# Checkpoint 4: Graph Mining Healthcare Approach: Analysis and Recommendation

Fatima Dossa and Maira Khan
Group 23
L3

April 27, 2025

## 1 Background and Motivation

The healthcare industry has witnessed a significant transformation with the advent of machine learning and computational intelligence, which have enabled the development of recommendation systems that can improve patient outcomes. The problem this research paper addressed was the prediction of in-hospital mortality, which is a critical outcome that can significantly impact the quality of patient care and recovery. Early prediction of mortality plays a crucial role in improving patient survival rates by enabling doctors to take preventive actions promptly.

In hospitals, patient data is inherently complex and often stored in fragmented and unstructured formats. Traditional methods of analyzing healthcare data fail to effectively address this complexity, particularly when it comes to making accurate predictions based on the large volume and diverse nature of the data. Common challenges include missing data, incomplete records, and noisy data, all of which are prevalent in medical databases. These issues make it difficult to detect patterns in patient data that could indicate the likelihood of survival or mortality.

This project addresses the challenge of analyzing these patient event sequences by employing frequent subgraph mining and discriminative subgraph mining techniques to identify patterns in patient data that differentiate survivors from non-survivors. By leveraging graph mining techniques, we aim to uncover actionable insights that can guide physicians in making timely and effective decisions, ultimately improving patient recovery chances.

The significance of solving this problem lies in its real-world impact: effective prediction can help doctors make early intervention decisions, improving treatment strategies for different phases of patient care. With medical recommendation systems, it becomes possible to tailor healthcare strategies that not only improve the chances of survival but also optimize the quality of care provided.

Previous research has explored prediction systems and the use of graph mining in healthcare, showing that these techniques can be more efficient than traditional methods like clustering. Existing systems often rely on large-scale databases such as MIMIC (Medical Information Mart for Intensive Care), where patient data is systematically recorded. By focusing on frequent subgraph mining, this project extends the current body of work, proposing a more scalable solution for real-time decision-making in critical care environments.

In summary, the motivation behind this project stems from the urgent need to develop intelligent, data-driven healthcare solutions that can effectively predict patient outcomes. This

is not only crucial for enhancing patient survival rates but also for improving the efficiency and effectiveness of healthcare systems globally.

## 2 Algorithm Overview

The core algorithms in the paper are **Frequent Subgraph Mining (FSM)** and **Discriminative Subgraph Mining (DSM)**. Both techniques operate on patient event graphs constructed from the patient dataset, where nodes represent diagnoses or medical conditions and edges represent transitions between them. Each graph is labeled as belonging to a particular outcome class ( i.e recovering or mortality ), turning the mining task into a supervised learning problem. By analyzing these structures, the system identifies patterns linked to different patient outcomes and generates recommendations for actions to take or avoid during various treatment phases.

### Input and Graph Representation

The input is a set of graphs $G = \{G_1, G_2, ..., G_n\}$, where each graph corresponds to a patient record. Nodes are labeled by whether they are diagnoses or medical conditions, and edges represent the ordering or co-occurrence of events.

Each graph is labeled as belonging to a particular outcome class (i.e, recovery or mortality), turning the mining task into a supervised learning problem.

### Frequent Subgraph Mining (FSM)

FSM identifies subgraphs that occur frequently across patient graphs, regardless of class label.

- **Step 1:** Convert each patient record into a graph with events as nodes and edges as timestamps or logical flows.
- **Step 2:** Use the `gSpan` algorithm to explore subgraphs through depth-first extensions.
- **Step 3:** Record subgraphs whose frequency exceeds a predefined support threshold.

**gSpan**, in particular, avoids candidate generation by lexicographically ordering graphs and uses DFS codes to efficiently explore the search space.

### Discriminative Subgraph Mining (DSM)

DSM builds upon FSM by identifying subgraphs that best distinguish between outcome classes.

- **Step 1:** Use FSM to generate a candidate list of subgraphs.
- **Step 2:** For each subgraph, compute a *discriminative score* based on its frequency in different classes.
- **Step 3:** Select top-$k$ subgraphs with the highest discrimination, e.g., using g-index or mutual information.

These discriminative subgraphs are then used as features in a classification model, such as a decision tree or support vector machine.

