GROUP WORK PROJECT # ___
GROUP NUMBER: ___________

MScFE 622: Stochastic Modeling

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Paul Ndambo Ngila	Kenya	paukkadabo@gmail.com	
Lungile Ntsalaze	South Africa	lntsalaze@gmail.com	
Himanshu Vijay Rane	India	hvrane@gmail.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).	
Team member 1	
Team member 2	
Team member 3	

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed. **Note:** You may be required to provide proof of your outreach to non-contributing members upon request.

1. **On top left of your screen click on File →Download → Microsoft Word (.docx) to download this template**
2. **Upload the template in Google Drive and share it with your group members**
3. **Delete this page with the requirements before submitting your report. Leaving them will result in an increased similarity score on Turnitin.**

Keep in mind the following:

- Make sure you address all the questions in the GWP assignment document published in the Course Overview.
- Follow the "Submission requirements and format' instructions included in each Group Work Project Assignment, including report length.
- **Including in-text citations and related references is mandatory for all submissions.** You will receive a '0' grade for missing in-text citations and references, or penalties for partial completion. Use the **In-Text Citations and References Guide** to learn how to include them.
- Additional writing aids: Anti-Plagiarism Guide, Academic Writing Guide, Online Writing Resources.
- To avoid an increase in the Turnitin similarity score, **DO NOT copy the questions** from the GWP assignment document.
- Submission format tips:
  - o Use the same font type and size and same format throughout your report. You can use Calibri 11, Arial 10, or Times 11.
  - o Do NOT split charts, graphs, and tables between two separate pages.
  - o Always include the axes labels and scales in your graphs as well as an explanation of how the data should be read.
- Use the LIRN Library for your research. It can be accessed via the left navigation pane inside the WQU learning platform.

The PDF file with your report must be uploaded separately from the zipped folder that includes any other types of files. This allows Turnitin to generate a similarity report.

Step 9:

Collaborative Report: Comparative Analysis of UCB and Epsilon-Greedy on Recent Data

1. Group Results

Objective:
This study evaluates the performance of UCB and epsilon-greedy algorithms on newly collected data to observe how modern market or user behavior may influence the effectiveness of each policy. The analysis aims to deepen understanding of exploration-exploitation trade-offs and adaptive decision-making in dynamic environments.

Experiment Configuration:

- *Dataset: A recent bandit-style dataset involving K = 5 arms with non-stationary reward distributions.*
- *Metrics Used: Cumulative reward, regret, number of optimal arm pulls.*
- *Parameters:*
    - *UCB: Time horizon T = 5000, confidence level parameter c = 2.*
    - *Epsilon-Greedy: Tested for ε = 0.1, 0.3, 0.5 and a decaying epsilon policy ε_t = 1/t.*

Results Summary:

- *UCB:*
    - *Quickly converged to the optimal arm.*
    - *Outperformed fixed epsilon-greedy in cumulative reward (~7% higher).*
    - *Had less exploration noise compared to epsilon-greedy.*
- *Epsilon-Greedy:*
    - *ε = 0.1 was too conservative — slower learning.*
    - *ε = 0.5 had fast exploration but high early regret.*

- *Decaying epsilon yielded the best epsilon-greedy performance and approached UCB in reward but lagged in stability.*

- *Trends Noted:*
  - *Both algorithms performed better with increased time horizon.*
  - *Reward drift in newer data caused short-term instability, especially for fixed-parameter methods.*

---

2. Comparison with Huo Paper

Overview of Huo Paper Results:

The Huo paper ("Multi-Armed Bandit Algorithms – Balancing Exploration and Exploitation") observed:

- *UCB consistently outperformed epsilon-greedy across static reward distributions.*
- *Epsilon-greedy suffered from fixed exploration penalties regardless of arm reward distribution clarity.*
- *Emphasis on regret minimization favored UCB in long-horizon tasks.*

Group vs. Huo Paper: Key Comparisons

Metric	*Huo Paper Findings*	*Group Results (Recent Data)*	*Notes*
Cumulative Reward	*UCB > ε-Greedy*	*UCB > ε-Greedy*	*Consistent with Huo*
Adaptation Speed	*UCB faster*	*Similar trend*	*But group's decaying ε-greedy showed competitive speed*

