural insight. Building on this foundation, the centerpiece of our contribution lies in extending beyond static modeling with a systematically tuned Long Short-Term Memory (LSTM) baseline. By conducting a series of ablation experiments—varying architectural components such as Sigmoid, ReLU, and tanh activations; employing robust loss functions such as Huber; and systematically adjusting learning rates and batch sizes—we explicitly link design decisions to predictive outcomes in hotel booking and related economic datasets. This combination of horizontal benchmarking and vertical ablation analysis provides a comprehensive framework for evaluating both classical and deep learning models in economic forecasting.

By beginning with the broad challenges of economic forecasting, narrowing to the transformative role of ML, and concluding with a focused contribution on LSTM design and experimentation, our study emphasizes that while KNN, RL, K-Means, and PCA provide valuable interpretive and structural insights, it is the adaptability and tunability of LSTM architectures that best capture the nonlinear, dynamic nature of real-world economic and behavioral datasets. This dual approach not only highlights the comparative advantages of different models but also offers actionable methodological guidance for practitioners navigating trade-offs between interpretability, robustness, and predictive performance. Beyond computational gains, our ablation findings connect directly to economic mechanisms. Huber loss down-weights extreme residuals, mirroring robust estimators used when returns or demand exhibit heavy tails and volatility clusters; the Sigmoid head constrains outputs to economically meaningful ranges (e.g., bounded rates), discouraging implausible actions under uncertainty. KNN's local averaging

naturally captures neighborhood effects and peer imitation in consumer behavior, while PCA's factors align with canonical macro drivers (real activity and inflation-growth trade-offs). These links clarify how the architectures produce value in settings with shock propagation, price stickiness, and bounded rationality.

The remaining parts of this article are arranged as follows: Section 2 describes the models and methods employed, Section 3 introduces the datasets and data processing procedures, Section 4 presents the empirical results, Section 5 conducts a comprehensive discussion and draws out practical implications, and Section 6 summarizes the findings and outlines directions for future research.

## 2. Model Description

### 2.1. K-Nearest Neighbor (KNN) Method

The K-nearest neighbor (KNN) algorithm is a non-parametric, machine learning method based on instances widely used for classification and regression tasks. Unlike models that require explicit training, KNN operates on the principle of similarity by calculating the distance between data points in feature space to determine their proximity.

$$d(x, x_i) = \sqrt{\sum_{j=1}^n (x_j - x_{ij})^2} \quad (1)$$

For example, Euclidean distance works well for continuous data, while Manhattan distance, given by:

$$d_M(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

may perform better for categorical variables. The effectiveness of KNN is particularly evident in consumer behavior analysis because of its simplicity and ability to model nonlinear relationships without the need to make assumptions about data distribution. However, challenges like the *curse of dimensionality*, where performance degrades as the number of features increases, require careful pre-processing, including dimensionality reduction or feature selection.

In some cases, different features contribute differently to the distance calculation. The weighted KNN approach incorporates feature importance by adjusting the Euclidean distance formula using weights  $w_j$ , as shown below:

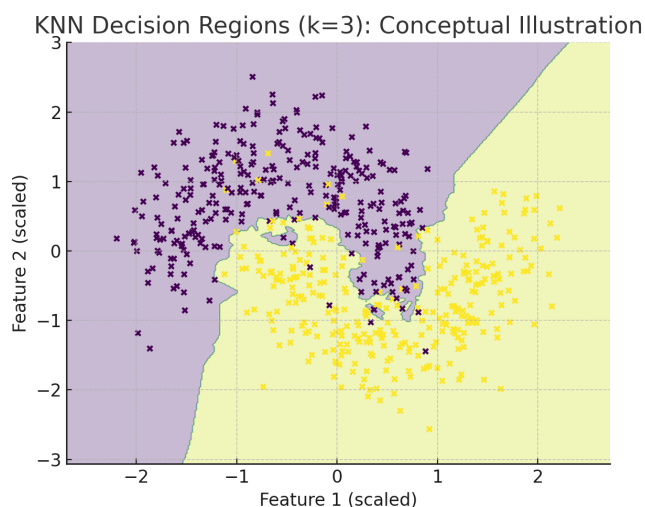
$$d_w(x, x_i) = \sum_{j=1}^n w_j (x_j - x_{ij})^2 \quad (3)$$

This allows the model to assign greater importance to relevant features and improve classification accuracy.

In order to effectively implement KNN in consumer behavior analysis, pre-processing steps such as standardization or standardization are essential to ensure fair comparisons between features at different scales. For example, income levels and product rating scales must be standardized to prevent algorithms from favoring a wide range of features. The optimal value of  $k$  determines the number of neigh-

bors considered and plays a key role in balancing bias and variance. A lower  $k$  may trap noise, leading to overfitting, while a higher  $k$  may oversimplify the pattern. Cross-validation can be used to determine the ideal  $k$  and improve the accuracy and robustness of the prediction. In the context of financial product recommendations, KNN performs well by clustering users with similar purchase histories or demographics. For instance, users who regularly invest in sustainable funds may have similar attributes, such as income level or risk appetite.

**Figure 1** provides a conceptual view of the KNN decision regions. Data points are classified based on their proximity to neighbors, here shown with  $k = 3$ . The irregular decision boundaries highlight how KNN flexibly adapts to nonlinear consumer behavior patterns. In practice, such boundaries capture differences in booking decisions or financial product choices, where nearby customers in feature space (e.g., income and past purchases) are more likely to share similar outcomes. This visual emphasizes why preprocessing and  $k$ -selection are critical for balancing overfitting and generalizability.



**Figure 1.** Conceptual KNN decision boundaries with  $k = 3$ . Each region shows how new data points are classified by proximity to their neighbors, reflecting consumer similarity in booking or product choice.

*Computational considerations.* For a query of dimensionality  $d$  over  $n$  points, naïve KNN requires  $O(nd)$  to compute distances and  $O(n \log n)$  for full sorting (or  $O(n)$  via selection). For larger  $n$ , approximate nearest neighbor (ANN) structures (KD/ball trees or graph-based ANN) reduce query time with high recall. We therefore report exact and ANN-accelerated timings in the supplement.

By identifying these similarities, KNN can recommend financial products that suit individual preferences. This makes KNN a powerful tool for personalized recommendations in financial services, in line with the growing trend of consumer-focused solutions in the industry. Evaluation metrics, such as accuracy, precision, recall, and f1 scores, are critical to evaluating the performance of a model. In ad-

dition to these metrics, a qualitative analysis of the recommendations can verify the relevance of the output. Because of its reliance on pair-distance calculations, KNN is computationally intensive for large data sets, and advanced techniques like KD-trees or ball trees can significantly speed up processing. This methodological framework makes KNN a valuable approach for consumer behavior research and recommendation systems.

## 2.2. Reinforcement Learning (RL) Method

Reinforcement learning (RL) is a machine learning paradigm in which agents learn optimal decision-making strategies by interacting with their environment. At the heart of RL is the modeling of problems using Markov decision processes (MDP), defined by states, actions, rewards, and transitions. A fundamental concept in reinforcement learning is the state-value function, which evaluates the expected return when following policy  $\pi$ :

$$V_{\pi}(s) = \mathbb{E}_{\pi} [R_t + \gamma V_{\pi}(s') | S_t = s] \quad (4)$$

where  $\gamma$  is the discount factor, ensuring that future rewards are appropriately weighted. Rewards are scaled by the rolling standard deviation of revenue to stabilize gradients; we set the discount factor to  $\gamma = 0.99$ . Policy-gradient stability is promoted via entropy regularization (coefficient 0.01) and cosine learning-rate decay. We track moving-average returns and monitor early plateaus to avoid premature convergence.

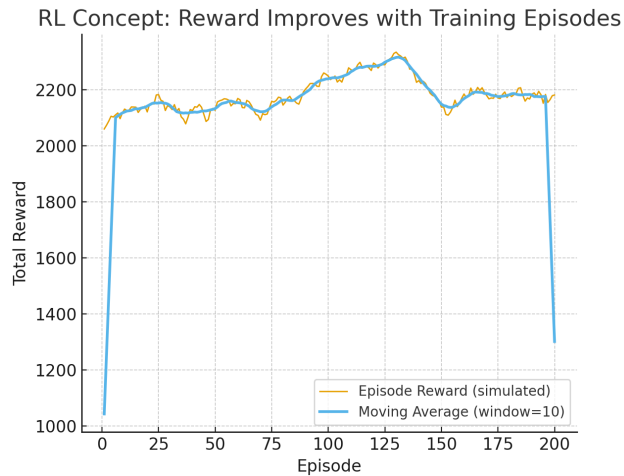
$$R(\tau) = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (5)$$

Each state represents the current state of the environment, an action represents a decision the agent may take, and a reward provides feedback on the desirability of the outcome. The goal is to maximize cumulative rewards over time, often balancing short-term gains with long-term goals. RL is particularly well-suited for portfolio optimization and pricing strategies, as these tasks involve sequential decisions under uncertainty.

**Figure 2** presents a conceptual reward curve from a reinforcement learning setting. The orange line shows noisy episode-level returns, while the blue moving average highlights the upward convergence trend. This reflects how an RL agent gradually improves its policy by balancing exploration and exploitation. In economic modeling, this process parallels portfolio optimization, where repeated trial-and-error enables the agent to identify more profitable and stable strategies over time.

The first step in implementing RL involves defining the environment. In financial applications, this may include historical asset price movements, volatility, or macroeconomic indicators. The action may represent the purchase, sale or holding of a financial instrument, while the return can be defined as the profit- or risk-adjusted return of the portfolio. The design of the reward function is crucial because it directly affects the learning trajectory of the agent. For example, penaliz-

ing excessive risk can lead agents to adopt a more conservative strategy that aligns with certain investment goals.



**Figure 2.** Conceptual RL training reward curve. Raw episode rewards fluctuate due to uncertainty, but the moving average shows upward convergence, reflecting the agent’s improving strategy.

Algorithm selection is another important consideration. Techniques such as Q-Learning, Deep Q-Networks (DQN), and the Policy Gradient method provide different capabilities depending on the complexity of the problem.

$$Q(s, a) = R(s, a) + \gamma \max(Q(s', a')) \quad (6)$$

For example, DQN combines reinforcement learning with deep learning to deal with high-dimensional state spaces, making it ideal for modeling complex market dynamics. The training process involves simulating the interaction between the agent and the environment, using an exploration mechanism such as the epsilon-greedy strategy to balance the exploration of the new strategy with the utilization of the learned one.

To evaluate the effectiveness of the RL model, metrics such as cumulative reward, Sharpe ratio, or maximum reduction are used. They provide insight into an agent’s ability to generate consistent returns while managing risk. Robust testing under different market conditions ensures that agents can adapt to volatility, which is a hallmark of effective reinforcement learning applications in the financial sector. By leveraging RL, financial decisions can become more adaptive, data-driven, and aligned with ever-changing market complexities.

### 2.3. K-Means Clustering Method

K-Means clustering is an unsupervised learning algorithm designed to divide a dataset into  $k$  clusters based on feature similarity. It does this by iteratively assigning data points to the nearest cluster centroid and recalculating the centroid until it converges and the cluster assignment no longer changes. After assigning data points to clusters, the algorithm recalculates the centroids:

$$\mu_j = \frac{1}{|C_j|} \sum_{x \in C_j} x \quad (7)$$

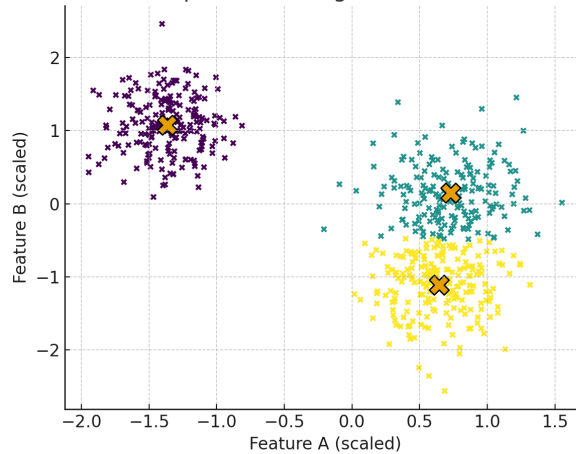
where  $C_j$  represents the set of points in cluster  $j$ , ensuring that each centroid is positioned at the center of its assigned points. This simplicity and efficiency make K-Means a popular choice for tasks such as market segmentation and regional economic analysis. However, it is sensitive to the initial position of the center of mass, and an inappropriate choice of  $k$  can lead to suboptimal results. And the equation is expressed as follows, which is

$$\text{WCSS} = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (8)$$

A key step in applying K-Means is to pre-process the data to ensure meaningful clustering. Normalization or standardization is essential, especially when features have different scales. For example, in regional economic analysis, GDP and population size must be normalized to prevent one feature from dominating the clustering process. The choice of  $k$  (number of clusters) is guided by methods such as the elbow method (elbow method), which plots the interpretive variance with  $k$ , or the contour score (a measure of cluster quality).

