**Question 2: AI-Driven Decision Systems in Large-Scale Enterprises: Challenges,**

**Innovations and Future Directions**

**NOTE: (Revised Version based on Prof. Aram Inputs)**

**Listing extra additions in revised version**

* KernelPCA (RBF) dimensionality reduction and reconstruction MSE.
* Developed a few tests to understand the performance of other methods in comparison to Autoencoder.
* Importance of Autoencoder for this project.
* PCA reconstruction and silhouette-score comparison.
* Clustering with raw features and silhouette-score evaluation.
* Comparative results of silhouette scores for raw, PCA, and KernelPCA.
* Explanatory paragraphs on code functionality and significance.
* Bullet-point summary of autoencoder justification tests.
* Wrote concise functionality & significance paragraphs explaining each block of code.

*This question has 2 parts.*

***Question 2 - Part 1 (50 pts): Theoretical Analysis***

1. **AI in Enterprise Systems**

* Identify three enterprise domains where deep learning and AI-driven decision-making systems have been successfully implemented (e.g., finance, healthcare, supply chain).

Deep learning and AI-driven decision-making systems been implemented with remarkable success across various enterprise domains. In healthcare, enormous amounts of clinical data get processed including the health records, medical imaging and genomic data which help facilitate precise diagnoses and enable personalized treatment strategies (Fardin Sabahat Khan, 17 November 2024).

Another domain where AI made substantial growth is supply chain management. In this realm, the AI is integrated into the Enterprise Resource Planning (ERP) systems to analyze historical trends and current market data for inventory optimization, demand forecasting and logistics management (Choudhuri, 2024).

In the pharmaceutical industry, AI-driven platforms have transformed research and development processes, particularly in drug discovery and development. These digital platforms employ machine learning algorithms to sift through vast datasets from clinical trial results to biomedical literature enabling the rapid identification of potential drug candidates (Yoshimasa Masuda, 2021).

* Discuss how AI enhances decision-making in these domains and the key benefits achieved.

In healthcare domain, by integrating the predictive analytics into healthcare supply chain management, AI enhances decision-making by forecasting patient demand, optimizing resource allocation, and automating routine tasks. These improvements lead to reduced operational costs and better patient outcomes, as hospitals can dynamically adjust staffing and inventory levels based on real-time insights (Fardin Sabahat Khan, 17 November 2024).

In supply chain management domain, real time analysis enables businesses to anticipate market fluctuations, minimize overstock or stockouts and streamline distribution processes. The benefits of these AI-enhanced systems include increased supply chain ability, cost savings, and improved customer satisfaction due to faster and more accurate decision making (Choudhuri, 2024).

In pharmaceutical industry, by expediting the analysis of complex data AI reduces the time and expenses required to bring new drugs to market. Furthermore, the integration of AI supports informed decision making in clinical trial design and patient safety evaluations, ultimately foresting innovation and offering a competitive edge in a highly regulated and dynamic market (Yoshimasa Masuda, 2021).

Collectively these three domains healthcare, supply chain management and pharmaceutical research demonstrate that AI can not only enhance operational efficiencies but also fundamentally transform strategic decision-making. The key benefits achieved include improving forecasting accuracy, cost reduction, enhancing agility and accelerated innovation, all of which are critical in the current rapidly evolving digital landscape as discussed in these papers (Choudhuri, 2024), (Fardin Sabahat Khan, 17 November 2024) and (Yoshimasa Masuda, 2021).

1. **Challenges in Deployment**

* What are the main challenges associated with deploying AI-driven decision-making models in large-scale enterprise environments?

Deploying the AI-driven decision-making models in large-scale enterprise environments presents a multitude of challenges that span technical, ethical, and operational dimensions. One of the forecast challenges is **model interpretability**. Deep learning models despite their high accuracy, it is known as “black boxes”, making it difficult for the decision-makers to understand the rationale behind their predictions (Fardin Sabahat Khan, 17 November 2024).