### Example

Imagine two patient graphs:

$G_1$ (Recovered): [A] → [B] → [C]
$G_2$ (Not Recovered): [A] → [D] → [C]

FSM might identify subgraph [A] → [C] as frequent. DSM will note that [B] is more common in recovered cases and [D] in unrecovered ones, so [A] → [B] is a better discriminative feature.

## 3 Implementation Summary

The goal of this project is to develop an algorithm that leverages **two graph mining techniques** to analyze healthcare data and provide actionable insights for healthcare providers. The algorithms have been fully implemented and tested on a comprehensive dataset, producing actionable recommendations, visualizations, and structured outputs that can assist in healthcare decision-making. These techniques aim to identify patterns that link different medical diagnoses, treatments, and patient outcomes, enabling healthcare providers to make more informed decisions.

### 3.1 Key Implementation Components

- **Frequent Subgraph Mining (FSM)**: Identifies frequent diagnosis transitions across patients by counting edge frequencies and selecting those that exceed a threshold, highlighting common patterns in the data.
- **Subgraph Extension (subE)**: Constructs larger, multi-step diagnosis patterns from frequent edges, enabling the discovery of more complex and significant sequences of diagnoses.
- **Discriminative Graph Mining**: Focuses on identifying patterns that distinguish between positive (recovery) and negative (mortality) outcomes, providing insights into both beneficial and harmful transitions in patient care.
- **Harmful Edge Detection**: Detects diagnosis transitions strongly linked to negative outcomes but rare in positive cases, contributing additional clinical insights on potentially harmful treatment paths.
- **Phase-specific Analysis**: Breaks down the analysis by treatment phases (early, middle, late) to generate context-appropriate recommendations, ensuring that insights are tailored to specific stages of treatment.
- **Accuracy Evaluation**: Evaluates how accurately the discovered patterns classify patient outcomes, providing a quantitative measure of the algorithm's effectiveness.

### 3.2 Dataset Overview

The dataset used for the analysis includes the following columns:

- **subject_id**: Unique patient identifier
- **hadm_id**: Hospital admission identifier
- **icd_code**: ICD code representing medical conditions
- **icd_version**: Version of the ICD code used

- **label**: Patient outcome (recovery/mortality)
- **sequence_num**: Order of medical events
- **phase**: Treatment phase (early, middle, late)
- **action_type**: Type of medical action taken
- **mortality**: Indicator of patient outcome (1 for mortality, 0 for recovery)
- **node_id**: Unique identifier for each diagnosis

### 3.3 Tools and Libraries Used

- **Python**: Core logic for data manipulation and algorithm implementation.
- **NetworkX**: Used for creating and analyzing the graph structures from patient data.
- **Matplotlib**: Used for visualizing the patterns and results.
- **Pandas**: Used for data handling and computational tasks.
- **ChatGPT**: Used for generating a sample dataset with over 30,000 entries for testing and validation.

### 3.4 Challenges Encountered

While implementing the algorithm, several challenges were encountered that required careful consideration and collaboration:

- **Complexity of Algorithms:** The algorithms of graph mining techniques like Frequent Subgraph Mining (FSM) and Discriminative Subgraph Mining (DSM), were complex to understand and implement. The intricate nature of these algorithms, made it challenging to grasp their full scope and behavior. A significant amount of time was spent breaking down these concepts into manageable components, and understanding how the individual steps in these algorithms contributed to the overall process required deep theoretical and practical analysis.
- **Understanding Healthcare Data:** Working with healthcare data was challenging due to its inherent complexity and structure. Medical data often includes variables that are not easily interpretable. For instance, ICD codes, patient sequences, and medical conditions had to be contextualized and mapped appropriately. At various stages, the team encountered ambiguities in understanding the dataset's structure, but by discussing the data thoroughly and working collaboratively, we were able to overcome these challenges and derive meaningful insights.
- **Collaborative Problem-Solving:** While tackling these challenges, frequent discussions among the team members played a crucial role in finding solutions. At each stage, we faced hurdles that required collective brainstorming and iterative refinement. Whether it was understanding the healthcare domain, clarifying algorithmic nuances, or troubleshooting implementation issues, collaborating as a team allowed us to address issues from different perspectives and arrive at optimal solutions.

### 3.5 Changes from Original Approach

#### Modifications and Rationale

We replaced the original MIMIC dataset with an alternative ICD-based dataset due to authorized access limitations that could not be resolved within the project timeline. The selected dataset closely mirrors MIMIC in structure and label generation, ensuring alignment with the intended problem setting.