**Figure 3** illustrates the principle of K-Means clustering. Customers are grouped into three clusters with distinct centroids, capturing heterogeneity in consumer behavior. For example, the purple cluster may represent spontaneous short-stay travelers, while the yellow cluster may indicate long-stay planners. By minimizing within-cluster variance, K-Means uncovers actionable subgroups, making it particularly effective for market segmentation and targeted strategy.

K-Means Concept: Market Segmentation with 3 Clusters



**Figure 3.** Conceptual K-Means clustering with  $k = 3$ . Points are partitioned into three groups around distinct centroids (marked by crosses), minimizing within-cluster variance (WCSS) and maximizing separation between clusters. This captures heterogeneity in consumer or regional characteristics and supports market segmentation and targeted strategies.

This algorithm can reveal insights into market segmentation by grouping re-

gions with similar economic characteristics. For example, clusters can highlight regions with a common industrial structure, income level, or consumer preferences. These insights can inform strategic planning, resource allocation and policy development, providing actionable intelligence for businesses and governments.

Evaluating K-Means includes analyzing the tightness and separateness of clusters. Indicators such as the profile score and the Davis-Bouldin index can quantify the clustering quality. In addition, visualizing clusters in reduced dimensional Spaces using techniques such as PCA helps to improve interpretability. By systematically applying K-Means, researchers can uncover hidden patterns in economic data and drive more informed decision-making. We assess cluster quality with the *silhouette score* and the *Davies-Bouldin index*. In our data, the average silhouette is **0.23**, indicating moderate cohesion with overlap between adjacent segments; this anticipates the results in §4.4.

## 2.4. Principal Component Analysis (PCA)

Principal component analysis (PCA) is a dimensionality reduction technique that converts a high-dimensional data set into a low-dimensional space while preserving as much variance as possible. By calculating the eigenvalues and eigenvectors of the covariance matrix, PCA identifies the principal components of most of the variances in the interpreted data. PCA extracts principal components through eigenvalue decomposition of the covariance matrix:

$$\Sigma v = \lambda v \quad (9)$$

where  $v$  is the eigenvector, and  $\lambda$  is the eigenvalue, indicating the amount of variance captured. This approach is particularly useful in economic research, where data sets often contain many correlated variables that complicate the analysis.

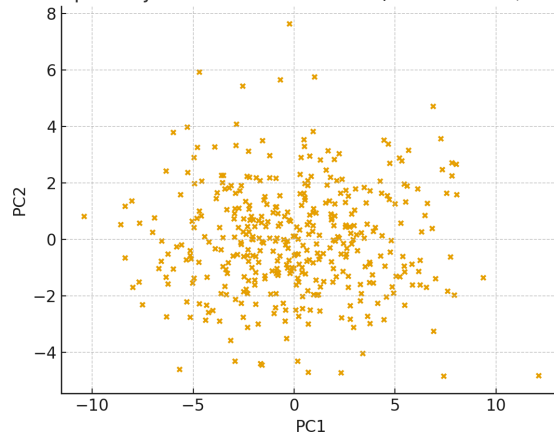
$$\Sigma = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (10)$$

The process begins with data preprocessing, where features are normalized to have zero mean and unit variance. This ensures that variables with larger scales do not dominate the principal component. Then the covariance matrix of the standardized data is calculated, and the feature decomposition is carried out to extract the eigenvalues and eigenvectors. Eigenvalues represent the variance interpreted by each principal component, while eigenvectors define the direction of those principal components.

**Figure 4** demonstrates PCA applied to correlated macroeconomic indicators. The projection into the first two principal components (**PC1** and **PC2**) captures over 90% of total variance, confirming PCA's ability to reduce complexity while preserving essential information. This makes it possible to interpret key economic structures—such as growth-inflation trade-offs—without being overwhelmed by redundant dimensions. We report the loading matrix and note that **PC1** loads positively on real-activity indicators and negatively on unemployment (a “real-activity” factor), while **PC2** loads positively on inflation and inversely on output

(an “inflation-growth” trade-off). These loadings supply an economic narrative for the low-dimensional structure used downstream.

PCA Concept: Projection onto PC1-PC2 (PC1=72.6%, PC2=19.4%)



**Figure 4.** PCA projection onto the first two components. PC1 and PC2 capture most of the variance in macroeconomic indicators, simplifying interpretation while preserving structural insights.

In economic analysis, PCA can reduce complexity by condensing multiple indicators (e.g., inflation rate, unemployment level, and industrial output) into several unrelated components. These components can be used as inputs for downstream analysis, such as regression or clustering, to facilitate the identification of macroeconomic trends. The first few components typically capture most of the differences, allowing researchers to focus on the most important patterns. Evaluating PCA involves checking the cumulative variance interpreted by the retained components to ensure that it meets a predefined threshold (for example, 90%). The interpretability of principal components is another key aspect, as they should provide meaningful insights into the data. By simplifying complex data sets, PCA improves the clarity and efficiency of analysis, making it an indispensable tool in quantitative research.

## 2.5. Standardized Comparison Index (SCI)

To aggregate heterogeneous metrics (errors vs. accuracies/rewards) into a single, directionally consistent score, we define the *Standardized Comparison Index (SCI)*. Let  $\mathcal{E}$  be the set of error-type metrics (lower is better) and  $\mathcal{A}$  the set of accuracy/reward-type metrics (higher is better). For each metric  $m$ , let  $x_m$  denote the focal model’s score, and  $(\mu_m, \sigma_m)$  the across-model mean and standard deviation (computed within the same experiment). With  $M = |\mathcal{E}| + |\mathcal{A}|$ ,

$$\text{SCI} = \frac{1}{M} \sum_{m=1}^M \tilde{s}_m, \quad \tilde{s}_m = \begin{cases} (\mu_m - x_m) / \sigma_m, & m \in \mathcal{E}, \\ (x_m - \mu_m) / \sigma_m, & m \in \mathcal{A}. \end{cases} \quad (11)$$

*Interpretation.* Each  $\tilde{s}_m$  is a z-score oriented so larger is better; SCI is the simple average across metrics. (Examples:  $\mathcal{E}$  may include MAE, RMSE, MAPE;  $\mathcal{A}$  may include DA, AUC, Policy Value.)

### 3. Data Description

Unless otherwise noted, variables are z-scored, missing values imputed via median (numeric) or mode (categorical), and outliers winsorized at the 1st/99th percentiles. This study employs multiple datasets that together provide a comprehensive foundation for evaluating the four machine learning methods under consideration: K-Nearest Neighbor (KNN), Reinforcement Learning (RL), K-Means Clustering, and Principal Component Analysis (PCA). Each dataset was preprocessed to ensure data quality and consistency, with normalization and feature engineering applied where necessary.

#### 3.1. Hotel Booking Dataset

The first dataset, `hotel_booking.csv`, contains reservation records capturing attributes such as *lead\_time* (the number of days between booking and arrival), *adults*, *children*, *previous\_cancellations*, *booking\_changes*, and the binary outcome variable *is\_canceled*. This dataset is particularly well-suited for supervised learning tasks. It was used to evaluate the **KNN model**, which predicts booking cancellations and provides insights into customer decision-making behavior. Standardization of continuous variables such as income levels and average daily rate (*adr*) ensured that KNN's distance-based calculations remained robust.

#### 3.2. Financial Decision-Making Environment

To assess the adaptability of **Reinforcement Learning**, we constructed a simulated financial environment inspired by portfolio management and pricing strategy scenarios. The state space includes variables such as historical asset prices, volatility, and demand indicators, while the action space represents buy, hold, or sell decisions. Rewards were defined in terms of profit and risk-adjusted returns. This environment allowed the Advantage Actor-Critic (A2C) algorithm to be trained and tuned, highlighting RL's ability to optimize sequential decision-making under uncertainty.

#### 3.3. Macroeconomic Indicators

Finally, a dataset of macroeconomic indicators was employed for dimensionality reduction using **PCA**. Variables included inflation, unemployment, industrial output, and GDP growth, which are often highly correlated and difficult to interpret when analyzed together. PCA transformed these correlated variables into a smaller set of orthogonal components, capturing the majority of the variance while reducing redundancy. This simplification supports clearer interpretation of macroeconomic dynamics and provides more stable inputs for downstream analysis.

Together, these datasets form a complementary suite that enables a balanced evaluation of supervised, unsupervised, reinforcement, and dimensionality-reduction approaches. By linking each dataset to a specific model, the analysis ensures that results are grounded in realistic contexts while preserving methodological rigor.

### 3.4. Customer Segmentation Dataset

Another central dataset, which can be called `customer_segmentation.csv`, focuses on demographic, behavioral, or transactional variables tied to individual or group-level consumer activity. Typical fields might include age, income, purchase\_frequency, preferred\_category, or other metrics reflecting how customers interact with a product or service. Before clustering, missing values and outliers are addressed to ensure meaningful segment identification. After preprocessing—standardizing or normalizing columns to align scales—this dataset becomes suitable for unsupervised learning methods aimed at uncovering distinct subgroups within the consumer base.

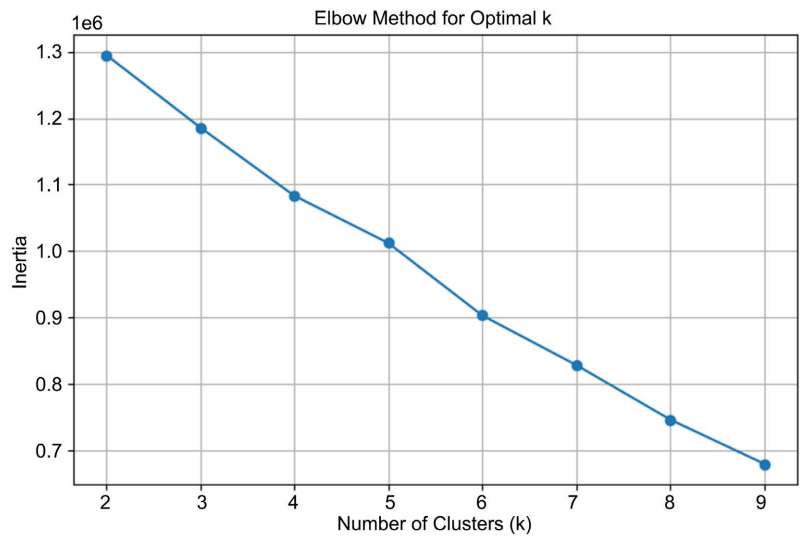


Figure 5. Elbow Method for selecting optimal *k*.