Another critical challenge is ensuring **fairness** in AI systems. These models are highly dependent on the quality and representatives of the training data. If the data is biased or incomplete, the resulting decisions may perpetuate existing inequalities, adversely affecting marginalized groups. Moreover, large scale enterprises often grapple with **real time inference constraints.** AI systems must deliver rapid, accurate decisions, a requirement that can be challenging when integrating the legacy infrastructures and processing massive heterogeneous datasets (Choudhuri, 2024).

* Discuss aspects such as model interpretability, fairness, robustness, data bias, and real-time inference constraints.

The lack of transparency can undermine stakeholder trust and hinder regulatory compliance, especially in sensitive sectors such as healthcare where understanding the basis of the clinical decisions is crucial. The opacity of these models the development of methods to extract **interpretable** insights, which is a significant technical challenge (Fardin Sabahat Khan, 17 November 2024).

To address fairness requires not only meticulous data curation and bias mitigation strategies but also continuous audits of model performance in real-world scenarios. In addition, **robustness** is essential. Models must reliably handle data variations, adversarial attacks and dynamic market conditions without significant performance degradation

The computational overhead of real time processing can strain existing IT systems, substantial investments in hardware and software optimizations. These challenges are compounded by the need for continuous monitoring, periodic model updates and seamless integration into existing workflows all while maintaining data integrity and security (Choudhuri, 2024).

As per the discussion in (Yoshimasa Masuda, 2021), such efforts are essential to realize the full potential of AI-driven decision-making models and to secure their long-term reliability and ethical deployment in large-scale enterprise environments.

1. **Multi-Agent AI and Reinforcement Learning in Enterprises**

* Explain how multi-agent AI or reinforcement learning (RL) can improve decision-making in complex enterprise environments.

Multi-agent AI and reinforcement learning (RL) offers transformative advantages in complex enterprise environments by decentralizing decision-making, enhancing adaptability and improving real-time responsiveness. In contrast to traditional centralized systems, multi-agent systems (MAS) distribute decision-making tasks across autonomous agents. Each agent operates based on localized information, makes decisions independently, and communicates with other agents to coordinate actions toward achieving global objectives. Additionally, reinforcements learning empowers these agents to learn from their interactions with the environment, gradually refining their decision policies based on feedback in the form of rewards or penalties. This continuous learning process allows the system to optimize performance metrics such as minimizing delays, balancing workloads, and reducing production costs in the face of uncertainty and variability inherent in complex enterprise settings (Yun Geon Kim, 2020).

Moreover (D, 2022) emphasizes that multi-agent reinforcement learning (MARL) can be applied to various aspects of smart factories beyond scheduling. Their review maps specific smart factory attributes such as the need for adaptive coordination, real-time decision-making, and efficient handling of uncertainties to the capabilities of MARL. Multi-agent AI and RL not only enhance the decision-making by distributing the computational load and enabling adaptive leaning but also drive substantial improvements in efficiency, agility, and robustness in enterprise environments as discussed in (D, 2022) and (Yun Geon Kim, 2020).

* Provide a real-world example where these approaches have been successfully used or could be applied.

A real-world example of these approaches is found in smart manufacturing systems. In a flexible smart manufacturing factory environment, multi-agent RL has been used to optimize production scheduling and resource allocation. For instance, (Yun Geon Kim, 2020) describe a system where intelligent agents embedded in manufacturing machines autonomously evaluate job priorities and negotiate job allocations. Reinforcement leaning algorithms enables these agents to learn from real time production data and adapt their scheduling strategies to dynamic conditions such as fluctuations in demand or machine breakdowns. As a result, the system achieves improved metrics in terms of reduced make span and minimized tardiness thereby enhancing overall productivity and efficiency (Yun Geon Kim, 2020).

In (D, 2022), an example of MARL approaches enables agents to cooperatively solve logistics and transportation problems in a decentralized manner, ensuing that resources are optimally utilized even under dynamic conditions. This capacity is particularly crucial in environments characterized by high dimensional data and rapidly changing operational parameters, where traditional rule-based systems often fall short (D, 2022).