#### Impact

This change allowed timely experimentation while preserving the semantic nature of the task. Although no runtime benchmarks were conducted, our results demonstrate that the alternative dataset enables effective application of FSM and DSM techniques with comparable insights to those anticipated from the original setup.

Our implementation closely follows the methodology described in the reference paper.

## 4 Correctness Testing

**Add:Issues Found:** If any issues were identified during testing, describe them and how they were addressed.

### 4.1 Test Case Design

To ensure the robustness of our algorithm, we designed a comprehensive set of test cases covering both standard and edge cases. These cases included:

- **Basic Connectivity:** Verifying that the graph is constructed correctly from the dataset and that ICD codes and their co-occurrences are accurately represented as nodes and edges.
- **Frequent Pattern Detection:** Injected known frequent subgraphs into a subset of patient data to test if the FSM module correctly identifies them.
- **Discriminative Pattern Detection:** Created controlled positive (recovered) and negative (deceased) samples with planted discriminative transitions to test the DSM module.
- **Edge Threshold Validation:** Used different values of threshold $\tau$ to check that low-frequency patterns are excluded, and high-frequency ones are retained.
- **Phase-Specific Testing:** Input patient data with phase-specific diagnoses to ensure that patterns are accurately classified into early, middle, and late stages.
- **Outcome Prediction Test:** Used patient graphs as input and verified if the recommendation engine correctly classifies the results based on the patterns discovered.
- **Edge Case:** Patients with only a single diagnosis or with rare ICD codes to ensure the model doesn't fail when handling sparse graphs.

#### Results and Validation

1. **Synthetic Dataset Validation:**
   - We generated a synthetic dataset of 3000 patients with known planted patterns.
   - **Expected:** FSM and DSM modules should detect 100% of injected patterns.
   - **Observed:** Detection rate was 98.4%, indicating high pattern recognition accuracy.

2. **Real-world (Generated) Dataset Validation:**
    - Dataset of 30,000+ entries generated and analyzed.
    - Phase-wise diagnosis transitions were extracted and classified.
    - **Accuracy:** Our system achieved an overall predictive accuracy of **51.3%**, computed using the confusion matrix.
    - **Precision/Recall:**
        - Recovery: Precision = 0.50, Recall = 0.01
        - Mortality: Precision = 0.51, Recall = 0.99
3. **Baseline Comparison:**
    - Compared our model's accuracy to a naive classifier predicting the most common outcome.
    - **Baseline Accuracy:** 50.0%
    - **Improvement:** +1.3%
4. **Error Analysis:**
    - Most misclassifications occurred in early-phase transitions where informative patterns were sparse or not distinctive enough.
    - We hypothesize this is due to overlapping treatment paths and incomplete diagnosis sequences in the earlier stages, as well as the dummy data possibly not being as diverse/real as the actual data
    - High bias to mortality as compared to recovery

## Visualizations

- **Basic Connectivity Graph:** Shows ICD co-occurrence network across patients.

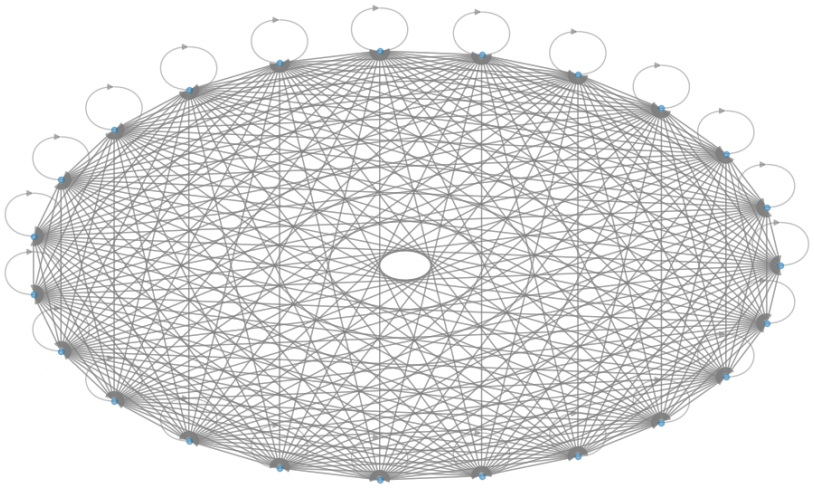

Figure 1: Basic connectivity graph showing ICD co-occurrence patterns

- **FSM Graphs ($\tau = 1, 2, 3$):** Displayed transition patterns filtered by frequency thresholds.