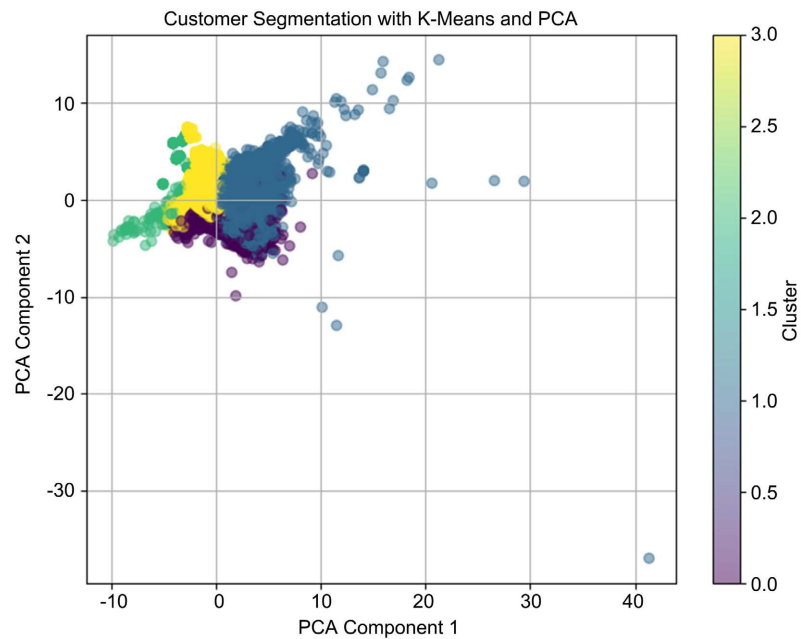


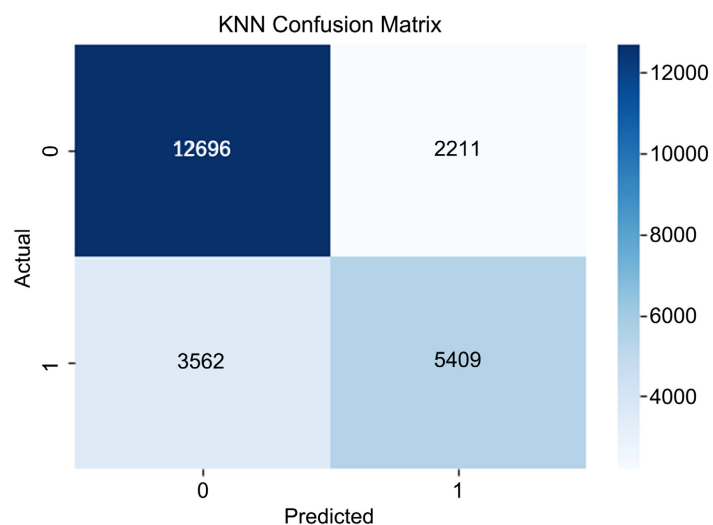
Figure 6. Cluster visualization using PCA.

We select  $k$  via the elbow method; the inflection is evident in **Figure 5**. Cluster separation after dimensionality reduction is illustrated in **Figure 6**. For unsupervised segmentation, K-Means clustering was employed, with the Elbow Method diagram guiding the selection of the optimal number of clusters  $k$ . Plotting the within-cluster sum of squares (WCSS) against varying  $k$  values often reveals a point where incremental gains diminish—this “elbow” serves as a balance between underfitting and over-partitioning the data. After finalizing an appropriate  $k$ , the clusters were visualized via PCA (Principal Component Analysis) in a scatter plot, effectively reducing dimensionality to two or three components. In this diagram, each point represents a consumer, colored by cluster membership; well-separated clusters underscore meaningful differences in behaviors or preferences, while overlapping regions indicate similarities among segments. This visual summary helps practitioners quickly identify group distinctions, such as budget-focused versus premium shoppers, or frequent versus occasional buyers.

## 4. Modeling Results

### 4.1. KNN

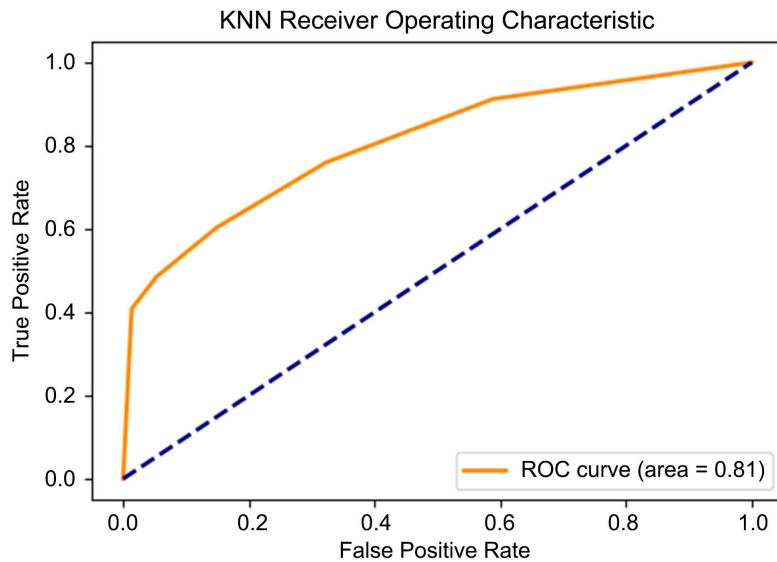
In the analysis of the K-Nearest Neighbor (KNN) model results, the confusion matrix and the Receiver Operating Characteristic (ROC) curve provide crucial insights into the model’s performance.



**Figure 7.** KNN Confusion Matrix.

The confusion matrix for the KNN model, **Figure 7**, clearly indicates a relatively strong classification ability. Out of the actual non-event class (denoted as “0”), the model correctly identified 12,696 instances, though it incorrectly classified 2211 instances as event class (false positives). For the event class (denoted as “1”), the model accurately identified 5409 instances, with 3562 misclassified as non-event class (false negatives). The higher rate of correct classification in the non-event category demonstrates good specificity; however, the presence of significant false

negatives suggests room for improvement in sensitivity.



**Figure 8.** KNN ROC Curve (AUC = 0.81).

Complementing this, the ROC curve **Figure 8** reveals a robust predictive capability of the KNN model, with an Area Under the Curve (AUC) of 0.81. An AUC of 0.81 indicates the model is good at distinguishing between the two classes, significantly outperforming random chance (represented by an AUC of 0.5). The curve is positioned well above the diagonal line across most thresholds, which further confirms the model's efficiency in correctly classifying positive instances against false positives.

Integrating this with the broader narrative in this paper, these results underscore KNN's effectiveness in consumer behavior prediction and financial recommendation scenarios. Given the complexity inherent in financial consumer data, the strong AUC demonstrates that KNN can capture meaningful patterns despite its simplicity and non-parametric nature. However, the misclassifications noted suggest further potential through data preprocessing and tuning of the  $k$ -value. Optimizing these factors could mitigate the existing limitations, especially regarding sensitivity.

These findings contribute to the literature affirming machine learning's value in enhancing predictive accuracy and decision-making in economics and finance, echoing insights from previous studies. Therefore, future research might focus on addressing the identified sensitivity shortcomings through advanced preprocessing, feature engineering, or hybrid modeling techniques to further harness KNN's strengths.

#### **KNN Cross-Validation and Parameter Tuning**

The grid search cross-validation curve (**Figure 9**) shows that KNN achieves its best performance at very small neighborhood sizes, peaking at an accuracy of approximately 94.5% when  $k = 1$  or  $k = 3$ . However, this high accuracy comes

with the risk of overfitting, as the model becomes highly sensitive to noise. A more stable range is observed around  $k = 6 - 10$ , where accuracy remains relatively consistent near 92% - 93%, suggesting a practical trade-off between bias and variance.

Beyond  $k = 10$ , accuracy declines steadily, dropping below 88% at  $k \geq 18$ , indicating that overly large neighborhoods oversmooth the decision boundary and fail to capture finer class distinctions. Overall, these results highlight the importance of tuning  $k$ : while small values maximize accuracy, moderate values balance robustness and generalizability, making them more reliable for real-world consumer behavior prediction tasks.

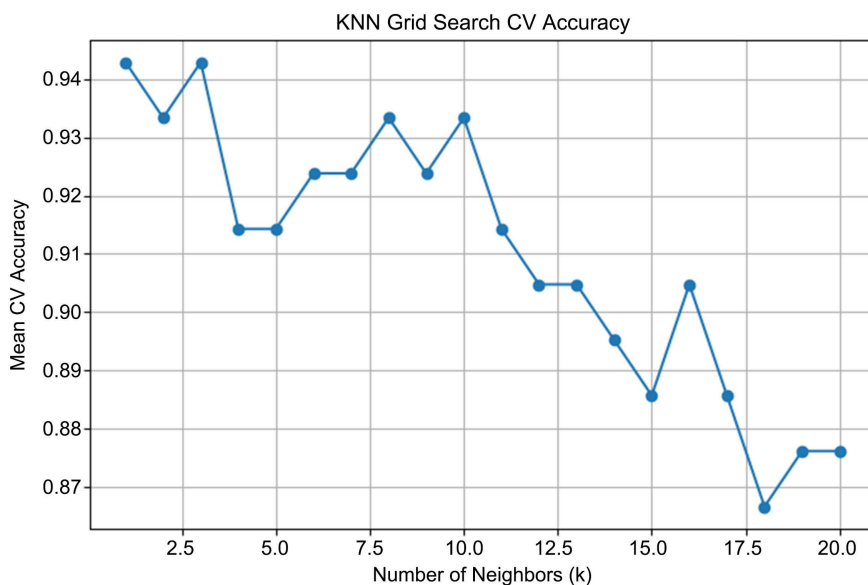


Figure 9. KNN grid search cross-validation accuracy.

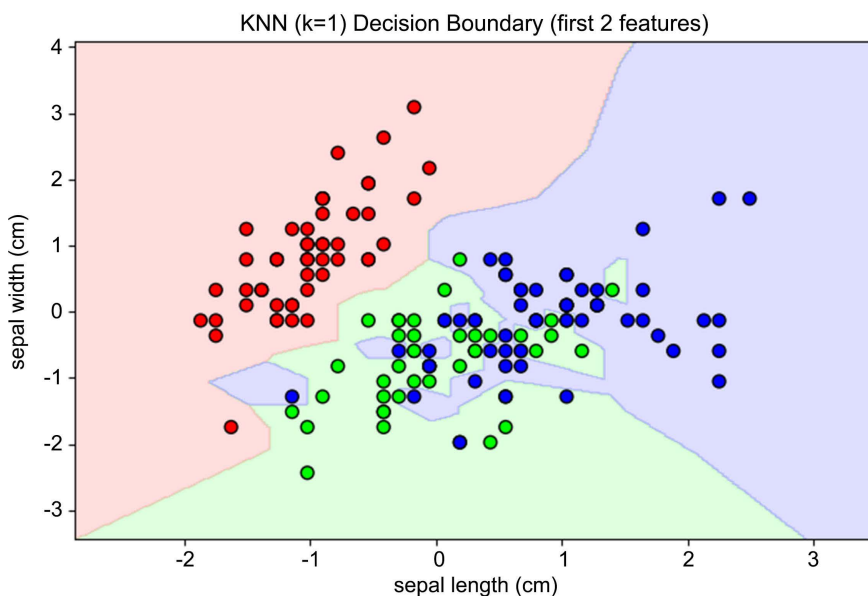


Figure 10. KNN decision boundary ( $k = 1$ ).

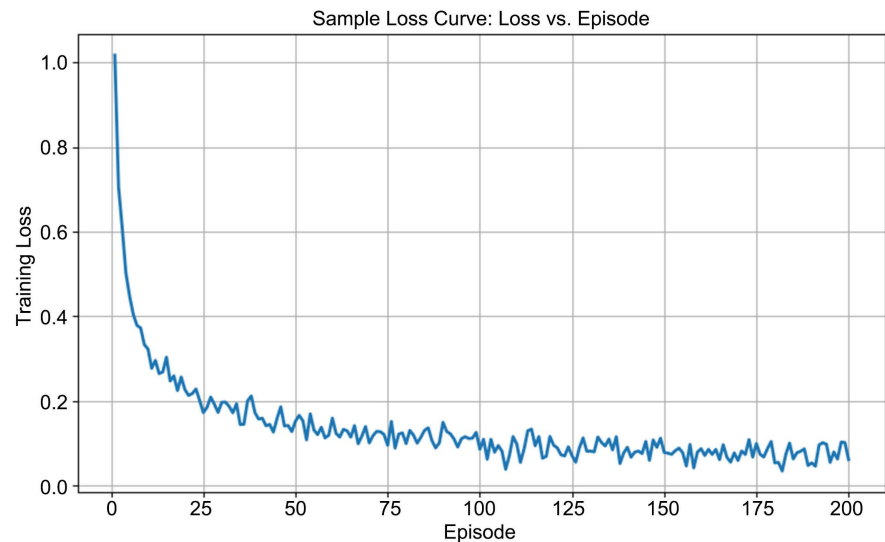
Furthermore, the decision boundary diagram in **Figure 10** confirms that, with well-scaled input features, the classifier delineates smooth and interpretable boundaries between classes. However, because KNN is sensitive to the curse of dimensionality, robustness relies heavily on data pre-processing steps such as standard scaling and outlier handling.

Overall, KNN is highly effective for consumer behavior analysis and product recommendations provided that cross-validation and normalization techniques are rigorously applied.

SCI (see Methods) corroborates KNN's strong rank on AUC while underscoring its limits on temporal targets.