These approaches allow enterprises to respond dynamically to real-world uncertainty, thereby achieving operational excellence in complex, modern manufacturing and other critical domains.

1. **Advancements and Open Research Challenges**

* Review recent advancements in deep learning and reinforcement learning relevant to enterprise applications.

The recent advancements in deep learning breakthroughs have emerged from improved neural architectures, large-scale training data, and hardware acceleration. Convolutional networks, originally used for computer vision tasks, have evolved into more diverse architectures, including transformers for language processing. Such models now serve as foundational building blocks for tasks ranging from image recognition to text generation, with self-supervised learning techniques enabling the robust feature extraction without massive labels. In enterprise environments, these advanced deep learning systems power high-impact applications like customer service chatbots, predictive maintenance for industrial machinery, and automated document processing, greatly reducing operational costs and enhancing decision making (Nguyen, 2020)

Meanwhile, reinforcement learning (RL) has expanded its scope through deep reinforcement learning (DRL), wherein deep neural networks approximate value functions or policies in complex, high-dimensional spaces. This has unlocked new possibilities for enterprises looking to optimize their decision making in dynamic environments. Multiagent DRL solutions, for example, coordinate fleets of autonomous vehicles, robotic teams in manufacturing, or load balancing in data centers. Recent work also explores DRL in wireless networks slicing, aiming to maximize resource utilization and service quality. Such advancements promise more resilient, scalable, and cost-effective operations across industries, bridging simulation-based training with real-world deployments (Ssengonzi, 2022).

* Use citations where appropriate and critically analyze both advantages and limitations of AI applications in enterprises.

AI applications have significantly transformed enterprise operations by automating processes, enhancing decision making, and optimizing resource allocation. Deep learning models such as transformers and convolutional networks have empowered enterprises to analyze vast amounts of data for tasks ranging from predictive maintenance to natural language processing in chatbots and customer service automation. Moreover, deep reinforcement learning (DRL) offers adaptive, self-optimizing systems that can learn optimal policies in dynamic environments. A crucial advantage for multiagent coordination in supply chain management and network operations (Nguyen, 2020).

However, several limitations temper these advantages. First, the complexity and computational demands of modern AI systems necessitate significant investment in infrastructure and high-quality data, making them prohibitively expensive for some enterprises. Additionally, the interpretability of deep models remains an issue; the “black box” nature of many algorithms can hinder transparency and regulatory compliance, while also complicating troubleshooting and model trustworthiness. In multiagent DRL setups used for instance, in network slicing and orchestration in next-generation networks challenges such as nonstationary and partial observability can lead to instability and suboptimal performance in real-world conditions (Ssengonzi, 2022).

Furthermore, biases inherent in training data and lack of proper contextual adjustments can result in unfair or inappropriate outcomes, and integration challenges with legacy systems can impede seamless deployment. In summary, while AI offers transformative advantages for enterprises by driving efficiency, and robust decision making, issues related to cost, interpretability, stability, and bias call for careful consideration and ongoing research to fully harness its potential in practical applications as discussed in (Nguyen, 2020) and (Ssengonzi, 2022).

***Question 2 - Part 2 (50 pts): Practical Coding with Google Colab: Implementation of an AI Model for Enterprise Decision-Making****.*

1. **Select an enterprise use case where deep learning can be applied (e.g., demand forecasting, fraud detection, customer segmentation).**

The chosen enterprise use case is **Customer Segmentation** within the context of credit assessment for a German bank. The objective is to group customers who have taken credit from the bank into distinct segments based on their demographic and financial attributes. This segmentation can help the bank make more informed decisions in areas such as risk management, personalized marketing and targeted credit product offerings.