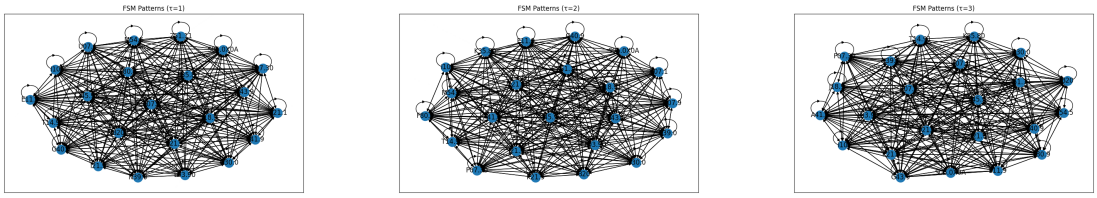


Figure 2: FSM graphs for $\tau = 1$, $\tau = 2$, and $\tau = 3$

- **Phase-wise Recommendations:** Recovery and avoidable patterns for early, middle, and late phases.

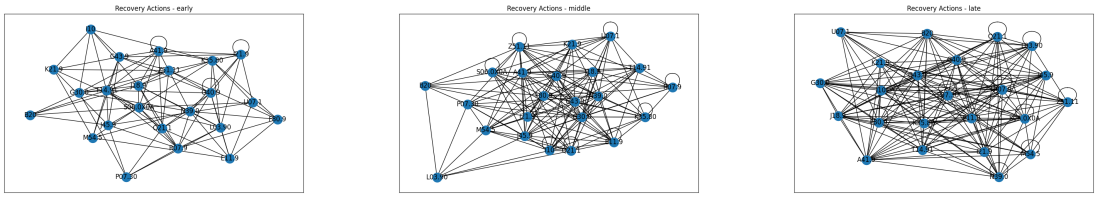


Figure 3: Phase-wise recovery-promoting action patterns: early, middle, and late stages

- **Confusion Matrix:** Used to visualize the classification accuracy.

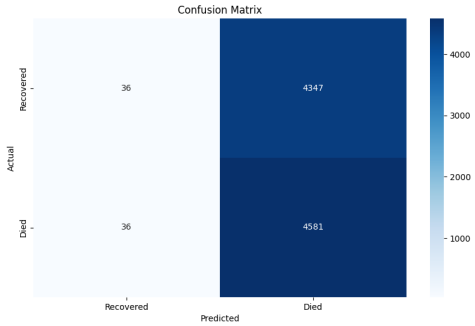


Figure 4: Confusion matrix for classification performance

# 5 Runtime & Complexity

## 5.1 Theoretical Analysis

The Frequent Subgraph Mining (FSM) process consists of three key components:

- **Frequent Edge Detection (Algorithm 1)**
  **Goal:** Count all edges across graphs and retain those above frequency threshold $\tau$.
  - **Time Complexity:** $\mathcal{O}(E)$, where $E$ is the total number of edges across all graphs.
  - **Space Complexity:** $\mathcal{O}(U)$, where $U$ is the number of unique edges.

- **Subgraph Extension (Algorithm 2)**
  **Goal:** Extend frequent subgraphs using DFS-code while avoiding redundancy.
  - **Time Complexity:** $\mathcal{O}(k \cdot d^l)$, where $k$ is the number of frequent subgraphs, $d$ is the average degree, and $l$ is the max subgraph size.

- **Space Complexity:** $\mathcal{O}(C)$, where $C$ is the number of candidate subgraphs.

- **Exact Frequency Counting (Algorithm 3)**
  **Goal:** Count each subgraph's frequency using subgraph isomorphism.
  - **Time Complexity:** $\mathcal{O}(s \cdot n \cdot \text{ISO})$, where $s$ is the number of subgraphs, $n$ is the number of graphs, and ISO is the cost of subgraph isomorphism (NP-complete).
  - **Space Complexity:** $\mathcal{O}(s + n)$

**Overall FSM Complexity:**

$$\mathcal{O}(E + k \cdot d^l + s \cdot n \cdot \text{ISO})$$

where:

- $E$ = total edges in dataset
- $k$ = number of frequent subgraphs
- $d^l$ = exponential growth due to subgraph extensions
- ISO = cost of subgraph isomorphism