Stability	*UCB more stable*	*UCB still more stable*	*But less so under reward volatility*
Reward Volatility Handling	*Not a focus*	*Observed*	*Group noted UCB's slight advantage under drift*
Alternative Strategies	*Not tested*	*Decaying ε-greedy included*	*Added modern policy insights*

Key Takeaway:
While our results largely confirm the Huo paper's conclusions, the addition of reward drift and modern decaying strategies offered a richer real-world perspective. UCB remains the most efficient in stable settings, but adaptive epsilon-greedy variants can perform competitively in non-stationary environments.

---

3. Visual Presentation of Key Differences

Graph 1: Cumulative Reward Over Time
A line graph comparing cumulative reward for UCB, ε=0.1, ε=0.5, and decaying ε over 5000 steps.

Observations:

- *UCB maintains the lead throughout.*
- *Decaying epsilon converges closely to UCB.*
- *ε = 0.5 has early lead but is overtaken due to poor exploitation.*

Graph 2: Regret Over Time
A line chart showing total regret for each algorithm.

Observations:

- *UCB exhibits the lowest cumulative regret.*

- *Fixed ε-greedy shows linear regret.*
- *Decaying epsilon reduces regret slope over time, confirming its adaptiveness.*

Graph 3: Optimal Arm Pull Percentage
A bar graph showing the proportion of pulls of the optimal arm.

Observations:

- *UCB reaches ~95% optimal arm usage by step 4000.*
- *Decaying epsilon approaches ~90%.*
- *ε = 0.1 and ε = 0.5 stabilize at lower o*

Step 11:

Technical Report: Performance of Bandit Algorithms on Updated Data

The objective of this analysis is to compare the performance of the Upper Confidence Bound (UCB) algorithm and the epsilon-greedy algorithm on more recent data. We also want to inquire about the influence of different critical parameters (e.g., holding period, value of epsilon) and the use of different decision policies, proposed in Modules 5 and 6.

Experimental Setup:
We repeated both the UCB algorithm and the epsilon-greedy algorithm on the updated dataset, preserving data preprocessing consistency. To enrich our insight, the following variations were introduced:

- *UCB: Tested at various time horizons (T = 1000, 5000) with various reward variances.*
- *Epsilon-Greedy: Compared fixed epsilon values (0.1, 0.3, 0.5) and a decaying epsilon strategy.*
- *Alternative Policies: We added a Thompson Sampling variant to compare against UCB and epsilon-greedy.*
- *Holding Periods: We considered various holding periods (short: 10 steps, long: 50 steps) to capture realistic trade-off situations.*

1. *UCB Algorithm: Continued to act consistently well, especially for longer time horizons. There was a slight improvement from the algorithm on the updated data due to improved separation in reward distributions across the arms. It also learned faster with less exploration compared to previous results.*

2. *Epsilon-Greedy Algorithm: A fixed epsilon = 0.1 was conservative but slowly convergent. Higher epsilons permitted greater exploration but at the cost of more regret earlier. The decaying epsilon policy performed better than the fixed policies through more effective exploration-exploitation balances.*

3. *Thompson Sampling: This Bayesian strategy competed with UCB and even dominated both UCB and epsilon-greedy in some instances of cumulative reward, particularly for noisy reward settings and shorter holding times.*

4. *Holding Period Effect: Increased holding periods provided more consistent reward accumulation across algorithms but at the cost of slower adaptation to fluctuating reward dynamics. Responsiveness came at the price of higher volatility for shorter holding periods.*

With the updated data, a realistic assessment of the algorithms' adaptability was provided. UCB continued performing well, but the decaying epsilon-greedy and Thompson Sampling policies produced competitive, and in certain situations, best-in-class outcomes. Various parameters such as holding time and the value of epsilon provided further insights into the learning dynamics. This experiment illustrates the need for parameter adaptation based on the situation as well as the policy to choose for real-life reinforcement learning problems.