1. **Implement a deep learning model (e.g., CNN, RNN, Transformer, or Reinforcement Learning agent) to address the chosen problem using Google Colab and a relevant framework (TensorFlow or PyTorch).**

**Comparative Evaluation of Linear PCA and Nonlinear KernelPCA for Customer Segmentation via Reconstruction Error and Silhouette Analysis:**

***Code Functionality***

This script begins by loading the German credit dataset and performing basic cleaning: The basic dataset cleaning includes trimming spaces from column names and replacing missing categorical values with "Missing" while removing the target column when it exists. The script one-hot encodes the chosen categorical features then changes all columns to numeric types and replaces any NaNs with zeros. The resulting feature matrix undergoes standardization to achieve a mean value of zero and a variance of one. The process applies two dimensionality reduction techniques which include linear PCA and nonlinear KernelPCA using an RBF kernel to project the data into a 5-dimensional latent space. Both dimensionality reduction methods produce the original high-dimensional data from the latent representation to calculate mean squared reconstruction error (MSE). The algorithm divides each latent embedding into three clusters through K-means and measures the quality of these clusters with the silhouette score. The script generates and shows a table comparing both PCA and Kernel PCA in terms of reconstruction error and clustering quality.

***Significance***

This code evaluates the PCA versus Kernel PCA trade-off by measuring reconstruction MSE and silhouette scores to assess data structure preservation and cluster discovery ability. The preservation of feature relationships improves with lower MSE values and clusters become more distinct and internally cohesive when silhouette scores increase. PCA functions as a basic linear approach to variance reduction while KernelPCA's RBF kernel applies fixed nonlinear transformations to reveal complex manifold structures. KernelPCA demonstrates superior clustering performance through higher silhouette scores despite its occasional higher MSE which shows how nonlinear methods benefit segmentation tasks. The diagnostics show which expressive trainable nonlinear models like autoencoders can enhance both reconstruction fidelity and cluster separability in customer segmentation tasks (Abdulhafedh, 2021)

**Comparative Silhouette Analysis of K‑Means Clustering on Raw Features, PCA, and RBF KernelPCA Embeddings:**

***Functionality***

The script implements three distinct clustering experiments on the German credit dataset to evaluate the effect of different feature transformations on cluster cohesion. The script begins by loading data and cleaning it followed by one-hot encoding of categorical columns while converting all features to numeric before filling missing values with 0 and standardizing each feature to mean 0 and unit variance. The script performs an initial K Means clustering on the complete standardized feature set (“Raw Features”) to determine a baseline silhouette score. The data is reduced to a 5-dimensional linear subspace using PCA before K-Means clusters this lower-dimensional representation and calculates the silhouette score. Kernel PCA with an RBF kernel operates in 5 dimensions to transform the data before clustering and evaluating the quality of the clusters using the silhouette score. We compile the results from all three clustering methods into a DataFrame for display which facilitates straightforward comparison of cluster quality between raw features and both linear and fixed nonlinear representations.

***Significance***

The code uses silhouette scores to evaluate each method's effectiveness at uncovering natural data groupings. Raw features produce low silhouette scores because of high dimensionality and noise in the space hide actual data segments. The moderate improvement achieved by PCA demonstrates its effectiveness at isolating clusters through principal linear directions while failing to detect curved relationships. The enhanced performance of KernelPCA (al., 2024) reveals that implementing fixed nonlinear transformations through RBF kernels allows the data manifold to separate more effectively resulting in improved clustering. These analyses demonstrate that learned nonlinear mappings such as autoencoders lead to superior cluster separation which supports their application in customer segmentation.

By using the TensorFlow/Keras to build an autoencoder for feature extraction and then applies KMeans clustering to segment the German bank credit customers. Then by using the silhouette score to evaluate clustering performance. This approach is suited well for the customer segmentation problem on tabular data. The autoencoder leans a lower-dimensional latent representation of the customers’ profiles that captures the nonlinear relationships among the various features (e.g. age, job, etc.). Clustering the latent representations then reveals natural segments in data.