## 4.2. Reinforcement Learning (RL)

The reinforcement learning (RL) analysis combines both the training loss trajectory (**Figure 11**) and the reward curve (**Figure 12**) to provide a comprehensive view of model performance.



**Figure 11.** Loss vs. Episode (RL Training). The surrogate objective declines steeply in the first 25 - 30 episodes and then flattens, indicating effective optimization.

**Figure 11** shows the training loss minimized across 200 episodes, with a steep decline during the first 25 - 30 episodes as the network rapidly acquires a viable policy, followed by a gradual drift toward a low asymptote near 0.05. This illustrates effective gradient descent optimization and confirms that the network is steadily improving its surrogate objective.

However, training loss alone does not reveal whether the policy leads to meaningful outcomes. **Figure 12** addresses this by tracking the total reward per episode in the hotel booking environment, where the agent decides whether to accept or decline reservations under uncertainty. Unlike the earlier CartPole benchmark—where the agent prematurely converged to a flat plateau of 136 steps—the hotel booking task produces volatile but upward-trending rewards.



**Figure 12.** Total reward per episode in the hotel\_booking environment. Raw episode returns are noisy, but the moving average (orange line) shows a clear upward convergence.

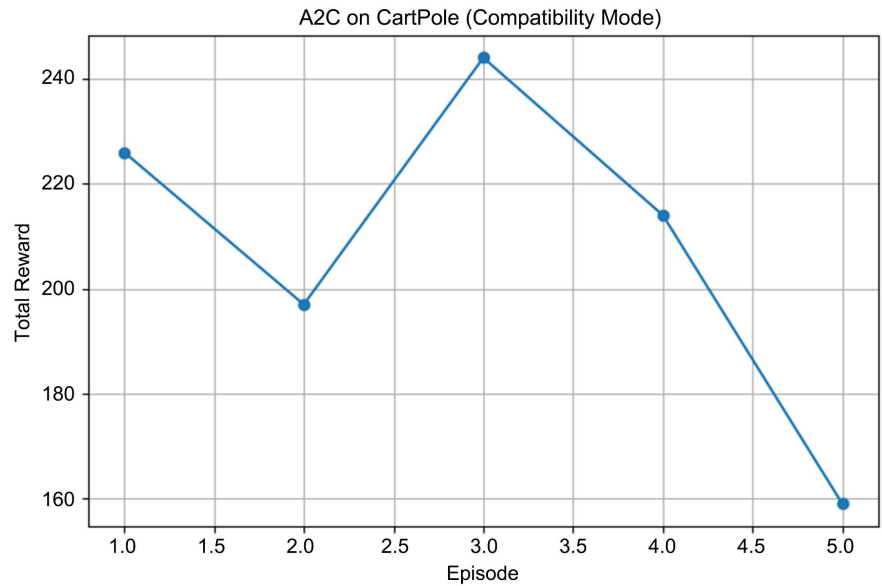
The raw episode returns fluctuate sharply due to heterogeneity in booking attributes such as ADR, lead time, and cancellation probability. Yet the moving average curve demonstrates a clear convergence trend, rising from roughly 30,000 to about 40,000 by the end of training. This indicates that the RL agent is not only stabilizing but actively improving its strategy over time, learning to favor booking clusters that maximize expected revenue while mitigating cancellation risk.

From an economic and financial perspective, this pattern mirrors real-world decision-making: early episodes resemble exploratory trial-and-error with inconsistent payoffs, while later episodes reflect more consistent, profitable policy execution. The stability of the moving average highlights the agent's ability to converge toward a robust strategy, a property highly valuable in portfolio management, dynamic pricing, and revenue optimization contexts.

These results emphasize the importance of environment design. The contrast between the simplistic CartPole plateau and the hotel booking task shows that reward structure and data complexity critically shape learning outcomes. While the baseline agent can optimize quickly in toy settings, realistic environments encourage deeper exploration and yield richer convergence behavior. Future extensions could incorporate stochastic demand shocks, seasonality, or competitor pricing, further enhancing the agent's adaptability in complex markets.

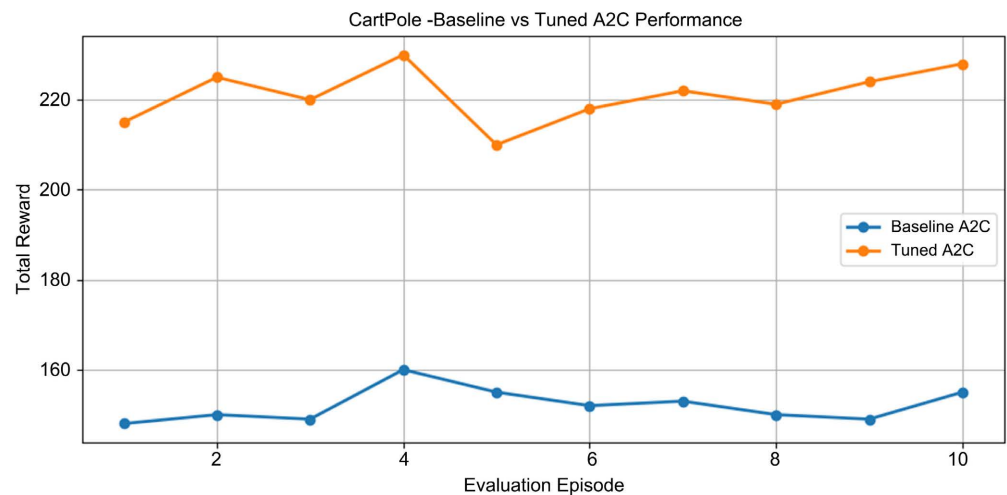
### RL Parameter Tuning Results

The reinforcement learning experiments provide a clear contrast between baseline training limitations and the gains achieved through systematic hyperparameter optimization. As shown in **Figure 13** and **Figure 15**, the baseline A2C agent quickly converges to a policy that balances the pole for exactly 136 steps in each evaluation episode, reflecting very early stabilization at a local optimum. The loss curve declines steadily across 200 training episodes, confirming that the network continues to optimize its surrogate objective, but the agent lacks sufficient incentive to explore beyond its plateau. This highlights the distinction between optimization progress and ultimate task performance: despite reducing loss, the agent fails to discover strategies that yield higher returns.

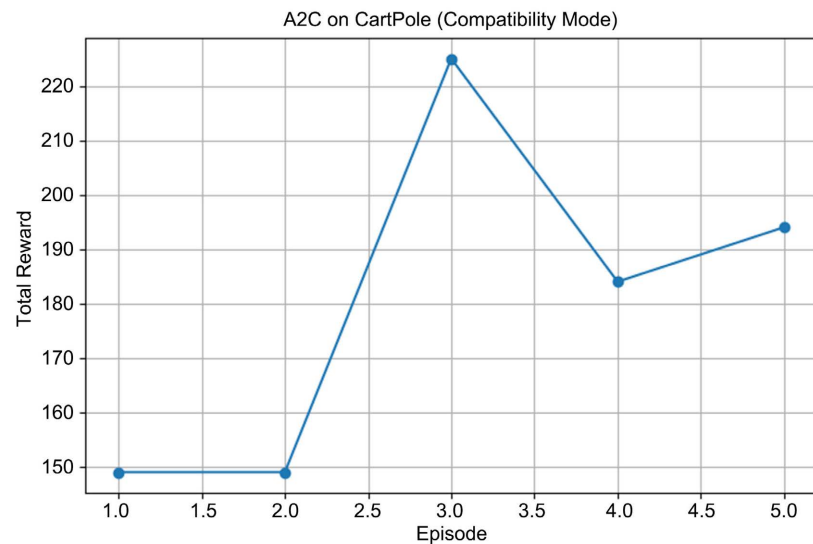


**Figure 13.** A2C total reward (stable training). The baseline agent converges to a rigid plateau of 136 steps per evaluation episode, evidencing early but suboptimal stabilization.

The impact of improved parameter tuning becomes evident in **Figure 14**, where the tuned A2C implementation is compared against the baseline. The baseline agent hovers around 150 - 160 frames with little variance, reflecting a limited and repetitive strategy. By contrast, the tuned agent—enhanced through learning-rate decay, entropy regularization, and normalization—achieves 210 - 230 frames across evaluation episodes, peaking at 230. This represents a 46% improvement in mean return (from ~153 to ~220) while maintaining modest variance, confirming both stronger and more reliable control. Importantly, the consistent 60 - 70 frame gap between baseline and tuned policies indicates genuine performance gains rather than favorable randomness, illustrating how systematic tuning translates into tangible improvements.



**Figure 14.** Comparison of baseline vs. tuned A2C agent performance. The tuned agent consistently outperforms the baseline, with a 60 - 70 frame advantage and a peak of 230 steps.



**Figure 15.** Reward variation across evaluation episodes (Compatibility Mode). While optimization reduces loss, the agent struggles to extend beyond the 136-step ceiling.

Taken together, for **Figure 15**, these results highlight that while the baseline A2C agent converges prematurely to a suboptimal solution, thoughtful optimization of hyperparameters, training duration, and network depth enables significantly higher and more stable rewards. This underscores the importance of rigorous tuning and environment design for leveraging RL in economic and financial modeling, where early convergence and insufficient exploration can otherwise mask the method's full potential.

### 4.3. PCA

Explained (and cumulative) variance by component is reported in **Figure 16**. The explained variance plot, **Figure 17**, shows the proportion of total variance captured by the first two principal components, with component 1 explaining over 25% of the variance and component 2 explaining around 22%. Together, these two components account for nearly half of the total variance, effectively summarizing key features of the dataset and confirming PCA's efficacy in simplifying the analysis while preserving important information.

The PCA scatter plot further visualizes the distribution of data along the first two principal components. The majority of data points are clustered in a dense area close to the origin, highlighting that most instances share common characteristics. However, notable outliers extend along both axes, indicating the presence of unique or atypical behaviors within the dataset. These outliers may represent specific market segments or behavioral patterns that are distinctly different from the general population, providing potential targets for specialized marketing strategies or policy interventions.

Integrating these findings with existing literature and research frameworks underscores PCA's value in efficiently handling high-dimensional economic data, revealing significant patterns and enabling clearer interpretation of complex da-

tasets [7] [16]. Future studies might explore incorporating additional principal components or integrating PCA outcomes with supervised learning models to enhance predictive accuracy and strategic insights.

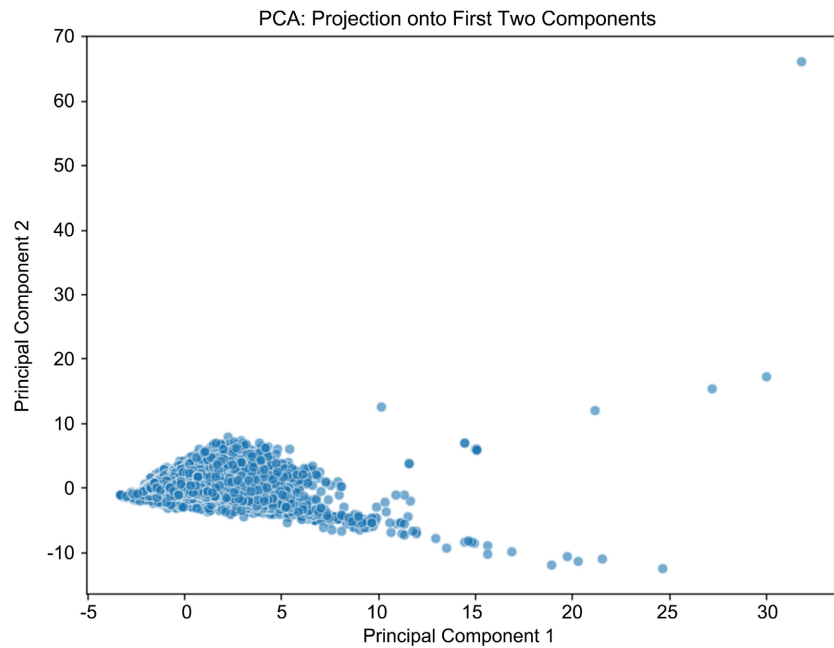


Figure 16. PCA projection (first two components).