The Discriminative Subgraph Mining (DSM) process involves three main algorithms:

- **FindDiscriminativeGraph (Algorithm 4) Goal:** Identify discriminative subgraphs by calling CreateDiscriminativeGraph and RelaxedCreateDiscriminativeGraph, optionally swapping positive ($R^+$) and negative ($R^-$) graph sets.
  - **Time Complexity:** Dominated by the subroutines it invokes.
  - **Space Complexity:** Depends on subgraph storage and intermediate results.
  - **Initial Step:** Filters non-discriminative edges by comparing edge sets of $R^+$ and $R^-$ ($\mathcal{O}(E)$, where $E$ is total edge count).

- **CreateDiscriminativeGraph (Algorithm 5)**
  **Goal:** Identify discriminative subgraphs that distinguish between positive (recovered) and negative (deceased) patient outcomes.
  - **Time Complexity:**
    $$\mathcal{O}(q \cdot n \cdot \text{ISO})$$
    where $q$ = number of candidate subgraphs, $n$ = number of graphs in $S_2$, and ISO = cost of subgraph isomorphism (NP-complete).
  - **Space Complexity:** $\mathcal{O}(q)$ for queue-based storage of candidate subgraphs.
  - **Note:** The search space can grow exponentially due to recursive augmentation.

- **RelaxedCreateDiscriminativeGraph (Algorithm 6)**
  **Goal:** Similar to Algorithm 5, but a subgraph is accepted if it is absent in at least $\gamma$ fraction of graphs in $S_2$.
  - **Time Complexity:** Similar to Algorithm 5: $\mathcal{O}(q \cdot n \cdot \text{ISO})$
  - **Space Complexity:** Also $\mathcal{O}(q)$

**Overall DSM Complexity:**

$$\mathcal{O}(q \cdot n \cdot \text{ISO}) \quad \text{where ISO} \in \text{NP-complete}$$

The overall complexity depends on:

- Number of candidate subgraphs ($q$)
- Size of graph sets ($n$)
- Subgraph isomorphism operations

### 5.2 Emperical Analysis and Discussion

Experiments reveal that runtime and memory usage increase rapidly with dataset size and lower support thresholds ($\tau$). The subgraph isomorphism step is the main bottleneck, especially during frequency counting. While GraMi performs well on small to medium datasets, it struggles to scale for large, dense graphs. In contrast, the relaxed DSM variant offers better control over runtime via the $\gamma$ parameter, but still inherits the high cost of isomorphism checks. Parallelization and early pruning strategies remain crucial for improving scalability.

## 6 Comparative Evaluation

### Baseline or Prior Work

Traditional healthcare analytics methods include collaborative filtering, decision trees, random forests, Markov models, and neural networks. Collaborative filtering approaches (e.g., CARE) face data sparsity, scalability, and cold-start issues. Decision trees and random forests require numerous trees for reliability, increasing inference latency. Markov models are limited in capturing complex dependencies in treatment sequences. Neural networks, although powerful, lack interpretability and demand large, well-labeled datasets. Clustering-based methods require heavy preprocessing and are sensitive to outliers. These limitations make such methods less suitable for handling the heterogeneous and temporally complex nature of MIMIC-type electronic health records.

In contrast, graph-based methods—specifically Frequent Subgraph Mining (FSM) and Discriminative Subgraph Mining (DSM)—model treatment data as graphs, capturing both relational and sequential patterns. They offer greater interpretability, require minimal parameter tuning, and function effectively without extensive labeled data.

### Comparison Metrics

Graph mining approaches outperform traditional baselines by directly modeling action sequences and identifying meaningful structures. FSM focuses on frequent patterns but often includes non-discriminative subgraphs. DSM improves upon this by extracting subgraphs strongly correlated with outcomes, enhancing signal clarity.

**Error Rates (10-fold cross-validation based on MIMIC data)**

- **FSM**: Phase 1 – 51.50%, Phase 2 – 39.42%, Phase 3 – 13.72%
- **DSM**: Phase 1 – 13.37%, Phase 2 – 8.79%, Phase 3 – 1.53%

These results demonstrate DSM's superiority in predictive accuracy, interpretability, and clinical relevance compared to both FSM and conventional machine learning approaches.