1. **Provide an evaluation of the model’s performance using appropriate metrics (e.g., accuracy, F1-score, RMSE, or policy rewards for RL models).**

In google collaboration by testing the comparative evaluation of linear PCA and nonlinear KernelPCA for Customer Segmentation via Reconstruction Error and Silhouette Analysis. After that I have tested the Comparative Silhouette Analysis of K Means Clustering on Raw Features, PCA, and RBF KernelPCA Embeddings. As a result, I have got the below values.

**Significance of the Numeric Results**

|  |  |  |
| --- | --- | --- |
| **Method** | **Silhouette Score** | **Reconstruction MSE (if computed)** |
| **Raw Features** | 0.1594 | n/a |
| **PCA (k=5)** | 0.1863 | ~0.5398 |
| **KernelPCA (RBF)** | 0.3727 | ~0.6650 |
| **Autoencoder** | ~0.5254 | ~0.2003 |

***Raw Features (0.1594):*** The original data shows poor separation between clusters and excessive noise throughout its 20+ dimensional space.

***PCA (0.1863):*** PCA reduces noise through linear variance direction capture yet fails to model nonlinear relationships, so clusters appear clearer but remain overlapping.

***KernelPCA (0.3727):*** The application of a fixed nonlinear transformation (RBF) more effectively separates curved data manifolds resulting in cleaner clusters and reduced reconstruction error compared to PCA.

***Autoencoder (0.5254):*** The autoencoder achieves minimal reconstruction error through its self-derived nonlinear encoding using backpropagation while forming highly compact and well-separated silhouette clusters in its latent space.

The google collab code produced all the below results. After training the autoencoder for plotting the training and validation loss curves. I got the plot below.

A graph of loss and loss

AI-generated content may be incorrect.

The training curve (blue) is consistent below the validation curve (orange). This is common in neural networks. The model fits the training data more closely than it does the data set aside for validation. There is only a noticeable gap, it does not appear to drastically be diverging. It indicates that the model is not severely overfitting, though it does for the training data more strongly than unseen data.

The Early Epochs, shows the loss decreases rapidly, shows that model is quickly learning fundamental patterns about how to reconstruct the input. Later Epochs (Gradual Slope), in about 20-30 epochs the slope flattens shows that model is refining smaller nuances in the data leading to gradual improvements. The consistently lowered and stabilized validation loss suggests **latent representations** are likely capturing meaningful structure in data.

KMeans applies to latent representations to segment the customers. The Sil\_score measures the quality of clustering is 0.5254495 moderately good. Interpretation is that on average the clusters are reasonably well separated. It is also because of some overlapping between the clusters when the data is complex or noisy. The clustering distribution shows the segmentation balance and as a result the cluster 2 holds more data points here.

The values for the low MSE and low RMSE signal that

1.   MSE of 0.2003

2.   RMSE of 0.4476

autoencoder has successfully learned a compact representation of the input data and is capable of accurately reconstructing the original inputs from this latent space.

For tasks like feature extraction and clustering a robust reconstruction with small error is crucial it implies that most information is preserved in the latent representation.

**Reconstruction‑Error Comparison (PCA vs. KernelPCA vs. Autoencoder):**

Using a latent dimensionality of 5 we assessed the reconstruction accuracy of each method for the standardized features. In comparison to other linear methods PCA achieved the smallest reconstruction error with an MSE around 0.54. The use of an RBF kernel in Kernel PCA allowed for some nonlinear data patterns to emerge but resulted in poorer reconstruction performance (MSE ≈ 0.67) since the inverse transform gives only approximate results.

Our autoencoder approach involved simultaneous learning of both encoder and decoder networks to reduce reconstruction errors which resulted in a significantly lower MSE value of about 0.20. A task-driven nonlinear model outperforms fixed linear and fixed kernel methods by better preserving the original data structure within a compressed latent space.