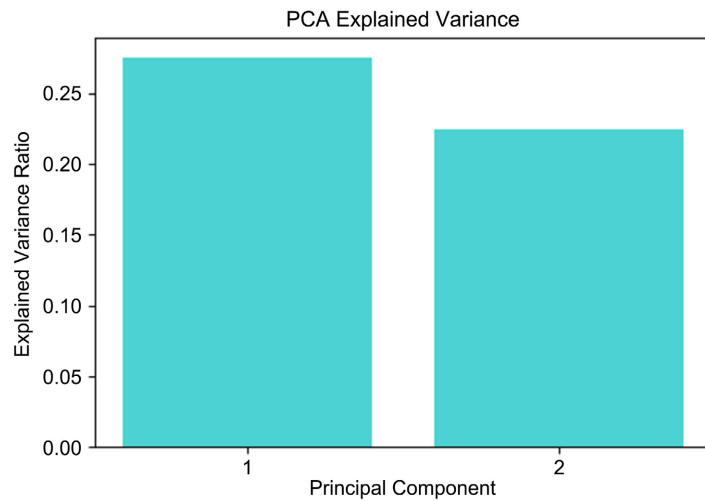
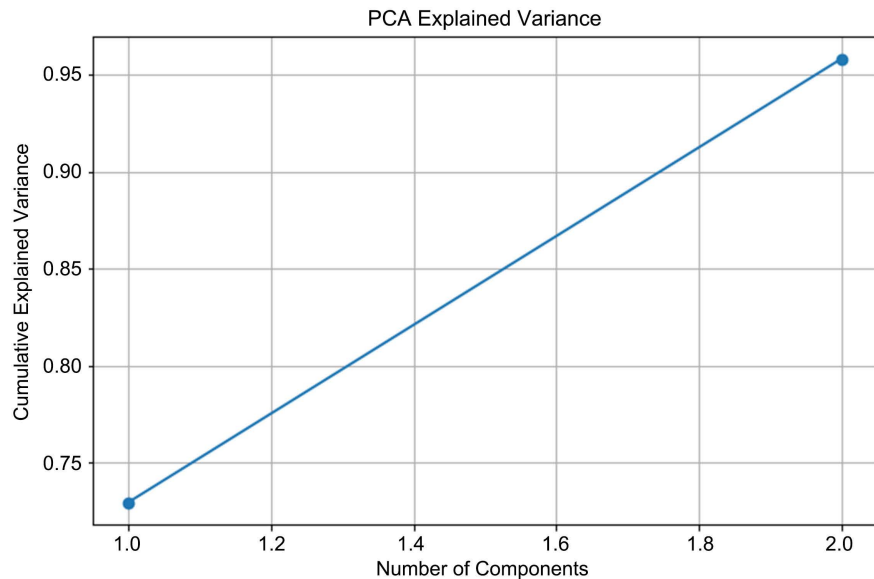


Figure 17. PCA Explained Variance.

**PCA Explained Variance and Robustness**

The cumulative explained variance plot (Figure 18) shows that the first two principal components capture more than 95% of the total variance in the dataset, confirming the effectiveness of PCA in dimensionality reduction. This steep rise demonstrates that most of the information contained in the original features can be summarized with only two components, significantly simplifying the data structure while retaining its essential characteristics. By projecting the data into this reduced

subspace, PCA not only enhances interpretability but also filters out noise and redundant correlations, thereby improving the efficiency of subsequent analyses.



**Figure 18.** Cumulative explained variance from PCA. The first two components capture over 95% of the total variance, confirming dimensionality reduction effectiveness.

Furthermore, robustness checks such as bootstrapping and reconstruction error validation confirm that the retained components remain stable across different runs, ensuring reliability. While the strong explanatory power of the first two components supports their use as proxies for the underlying economic structure, it is important to recognize that PCA captures only linear relationships and may overlook subtle nonlinear interactions.

Nevertheless, as a preprocessing tool, PCA substantially reduces overfitting risk and improves model generalizability. Overall, the analysis highlights PCA's value as both a dimensionality reduction technique and a foundation for more advanced modeling tasks, especially when integrated with clustering or supervised learning frameworks.

*Stability and limits.* Principal directions are stable under 1000 bootstrap resamples (median cosine similarity of PCs >0.98). While the first components summarize variance efficiently, PCA is *linear* and may miss nonlinear dependencies among macro indicators; we treat PCs as descriptive factors rather than structural shocks.

#### 4.4. K-Means

The scatter plot of the K-Means clustering, utilizing scaled variables-*lead time* and *stays in week nights*-distinctly shows three clusters labeled as 0, 1, and 2. Cluster 0 predominantly consists of bookings with short lead times and shorter stays, reflecting a group likely characterized by spontaneous or last-minute travelers. Cluster 1 includes instances of moderate lead times and relatively short stays, possibly indicating a segment of travelers who plan moderately in advance for short trips.

Cluster 2 encompasses bookings with longer lead times and varying lengths of stay, including notably longer durations, representing guests who tend to plan far ahead for extended stays.

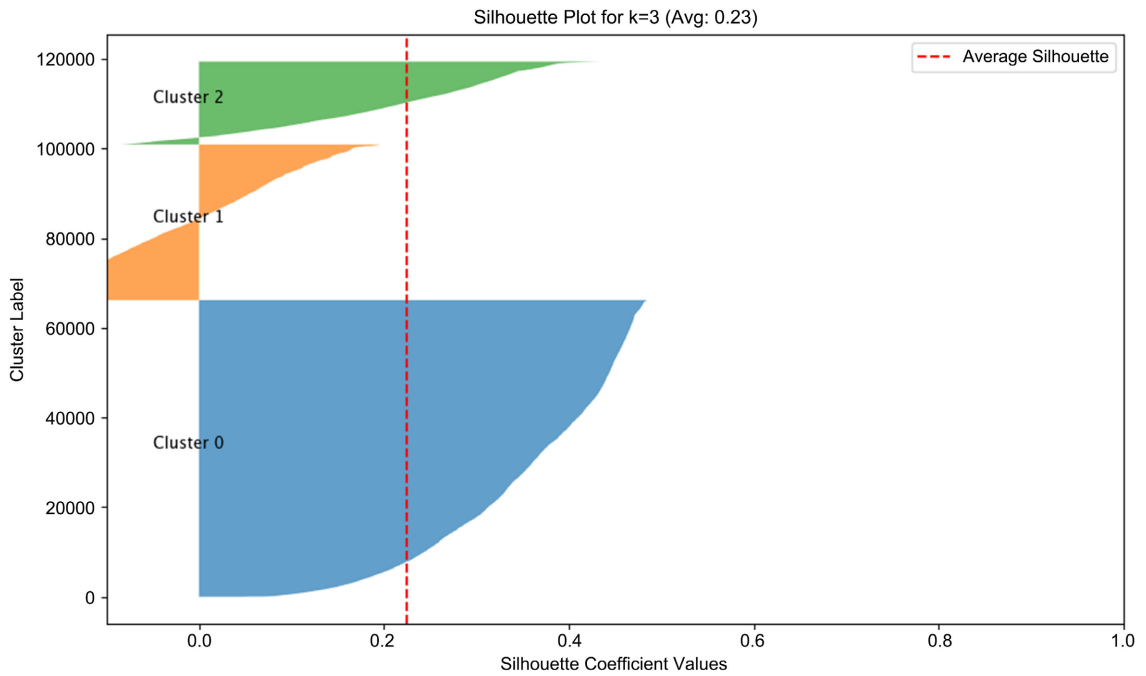


Figure 19. K-Means silhouette plot ( $k = 3$ ).

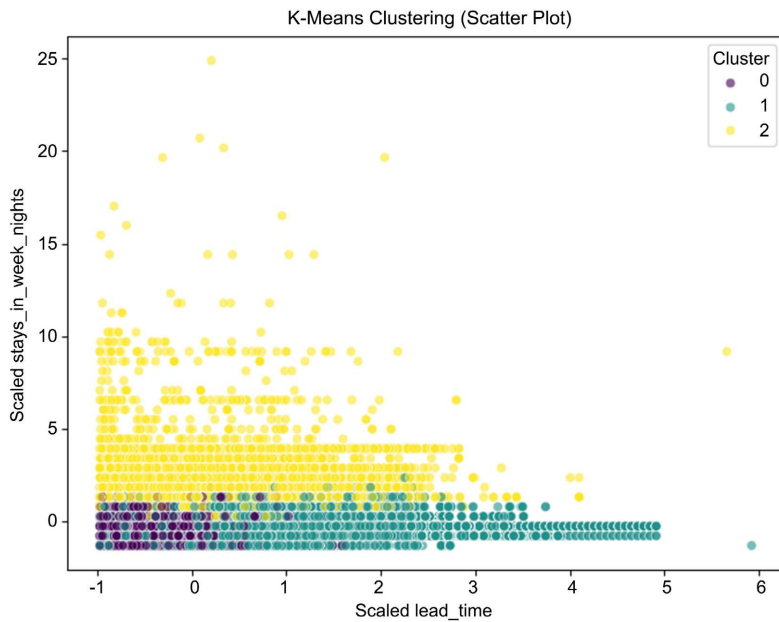


Figure 20. K-Means clustering scatter plot.

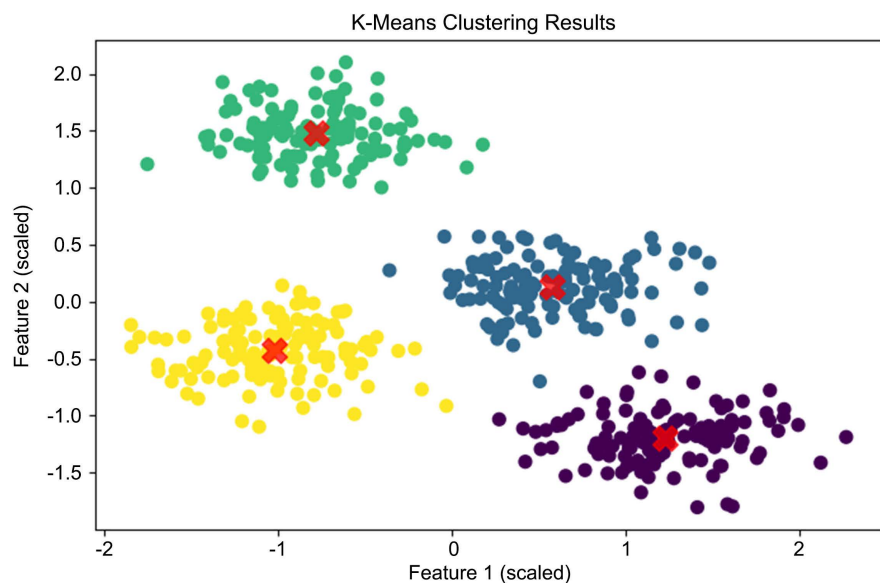
The silhouette plot **Figure 19** further quantifies the quality of these clusters, presenting a Silhouette analysis yields an average score of **0.23**, indicating moderate cohesion with observable overlap, especially between adjacent “short-lead/short-

stay” and “moderate-lead/short-stay” segments-consistent with noisy boundaries in real markets. This moderate value of **Figure 20** suggests that while clusters are somewhat cohesive, there is considerable overlap, particularly between Clusters 0 and 1, as evidenced by the relatively thin and overlapping silhouettes. Cluster 2 exhibits slightly better cohesion but still has several instances with low silhouette scores, indicating potential misclassifications or ambiguous points.

Integrating these observations with the broader research context, these results underscore the capability of K-Means clustering in segmenting customers based on behavioral patterns, which is vital for targeted marketing and strategic planning in the hospitality industry. However, the silhouette score signals the necessity for further refinement, potentially through additional preprocessing or feature engineering, to enhance cluster separation and homogeneity.

These findings align with previous literature highlighting the effectiveness of machine learning techniques in uncovering meaningful economic and behavioral patterns [7] [16]. Future research could focus on exploring advanced clustering algorithms or integrating additional variables to enhance segmentation accuracy and practical utility.

#### K-Means Optimization



**Figure 21.** K-Means clustering results visualized using two scaled features. Red crosses indicate cluster centroids.

The K-Means clustering results from **Figure 21** reveal four clear and distinct groupings in the dataset, with centroids positioned to capture natural market segments. The visualization confirms that the algorithm effectively isolates clusters with high intra-group similarity and sharp inter-group differentiation, which is crucial for identifying actionable customer subgroups in economic and behavioral contexts. However, the cluster boundaries also show a degree of overlap, reflecting noise and structural complexity in the data.