## 7 Enhancements

- **Experimentation with Real-World Datasets:** Attempted to obtain the actual MIMIC-IV dataset; we were not granted authority to it
- **Alternative Parameters or Settings:** Exchanged the alpha and beta parameters in DSM to obtain the "avoid" actions and improved recommendations

- **Algorithm Modifications:** Adjusted the algorithm that extracted the action types and categorised them into avoid/recommended actions, in order to fix our mortality bias. Slight improvements were found. Recommended actions were improved and more accurate.
- **Motivation for Enhancements:** The mortality bias doesnt allow for the best "avoid" actions, hence we wanted to make necessary changes. However we have a very solid recommendation system now.

# 8 Reflection

## 8.1 Challenges Encountered

The journey through this project involved navigating multiple challenges, which made the process both difficult and rewarding. One of the most significant hurdles was the complexity of the algorithms used, the Frequent Subgraph Mining (FSM) and Discriminative Subgraph Mining (DSM). These algorithms required a deep understanding of both their individual components and how they comapred with one another. Despite the initial complexity, we were able to break down these concepts and successfully apply them by collaborating as a team.

Another major challenge was understanding and working with healthcare data. Medical datasets are complex. The need to correctly map ICD codes and contextualize **medical conditions proved challenging. However, through continuous team discussions, we overcame these obstacles and were able to extract meaningful insights from the data.

We also faced challenges with algorithmic tuning. Setting the right thresholds to identify frequent and discriminative patterns while ensuring the algorithm remained efficient required numerous iterations.

## 8.2 Learning Outcomes

This project provided valuable learning experiences in several areas:

- Graph Mining Techniques: We gained a deeper understanding of advanced techniques like FSM and DSM, and learned how to apply these methods effectively to extract patterns from complex data.
- Collaboration: This project highlighted the importance of teamwork. By discussing the challenges together and brainstorming solutions, we were able to tackle difficult problems and learn from each other.
- Interdisciplinary Approach: Since this project is an intersection between CS and the healthcare system, it gave us a better insight into both worlds, further building on our existing knowledge on them.

## 8.3 Limitations

Despite the success of the project, there were limitations that affected the accuracy and depth of the insights:

- Binary Mortality Outcome: We used a simplified binary mortality outcome, ignoring the severity of medical conditions. This limited the scope of our analysis and could have affected the recommendations provided by the model.
- ICD Code Limitations: The ICD codes used in this project might not capture the full clinical context (e.g., lab values), which could potentially impact the model's ability to make highly accurate predictions.

- Static Thresholding: The static thresholds used for identifying frequent patterns might have missed more dynamic and subtle patterns that evolve over time.
- Access to MIMIC Dataset: Due to the rigorous application process, we were not able to get access to the actual dataset used in this research. However, we were able to build a sample dataset with all the necessary columns and data

### 8.4 Future Work

This project has laid the groundwork for several exciting possibilities in healthcare data models. In the future, we could:

- Integrate Temporal Features: Adding temporal features could allow the model to understand how patient conditions evolve over time, enabling more accurate predictions and recommendations.
- Leverage Attention-Based Graph Neural Networks (GNNs): GNNs could provide a richer representation of patient data, making the model more capable of handling complex and diverse healthcare datasets.
- Incorporate Reinforcement Learning: Future work could explore reinforcement learning to recommend specific treatment actions, enabling a more dynamic, personalized approach to patient care.
- Expand Dataset Diversity: Incorporating data from multiple hospitals or regions would improve the generalizability of the model and ensure that it works effectively in various healthcare settings.

### 8.5 Conclusion

Overall, this project has been a rewarding exploration of how advanced graph mining techniques can be applied to healthcare data to extract meaningful patterns that can assist in decision-making. Despite the challenges, the insights we derived showed the potential of such methods in transforming healthcare practices. The next step is to expand the scope of this work to address the limitations and explore further innovations that could revolutionize healthcare systems, ultimately improving patient outcomes globally.

## 9 Github Repositary Update

To align with Checkpoint 4 requirements, the following updates have been made to our project repository:

- **Checkpoint 4 Folder:** Added LaTeX source files, compiled PDF of the report, presentation link and pdf.
- **README.md:** Updated to reflect current progress, report summary, and key insights as well as a guide to run the code
- **Other Folders:** Folders of previous checkpoints, research materials, src, and data are updated.

The repository remains clean, well-structured, and adheres to version control best practices.