**Clustering‑Quality Comparison**

We determined K Means' customer separation effectiveness across feature spaces through silhouette score calculations. The silhouette score of approximately 0.16 demonstrated that clustering barely managed to maintain structure in the raw 20+ dimensional data. Projecting data into a 5-dimensional linear subspace using PCA raised coherence to roughly 0.19 and then using Kernel PCA with a fixed RBF kernel untangled curved manifolds to achieve coherence around 0.37.

The strongest separation between clusters was achieved through clustering based on the autoencoder's learned latent embeddings which yielded a separation score of approximately 0.53. The consistent improvements in cluster tightness and separation from no reduction through linear to fixed nonlinear and finally to learned nonlinear demonstrate that autoencoder representations provide the best customer segmentation.

In summary, the autoencoder model is effectively compressing the data and reconstructing it with a high degree of fidelity, which is a positive outcome for downstream tasks such as clustering and segmentation.

1. **A Google Colab Notebook (Link with code and explanations).**

**Collab Link:** [**https://colab.research.google.com/drive/1Cv4R3uXV76y4W-E2OAjlRnOtL2ZrbiYS?usp=sharing**](https://colab.research.google.com/drive/1Cv4R3uXV76y4W-E2OAjlRnOtL2ZrbiYS?usp=sharing)

1. **A brief report**

**Problem Statement and Dataset:**

The objective of this project is to segment customers of a German bank to enhance the decision making related to risk management, tailored credit offerings, and targeted marketing. The dataset publicly available and originally prepared for credit risk analysis, contains approximately 1000 customer records with attributes such as Age, Job, Credit Amount, Duration and several categorical variables (Sex, Housing, Saving accounts, Purpose). Each customer record includes both numerical and categorical data, and while a credit risk column might be present, it was excluded to focus solely on discovering natural customer segments. Data preprocessing involved cleaning column names, handling missing values and converting attributes into one-hot encoded features.

*Kaggle Dataset:* <https://www.kaggle.com/datasets/kamaumunyori/german-bank-credit-data/data>

**Model Architecture and Training Process:**

To achieve effective segmentation, an autoencoder was employed for non-linear dimensionality reduction. The architecture consists of an input layer matching the number of preprocessed features, followed by an encoder with two dense layers (64 and 32 neurons) that compress the high-dimensional data into a latent space of five dimensions.

A symmetric decoder network then reconstructs the input from this lower-dimensional representation using the two dense layers (32 and 64 neurons) and a final output layer with linear activation. The autoencoder was implemented using TensorFlow and Keras, compiled with the Adam optimizer, and trained to minimize mean squared error (MSE) over 50 epochs using 20% of the data for validation. After training, the latent representatives were extracted and clustered using KMeans clustering (with 3 clusters specified), and the clustering performance was evaluated by using the silhouette score.

**Results and Key Takeaways:**

**Why the Simpler Models Fall Short:**

***Raw Features:*** The curse of dimensionality effects emerges when datasets have high dimensionality and mixed data types even after encoding procedures. The presence of numerous irrelevant or noisy features makes it difficult for K-means to identify spherical clusters.

***PCA:*** PCA removes noise by capturing the main variance directions, but it operates under the assumption that structural patterns are linear. The real-world customer behavior patterns such as age-savings-credit purpose interactions exist on complex curved manifolds which PCA fails to properly reveal.

***KernelPCA:*** The RBF kernel detects nonlinear patterns, but its mapping remains constant because of the fixed γ and kernel formula settings. The model does not learn to reduce reconstruction loss for your unique data set which results in partial manifold untangling and only approximate inverse reconstruction.

**Why the Autoencoder Excels:**

***Task Driven Nonlinearity:*** The autoencoder trains its encoding and decoding networks together from start to finish to reduce reconstruction error using this dataset. The autoencoder generates specialized representations that fit the real data distribution while Kernel PCA utilizes a static kernel approach.

***Denoising Effect:*** Autoencoders concentrate on important patterns while discarding random noise by compressing data into a narrow bottleneck (latent dimension 5), generating purer features for clustering.