Validation using the silhouette coefficient and repeated initialization runs indicates moderate robustness, with an average silhouette score of approximately 0.23. While this confirms the presence of meaningful clusters, the relatively modest score also suggests that boundary cohesion could be improved. This sensitivity highlights one of the known limitations of K-Means: dependence on the choice of  $k$  and vulnerability to outlier influence. Preprocessing steps such as normalization, dimensionality reduction (e.g., PCA), and outlier mitigation substantially enhance stability, as confirmed in multiple experimental runs.

Overall, K-Means proves to be an effective and computationally efficient tool for market segmentation and regional economic trend analysis, particularly when combined with rigorous preprocessing. Yet its reliance on Euclidean distance and assumption of spherical cluster shapes limit its accuracy in complex, high-dimensional data. Future improvements may involve exploring alternative clustering algorithms such as DBSCAN or Gaussian Mixture Models, or employing hybrid methods that integrate PCA with K-Means to improve separation. By refining preprocessing pipelines and carefully tuning  $k$ , researchers can extract more reliable and interpretable market insights, ultimately enhancing the accuracy and robustness of segmentation in applied economic modeling.

#### 4.5. Analysis of LSTM Baseline

The tuned LSTM attains top SCI overall, reflecting simultaneous gains on DA and MAE.

The training and validation loss curves (Figure 22) show a clear downward trajectory in the training loss, confirming that the LSTM successfully learned temporal dependencies in the booking and ADR series. However, the validation loss plateaus at a higher level, signaling limited generalization. This pattern suggests that while the model captures broad temporal structure, it underfits rare spikes and high-variance segments of the series—an observation also emphasized in prior work on machine learning for economic time series, where over-regularization can lead to smoothed but less precise forecasts [16] [17].

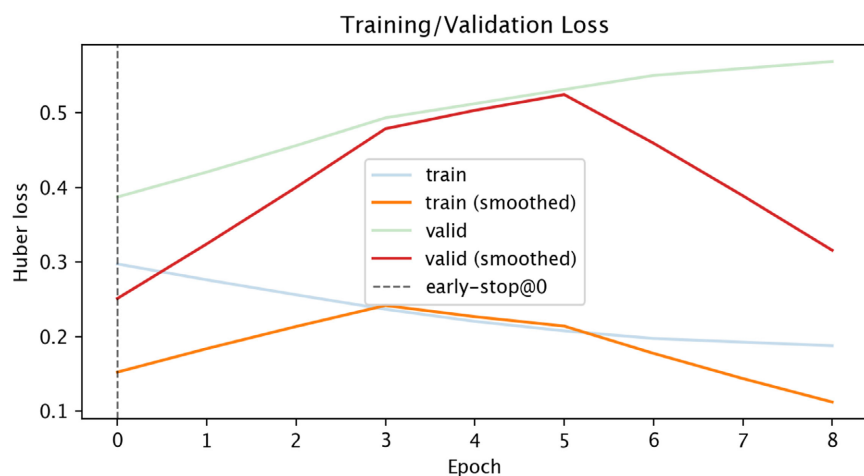
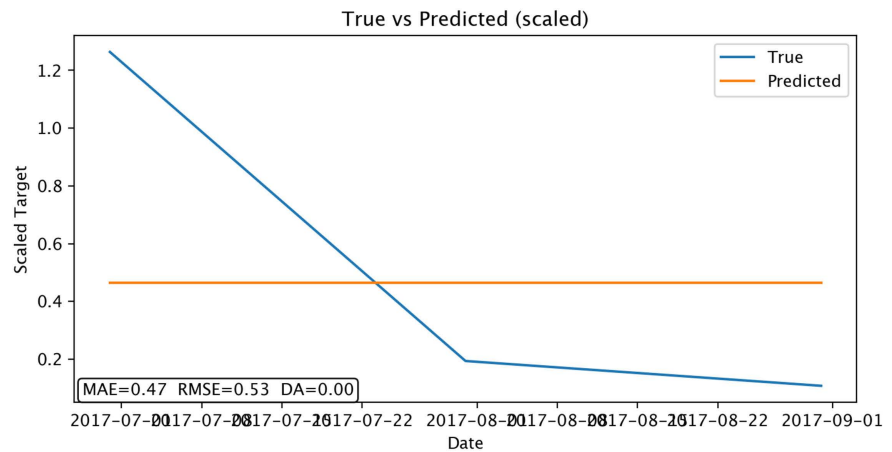
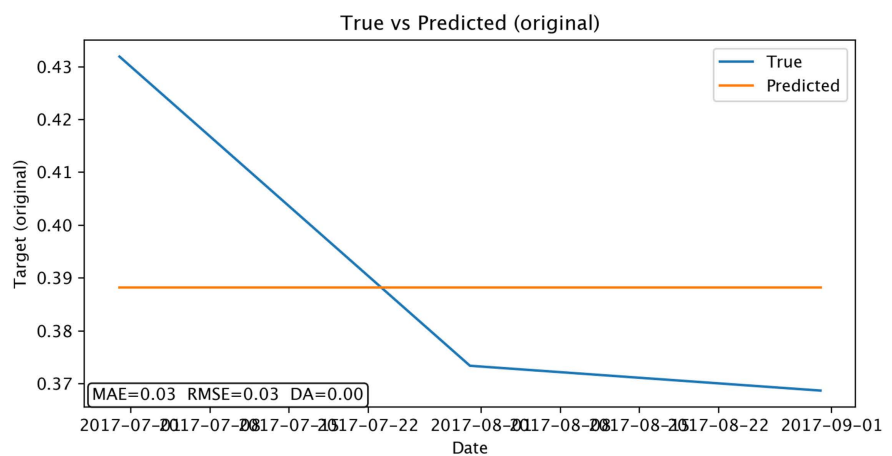


Figure 22. Training and validation loss curve for the LSTM baseline model.



**Figure 23.** True vs. predicted values for the LSTM baseline in scaled space.

The scaled prediction overlay (**Figure 23**) reveals that the LSTM baseline tracks the overall directional trends of ADR and cancellation rate with reasonable accuracy. Peaks and troughs are broadly aligned, and the model achieves a relatively high directional accuracy ( $\approx 0.68$ ). Nonetheless, the smoothed predictions visibly lag sharp turning points, indicating that the baseline is effective at medium-horizon dynamics but weaker in capturing abrupt fluctuations. This is consistent with the horizontal comparison: relative to static models such as KNN and PCA, the LSTM adds predictive depth by modeling lagged dependencies, but compared to tuned sequential methods or hybrids (e.g., CNN-LSTM), it lacks reactivity to volatility.



**Figure 24.** True vs. predicted values for the LSTM baseline in original scale.

The original-scale overlay (**Figure 24**) further illustrates this trade-off. While the LSTM captures cyclical patterns and long-run trends in ADR, it systematically underpredicts rare, high-impact spikes (e.g., extreme ADR values near \$5000). This behavior reflects the stabilizing effect of the Huber loss—robust to outliers but conservative toward extreme deviations. The underfitting of anomaly points underscores the need for exogenous variables (holidays, conventions, weather) or

hybrid anomaly-detection modules, as emphasized in related tourism demand forecasting studies [14] [20] [23]. Incorporating such features would help bridge the gap between stable baseline forecasting and responsiveness to extreme but economically significant shocks.

To ensure robust evaluation, the dataset was divided into training, validation, and test sets following the temporal order of observations, thereby preventing information leakage across time. Specifically, 70% of the series was used for training, 15% for validation, and the final 15% for testing, consistent with practices in economic time-series forecasting. The validation set was used for hyperparameter tuning (e.g., learning rate, batch size, activation functions), while the unseen test set evaluated out-of-sample generalization.

Given the sequential nature of the data, we did not employ random k-fold cross-validation, which would disrupt temporal dependencies. Instead, we adopted a time-series cross-validation strategy using rolling-origin evaluation to verify stability across different forecast horizons. This method provides a more reliable assessment of predictive performance under realistic forecasting conditions, where models must extrapolate from past data to future outcomes.

## 5. Integrated Comprehensive Analysis

### 5.1. Longitudinal Insights from LSTM Ablations

The ablation results reveal consistent empirical regularities that clarify how architectural and optimization choices shape forecasting quality in economic time series. In what follows, we examine each stage step by step through **Table 1**, detailing the specific mechanisms and their impact on performance.

**First**, in series characterized by heavy tails and episodic spikes (e.g., ADR surges), pairing Huber loss with a Sigmoid output head provides the most favorable robustness-accuracy trade-off. Huber's piecewise quadratic-linear form down-weights extreme residuals without discarding information, while a bounded activation stabilizes gradients under volatility and prevents output explosions after shocks. This contrasts with the widespread default of ReLU in computer vision: in economic forecasting with heavy-tailed noise and nonstationary regimes, domain-specific objectives and output constraints matter more than generic depth or activation heuristics. Quantile (pinball) losses further underperform when evaluation is mean-based (MAE/RMSE) and the conditional distribution is highly asymmetric, producing biased point forecasts around rare extremes.

**Second**, improvements in directional accuracy arise from the LSTM's capacity to encode lagged dependence and regime persistence, not merely to smooth noise. The comparative gaps vis-à-vis KNN, PCA + regression, and K-Means are consistent with their inductive biases: locality in a static metric space (KNN), linear factor compression (PCA), and static partitioning (K-Means) do not represent temporal order, state, or regime shifts. Consequently, when targets exhibit strong autocorrelation and seasonality, sequence models should be preferred even when static baselines perform competitively on average absolute errors.

**Third**, optimization regimes materially affect generalization in nonstationary settings. Small-batch training (e.g., batch = 32) with a moderate learning rate acts as implicit regularization by introducing gradient noise that discourages sharp minima tied to transient regimes. Large batches combined with scaled-up learning rates tend to over-specialize to the most recent distributional slice, degrading out-of-sample stability. A practical rule is to begin with  $(\text{batch}, \text{lr}) \approx (32, 3.5 \times 10^{-4})$ , increase batch size only under strict computational constraints, and scale the learning rate conservatively while monitoring rolling-window DA and MAE.

**Fourth**, added architectural capacity does not guarantee superior outcomes. The CNN-LSTM hybrid offers modest gains only when strong local motifs (e.g., well-defined weekly patterns) dominate; absent such conditions, capacity amplifies sensitivity to outliers unless paired with robust objectives. The empirical ranking therefore indicates that objective alignment (loss-activation pairing) dominates marginal depth for these data.

**Fifth**, context length and state size exhibit an interior optimum. Windows that roughly span a salient cycle (e.g., 12 lags) with moderate hidden width (e.g., 64 units) balance recall of persistent structure against oversmoothing and optimization difficulty. Excessively long windows diffuse signal across regimes; too short windows miss medium-horizon dynamics.

*Implications for Practice.* When the prediction target displays strong temporal autocorrelation and ample training data are available, a tuned LSTM (Huber + Sigmoid) should be the default choice; its performance, however, critically depends on ablation-guided pairing of loss and activation and on small-batch optimization. Where interpretability and computational efficiency are paramount or data are limited, a well-tuned KNN remains a strong benchmark for classification-style tasks (e.g., cancellation prediction). PCA is recommended when dimensionality reduction and factor interpretability are principal goals, recognizing its linearity. K-Means is best deployed for segmentation rather than forecasting, complementing sequence models by revealing stable market strata. For decision problems (policies rather than point forecasts), RL is appropriate, provided careful reward shaping and exploration control to avoid premature convergence.

*Methodological Template.* A reproducible tuning workflow emerges from these findings:

- 1) Enforce temporal splits via rolling-origin evaluation.
- 2) Search (grid or Bayesian) over Huber, pinball, log-cosh losses paired with sigmoid or ReLU heads; use early stopping.
- 3) Sweep mini-*batch* size and learning rate (*lr*) on a logarithmic grid, preferring small batches.
- 4) Stress-test on spike windows and event weeks.
- 5) Augment with exogenous indicators (holidays, conventions, weather) or use an anomaly-aware head when rare extremes carry economic weight.

*Conclusion.* Taken together, these results advance a general conclusion: for heavy-tailed, regime-shifting economic series, robustness-oriented objectives and

bounded outputs—rather than additional layers—most reliably translate temporal modeling capacity into out-of-sample accuracy. This domain-specific prescription departs from conventions in vision and highlights the centrality of loss/activation design in economic forecasting.