***Better Cluster Geometry:*** The autoencoder creates latent space features which form spherical and distinct clusters perfect for K Means clustering leading to a silhouette score increase from ~0.37 with KernelPCA to ~0.53.

**Autoencoder Outperforms both:**

***Learned inverse*:** Autoencoders learn both the encoder and decoder to reduce reconstruction errors for your data unlike fixed RBF mapping. Autoencoders generate a lower MSE compared to PCA while creating a more suitable structure for clustering than Kernel PCA.

***Task-driven nonlinearity:*** The network adapts its nonlinearities using weights and biases so it can capture crucial patterns for reconstruction and cluster separation while KernelPCA uses one Gaussian kernel throughout.

***Empirical confirmation:*** Initial autoencoder results showed a better performance with MSE ≈ 0.20 and silhouette ≈ 0.53 compared to PCA which had lower MSE but worse clustering and KernelPCA which produced moderate clusters but higher MSE.

The final reconstruction metrics on the full dataset were an MSE of 0.2003 and a RMSE of 0.4476 indicating that the autoencoder is effectively reconstructing the data. The Silhouette score for the clustering on the latent features was satisfactory, suggesting that the resulting clusters are well separated and internally coherent. Analysis of the cluster distribution revealed that most customers fell into one dominant cluster, while a smaller subset displayed distinct characteristics.

These results imply that the autoencoder successfully captured complex, non-linear relationships among the features. The latent space contains meaningful structure that can differentiate between the “mainstream” customer group and more atypical segments. The insights obtained can guide the banks’ strategies for targeted marketing, risk adjustment, and resource allocation.

**Limitations and Potential Improvements:**

Even though the current approach has yielded promising segmentation, there are limitations and areas for enhancements. The dataset, with only 1000 records the dataset is relatively small, which may limit the generalizability of the findings. Additional data or augmented synthetic datasets could be beneficial. The choice of latent dimensions as (5) and the network architecture were selected empirically.

Further hyperparameter tuning, potentially with techniques like grid search or Bayesian optimization, could improve the performance of models. Although reconstruction metrics and silhouette scores provide valuable insights, external validation could strengthen the findings. Selecting more sophisticated methods for handling the missing data and feature selection or creatin might help enhance the quality of the latent features.

Overall, the segmentation analysis provides actionable insights that can be leveraged by the bank to better understand customer behavior and tailor strategic decisions accordingly.

# **References:**

* Abdulhafedh, A. (2021). Incorporating K-means, Hierarchical Clustering and PCA in Customer Segmentation. *Journal of City and Development*, 12-30.
* al., Z. W. (2024). A Method of Electricity Bill Payment Behavior Pattern Recognition Based on Kernel PCA and Hierarchical Clustering. *2024 International Conference on Computers*, 243-247.
* Choudhuri, S. S. (2024). AI - Driven Supply Chain Optimization: Enhancing Inventory Management, Demand Forecasting, and Logistics within ERP Systems. *International Journal of Science and Research* .
* D, B. F. (2022). A review of the applications of multi-agent reinforcement learning in smart factories. *doi: 10.3389/frobt.2022.1027340*.
* Fardin Sabahat Khan, A. A. (17 November 2024). AI in Healthcare Supply Chain Management: Enhancing Efficiency and Reducing Costs with Predictive Analytics. *Journal of Computer Science and Technology Studies*.
* Nguyen, T. T. (2020). Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Transactions on Cybernetics*.
* Ssengonzi, C. K. (2022). A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization. . *https://doi.org/10.1016/j.array.2022.100142*.
* Yoshimasa Masuda, A. Z. (2021). Adaptive Enterprise Architecture for the Digital Healthcare Industry: A Digital Platform for Drug Development. *https://doi.org/10.3390/info*.
* Yun Geon Kim, S. L. (2020). Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system. *Journal of Manufacturing Systems*.
* Some parts of the code were developed with the reference of ChatGPT (OpenAI, 2025) to understand the importance of each test method and adapted to suit the project requirements.