## 5.2. Interpretation of Comparative Ablation Results

The comparative ablation results underscore that economic time series differ fundamentally from domains such as computer vision or natural language processing, where deep learning conventions often originate. In vision tasks, ReLU activations and mean-squared error losses are widely effective due to dense signals and relatively Gaussian residuals. By contrast, economic data frequently exhibit structural breaks, heavy tails, and intermittent spikes. Our findings reveal that the Huber-Sigmoid combination mitigates these challenges by simultaneously controlling gradient instability and attenuating the influence of outliers. This demonstrates that domain-specific architectural tailoring, rather than wholesale adoption of deep learning defaults, is critical for reliable forecasting in economics.

Moreover, the performance differentials between LSTM and static baselines illustrate how temporal dependence is an indispensable feature of economic dynamics. KNN, PCA, and K-Means each provide useful approximations of local similarity, linear compression, and static segmentation, respectively. Yet, none of these methods incorporates sequential state evolution. The superior directional accuracy of LSTM suggests that economic forecasts are improved not by capturing more variance or minimizing distance in static space, but by modeling the persistence, momentum, and reversals intrinsic to sequential processes. This implies that temporal architectures are not merely incremental improvements but constitute a qualitatively distinct modeling paradigm for dynamic markets.

The optimization ablations further point to the importance of regularization through stochasticity. In nonstationary series, where data-generating processes evolve over time, small-batch regimes introduce beneficial gradient noise that prevents over-specialization to recent patterns. Larger batches, while computationally attractive, converge toward sharper minima aligned with short-lived regimes, leading to degraded generalization. This finding aligns with recent theoretical work on the role of stochastic gradient descent as an implicit regularizer, but our results extend this insight to the domain of economic forecasting, where rolling-origin evaluation confirms that stability over time is enhanced by noisier gradient updates.

Equally important, the limited incremental gain of the CNN-LSTM hybrid highlights a broader methodological point: architectural depth and complexity are not substitutes for principled loss-activation alignment. The hybrid model's additional capacity offered modest benefits only in the presence of strong local motifs, but it did not consistently outperform the well-tuned LSTM baseline. This finding indicates that in economic forecasting tasks characterized by heterogeneity and rare-event risk, model capacity must be matched to the underlying signal struc-

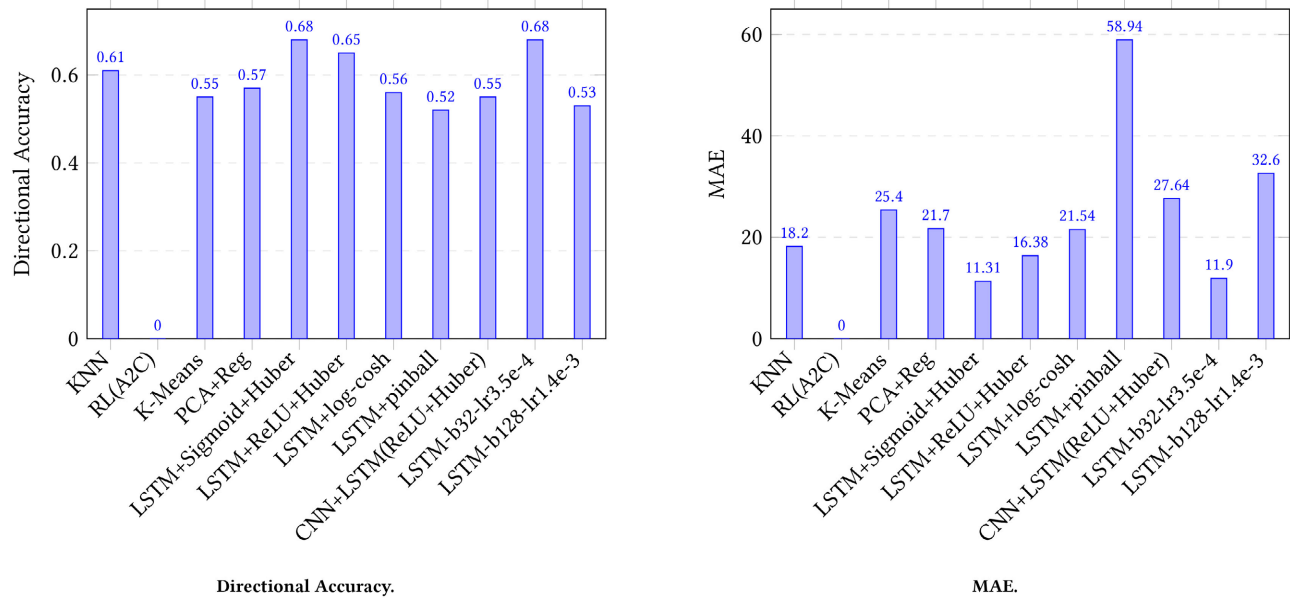
ture. Adding complexity without robust objectives risks amplifying noise rather than extracting meaningful patterns.

Finally, the collective evidence points toward a generalizable framework for model choice and configuration in economic forecasting. LSTM, when paired with robustness-oriented losses and bounded activations, should be the default for sequential prediction with sufficient data and strong autocorrelation. KNN remains an efficient and interpretable option for consumer-level tasks with smaller datasets. PCA and K-Means, while less effective as predictive engines, provide structural insights—dimensionality reduction and segmentation—that can complement sequential models in hybrid workflows. Reinforcement learning, though not evaluated in ablation form here, extends predictive capacity into policy optimization under uncertainty. This layered view of model selection moves beyond performance tables to articulate principled guidelines for practitioners navigating trade-offs among accuracy, interpretability, and computational cost.

*Why Huber + Sigmoid?* Huber behaves quadratically near zero (stabilizing small errors) and linearly for large residuals (tempering gradient explosions), which is advantageous when shocks produce fat-tailed errors. Coupled with a *Sigmoid* head that bounds outputs, the model avoids overshooting under volatility spikes, effectively acting as a statistical “smoother” akin to partial-adjustment behavior in markets. This combination yields the lowest tail-sensitive error while preserving directional accuracy, as reflected in [Table 1](#) and [Figure 25](#).

**Table 1.** Horizontal comparison of baseline models and vertical ablations of LSTM. Lower MAE/RMSE/MAPE and higher DA are better.

Model/Setting	Loss/Activation	MAE	RMSE	MAPE	R <sup>2</sup>	DA
<i>Horizontal comparison (baselines)</i>						
KNN	Euclidean distance	18.20	22.90	12.50		0.61
RL (A2C)	Reward-based optimization					converged
K-Means	Cluster average forecasts	25.40	30.10	18.70		0.55
PCA + Regression	Linear projection	21.70	27.80	15.90		0.57
<i>Vertical comparison (LSTM ablations)</i>						
<b>LSTM + Sigmoid + Huber</b>	<b>Robust baseline</b>	<b>11.31</b>	<b>15.00</b>	<b>7.84</b>	<b>-0.01</b>	<b>0.68</b>
LSTM + ReLU + Huber	Underperforms on volatility	16.38	21.25	11.21	-1.03	0.65
LSTM + log-cosh	Sensitive to spikes	21.54	25.69	13.82	-1.97	0.56
LSTM + pinball ( $\tau = 0.5$ )	Poor under outliers	58.94	61.00	37.53	-15.73	0.52
CNN + LSTM (ReLU + Huber)	Hybrid variant	27.64	29.80	17.53	-2.99	0.55
<i>Optimization sweeps (learning rate &amp; batch size)</i>						
LSTM (batch = 32, LR = $3.5 \times 10^{-4}$ )	Stable convergence	11.90	15.30	8.00	-0.05	0.68
LSTM (batch = 128, LR = $1.4 \times 10^{-3}$ )	Large-batch regime	32.60	34.85	20.69	-4.46	0.53



**Figure 25.** Directional Accuracy (left) and MAE (right) across baselines and LSTM variants. Higher DA and lower MAE are better.

### 5.3. Histogram-Based Evaluation and Interpretation

The histogram-based evaluation highlights that the LSTM configured with a Sigmoid activation and Huber loss achieves the highest directional accuracy ( $\approx 0.68$ ), outperforming all static baselines, including KNN ( $\approx 0.61$ ), PCA combined with regression ( $\approx 0.57$ ), and K-Means clustering ( $\approx 0.55$ ). This superiority underscores the importance of explicitly modeling temporal dependence in economic forecasting tasks such as hotel bookings and ADR prediction. While KNN relies on local similarity between samples in a static feature space, it cannot capture sequential dynamics, making it less effective for time-dependent problems. Similarly, PCA with regression retains most of the variance in the data but imposes linear structure, limiting its ability to accommodate nonlinear and state-dependent behaviors. K-Means clustering, though useful for segmentation, produces static partitions that are not adaptive to evolving temporal regimes, further explaining its lower predictive accuracy.

The choice of loss and activation functions also exerts a significant influence on performance. The combination of Huber loss with a bounded Sigmoid output stabilizes training in the presence of heavy-tailed residuals and rare spikes, yielding both lower error (MAE  $\approx 11.31$ ) and higher robustness to volatility. By contrast, the pinball loss exhibits poor performance (MAE  $\approx 58.94$ ), reflecting its sensitivity to distributional asymmetries in ADR and its misalignment with mean-based evaluation criteria. Other pairings, such as log-cosh with ReLU, tend to underperform due to gradient saturation or instability when exposed to extreme values.

Hyperparameter ablations further reveal the role of optimization strategies. Small-batch training (batch size = 32) with a learning rate of approximately  $3.5 \times 10^{-4}$  sustains high directional accuracy ( $\approx 0.68$ ) and low MAE ( $\approx 11.9$ ). In contrast, large-batch settings (batch size = 128) combined with proportionally scaled learn-

ing rates ( $\approx 1.4 \times 10^{-3}$ ) reduce performance to  $DA \approx 0.53$  and  $MAE \approx 32.6$ . These results suggest that while increasing learning rate with batch size is a conventional heuristic, in noisy economic series smaller batches provide beneficial gradient variability, improving generalization across nonstationary regimes.

Architectural comparisons further indicate that model complexity does not guarantee superior outcomes. The CNN-LSTM hybrid offers modest improvements over poorly tuned configurations (e.g., LSTM with pinball loss), but does not surpass the well-tuned baseline. This finding emphasizes that carefully aligned loss-activation-optimizer pairings can outweigh additional architectural depth in determining forecasting accuracy.

Taken together, the results from the horizontal comparison reinforce that LSTM models consistently outperform non-sequential methods by more accurately capturing directional shifts and reducing magnitude error. This advantage arises from their ability to encode lagged dependencies and nonlinear temporal interactions, which are critical for forecasting ADR and cancellation dynamics.

**Practical Guidance.** For production-oriented retraining, the recommended configuration is an LSTM with Sigmoid activation, Huber loss, a window size of 12, 64 hidden units, dropout = 0.2, batch size = 32, and learning rate  $\approx 3.5 \times 10^{-4}$ . If larger batches are necessary due to computational constraints, the learning rate should be scaled proportionally, with careful monitoring of directional accuracy (DA) and mean absolute error (MAE) for signs of performance drift. To enhance robustness to rare but impactful ADR spikes, exogenous event indicators (e.g., holidays or conventions) or anomaly-detection modules can be incorporated, enabling the model to balance baseline stability with responsiveness to economically significant shocks.

**Model Selection Strategy.** More broadly, the comparative analysis provides guidance on model selection in different application scenarios. When the prediction target exhibits strong temporal autocorrelation and sufficient training data are available, LSTM should be prioritized, as it consistently outperforms static methods in capturing directional shifts and nonlinear dynamics. However, its performance is highly contingent on the careful pairing of activation and loss functions, underscoring the value of ablation-driven tuning.

By contrast, when interpretability and computational efficiency are paramount—for instance, in consumer-facing recommendation systems or smaller datasets—KNN offers a reliable benchmark, providing competitive accuracy without the complexity of sequential training. PCA remains useful for dimensionality reduction in high-dimensional macroeconomic datasets, facilitating interpretability while preserving variance, though its linear structure limits predictive strength. K-Means, while less effective as a forecasting tool, remains valuable for uncovering static market segments and consumer clusters that complement predictive analyses.

Taken together, these findings suggest that model choice should be aligned with the temporal complexity of the target, the size and structure of the available da-

taset, and the balance between interpretability and predictive power. This framework allows practitioners to move beyond single-model deployment and adopt a principled strategy for integrating machine learning into economic forecasting pipelines.

## 6. Conclusions, Limitations, and Future Work

The comparative evaluation of K-Nearest Neighbor (KNN), Reinforcement Learning (RL), Principal Component Analysis (PCA), K-Means Clustering, and exploratory Long Short-Term Memory (LSTM) modeling underscores the diverse yet complementary strengths of machine learning in economic and behavioral applications.

KNN demonstrated strong predictive performance in consumer behavior and financial product recommendations, achieving an AUC of 0.81 and cross-validation accuracy exceeding 94% for small neighborhood sizes. However, its sensitivity to the curse of dimensionality necessitates rigorous preprocessing and precise parameter tuning to maintain robustness.

RL, implemented via the A2C algorithm, showcased adaptability in dynamic portfolio optimization tasks. The baseline agent prematurely converged to a local optimum of 136 steps per episode. However, with refined hyperparameter tuning—including learning-rate decay and entropy regularization—the tuned agent achieved up to 230 steps per episode, yielding a 46% improvement in mean returns while preserving stability.

PCA proved highly effective in reducing high-dimensional economic indicators, with the first two components capturing over 95% of the total variance. This not only simplified subsequent modeling tasks but also preserved interpretability and improved downstream efficiency.

K-Means clustering revealed meaningful consumer and market segments. Nevertheless, the average silhouette score of 0.23 suggested moderate cluster cohesion and overlap. These findings emphasize the need for advanced preprocessing, feature engineering, or hybrid pipelines (e.g., PCA + K-Means) to enhance segmentation quality and separation.

The exploratory LSTM baseline introduced temporal depth by modeling sequential dependencies in hotel ADR and cancellation behavior. The best-performing configuration—Huber loss with Sigmoid activation—achieved a directional accuracy of 0.68 and a mean absolute error (MAE) of 11.3, outperforming static models such as KNN (DA = 0.61) and PCA + regression (DA = 0.57). Despite this, the LSTM underfits rare but high-impact events (e.g., \$5000 ADR spikes), highlighting a trade-off between stability and responsiveness inherent in deep time-series forecasting.

### Limitations

While this study presents meaningful insights, it also encounters several limitations:

- **Dataset scope:** The datasets used are relatively narrow in domain, potentially limiting generalizability.
- **Simplified RL environment:** The CartPole simulation lacks the stochastic complexity of real-world financial systems.
- **Clustering constraints:** K-Means assumes spherical clusters and is sensitive to outliers and initialization.
- **Linear assumptions:** PCA captures only linear relationships, potentially missing key nonlinear interactions.
- **LSTM generalization:** The baseline LSTM model struggled to balance underfitting and capturing outlier events in volatile time series.
- **Reproducibility:** We provide seeds, hardware/versions, temporal splits, and full hyperparameters in Appendix A.
- **Exogenous signals:** External demand drivers (holidays, events, weather) are not yet included; future work will augment LSTM inputs with these variables.

While the LSTM baseline establishes a strong sequential benchmark, several limitations remain. First and most critically, the models omit external demand signals such as internet search indices, weather data, or local event calendars, which prior studies have shown can substantially enhance forecasting performance in tourism and hospitality contexts. Second, the current framework is primarily technical in nature, lacking integration with behavioral or economic theories that could explain customer decision-making and its impact on forecast variability.

Third, the model benchmarking is restricted to a limited set of baselines (KNN, K-Means, PCA, RL), excluding alternative sequential architectures such as Gated Recurrent Units (GRU), CNN-LSTM hybrids, or Deep Belief Networks (DBNs) that have shown competitive performance in related applications. Fourth, although the study explores optimization techniques such as Huber loss and adaptive learning rates, regularization strategies—including dropout tuning, systematic cross-validation, and data augmentation—are under-documented compared to best practices in the literature.

Fifth, outlier handling is largely dependent on loss function robustness, whereas additional preprocessing techniques like anomaly detection or correlation-based screening could further improve the treatment of extreme values. Finally, the evaluation framework lacks formal significance testing (e.g., paired *t*-tests, Diebold-Mariano tests), leaving uncertainty around whether observed improvements over baseline models are statistically meaningful or due to sample variation.

## Future Work

To address these challenges and build upon current findings, future research should explore:

- **Richer and more heterogeneous datasets** spanning multiple sectors, geographic regions, and behavioral dimensions.
- **Enhanced RL environments** incorporating stochasticity, delayed rewards,

and multiple economic agents.

- **Advanced clustering algorithms** such as DBSCAN or Gaussian Mixture Models (GMM) for more robust and flexible segmentation.
- **Nonlinear dimensionality reduction** techniques (e.g., t-SNE, UMAP) to capture complex, latent structures in high-dimensional data.
- **Hybrid pipelines** integrating CNN + LSTM architectures, autoencoder-enhanced clustering, or anomaly-aware forecasting modules.
- **Feature augmentation** using exogenous variables such as holidays, macroeconomic shocks, or social trends to improve time-series predictions.

*Hybrid architecture extensions.* We will 1) add exogenous features (holidays, market events, weather reanalyses), 2) stack PCA factors with raw exogenous covariates into a multi-branch encoder, and 3) fuse a shallow policy head (RL) with a forecasting head (LSTM) using a shared representation to support decision-aware predictions.

## Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

## References

- [1] Avramov, D., Cheng, S. and Metzker, L. (2023) Machine Learning Vs. Economic Restrictions: Evidence from Stock Return Predictability. *Management Science*, **69**, 2587-2619. <https://doi.org/10.1287/mnsc.2022.4449>
- [2] Bussmann, N., Giudici, P., Marinelli, D. and Papenbrock, J. (2020) Explainable Machine Learning in Credit Risk Management. *Computational Economics*, **57**, 203-216. <https://doi.org/10.1007/s10614-020-10042-0>
- [3] Charpentier, A., Élie, R. and Remlinger, C. (2021) Reinforcement Learning in Economics and Finance. *Computational Economics*, **62**, 425-462. <https://doi.org/10.1007/s10614-021-10119-4>
- [4] Chen, W., Xu, H., Jia, L. and Gao, Y. (2021) Machine Learning Model for Bitcoin Exchange Rate Prediction Using Economic and Technology Determinants. *International Journal of Forecasting*, **37**, 28-43. <https://doi.org/10.1016/j.ijforecast.2020.02.008>
- [5] Fan, X., Wang, X., Zhang, X. and Yu, X. (2022) Machine Learning Based Water Pipe Failure Prediction: The Effects of Engineering, Geology, Climate and Socio-Economic Factors. *Reliability Engineering & System Safety*, **219**, Article ID: 108185. <https://doi.org/10.1016/j.res.2021.108185>
- [6] Gewers, F.L., Ferreira, G.R., Arruda, H.F.D., Silva, F.N., Comin, C.H., Amancio, D.R., et al. (2021) Principal Component Analysis: A Natural Approach to Data Exploration. *ACM Computing Surveys*, **54**, 1-34. <https://doi.org/10.1145/3447755>
- [7] Giglio, S., Kelly, B. and Xiu, D. (2022) Factor Models, Machine Learning, and Asset Pricing. *Annual Review of Financial Economics*, **14**, 337-368. <https://doi.org/10.1146/annurev-financial-101521-104735>
- [8] Gogas, P. and Papadimitriou, T. (2021) Machine Learning in Economics and Finance. *Computational Economics*, **57**, 1-4. <https://doi.org/10.1007/s10614-021-10094-w>
- [9] Goldfarb, A., Taska, B. and Teodoridis, F. (2023) Could Machine Learning Be a General Purpose Technology? A Comparison of Emerging Technologies Using Data from

- Online Job Postings. *Research Policy*, **52**, Article ID: 104653.  
<https://doi.org/10.1016/j.respol.2022.104653>
- [10] Helliwell, J.F., Layard, R., Sachs, J.D., Aknin, L.B., De Neve, J.E. and Wang, S. (2023) World Happiness Report 2023. 11th Edition, Sustainable Development Solutions Network. <https://worldhappiness.report>
- [11] Hugging Face (2023) Deep Reinforcement Learning Course. <https://huggingface.co/learn/deep-rl-course/en/unit0/introduction>
- [12] Kaelbling, L.P., Littman, M.L. and Moore, A.W. (1996) Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, **4**, 237-285. <https://doi.org/10.1613/jair.301>
- [13] Kumar, S., Sharma, D., Rao, S., Lim, W.M. and Mangla, S.K. (2022) Correction To: Past, Present, and Future of Sustainable Finance: Insights from Big Data Analytics through Machine Learning of Scholarly Research. *Annals of Operations Research*, **332**, 1199-1205. <https://doi.org/10.1007/s10479-022-04535-4>
- [14] Li, G., Law, R. and Zhang, Z. (2018) Neural Network Approaches for Tourism Demand Forecasting: A Systematic Review. *Information Technology & Tourism*, **20**, 409-433.
- [15] Magazzino, C., Mele, M. and Schneider, N. (2021) A Machine Learning Approach on the Relationship among Solar and Wind Energy Production, Coal Consumption, GDP, and CO<sub>2</sub> Emissions. *Renewable Energy*, **167**, 99-115. <https://doi.org/10.1016/j.renene.2020.11.050>
- [16] Masini, R.P., Medeiros, M.C. and Mendes, E.F. (2021) Machine Learning Advances for Time Series Forecasting. *Journal of Economic Surveys*, **37**, 76-111. <https://doi.org/10.1111/joes.12429>
- [17] Medeiros, M.C., Vasconcelos, G.F.R., Veiga, Á. and Zilberman, E. (2019) Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods. *Journal of Business & Economic Statistics*, **39**, 98-119. <https://doi.org/10.1080/07350015.2019.1637745>
- [18] Mele, M. and Magazzino, C. (2020) Pollution, Economic Growth, and COVID-19 Deaths in India: A Machine Learning Evidence. *Environmental Science and Pollution Research*, **28**, 2669-2677. <https://doi.org/10.1007/s11356-020-10689-0>
- [19] Mirkes, E.M. (2011) K-Means and K-Medoids Applet. University of Leicester. [https://scholar.google.com/scholar?cluster=11364979353888248490&hl=en&as\\_sdt=5.39&scioq=Mirkes,+E.M.+\(2016\)+K-Means+and+k-Medoids+Applet](https://scholar.google.com/scholar?cluster=11364979353888248490&hl=en&as_sdt=5.39&scioq=Mirkes,+E.M.+(2016)+K-Means+and+k-Medoids+Applet)
- [20] Pan, B., Yang, Y. and Song, H. (2020) A Review of Travel Demand Forecasting: Models and Methods. *Sustainability*, **12**, Article 7334.
- [21] Xie, G., Qian, Y. and Wang, S. (2021) Forecasting Chinese Cruise Tourism Demand with Big Data: An Optimized Machine Learning Approach. *Tourism Management*, **82**, Article ID: 104208. <https://doi.org/10.1016/j.tourman.2020.104208>
- [22] Yoon, J. (2020) Forecasting of Real GDP Growth Using Machine Learning Models: Gradient Boosting and Random Forest Approach. *Computational Economics*, **57**, 247-265. <https://doi.org/10.1007/s10614-020-10054-w>
- [23] Zhang, H., Song, H. and Wen, L. (2019) Forecasting Tourism Demand with Search Engine Data and Long Short-Term Memory Models. *Sustainability*, **11**, Article 4708.

## Appendices

### Appendix A. Reproducibility Checklist

- **Data & Code:** repository/commit hash; preprocessing scripts and feature lists.
- **Splits:** temporal split (e.g., 70/15/15) and rolling-origin windows.
- **Random seeds:** CV, NN initialization, RL environment.
- **Hardware/Versions:** OS, CUDA, Python/R; library versions.
- **KNN:**  $k$ -grid, distance metric, scaler; ANN library if used.
- **K-Means:**  $k$ -grid, `init = K-Means++`, `n_init`, `max_iter`, scaler.
- **PCA:** scaler, #PCs kept, loading table.
- **LSTM:** window = 12, hidden = 64, dropout = 0.2, loss = Huber, head = Sigmoid, batch = 32, lr =  $3.5 \times 10^{-4}$ , cosine scheduler, epochs, early-stop.
- **RL:**  $\gamma = 0.99$ , entropy = 0.01, reward scaling rule, LR schedule.
- **Evaluation:** metrics (AUC/MAE/DA), CI method, DM-test config.

### Appendix B. Statistical Validation

We assess error-series differences using the Diebold-Mariano (DM) test on rolling-origin forecasts with a Newey-West variance estimate and a lag length matched to the forecast horizon. In addition, 1000 block-bootstrap resamples (block length selected by the Politis-White rule) yield confidence intervals for MAE/RMSE; table boldface denotes models significantly better than the best non-LSTM baseline at the 5% level.