Anjali Velamala
CSPB 2270-003: Data Structures & Algorithms
Project Proposal: Implementing a Matrix Class

# Matrix Class

**Motivation**

As I prepare to delve deeper into machine learning through upcoming courses in computer science and applied mathematics, I want to solidify my understanding of fundamental linear algebra concepts. Matrices are a crucial component in various machine learning algorithms, from simple linear regression to more complex neural networks. By implementing a matrix class from scratch, I will gain a more comprehensive understanding of matrix operations and their underlying mechanics. This project not only serves as a review but also as a practical exercise to bridge theoretical knowledge with practical implementation. I hope it will also serve as a valuable tool for future assignments.

**Project Scope and Goals**

The primary goal of this project is to create a matrix class that can perform a variety of matrix operations on any $m \times n$ matrix (given the size is correct for the operation). The class will include methods for:

1. Scalar Multiplication
2. Matrix Multiplication
3. Row Swaps
4. Matrix transpose
5. Gaussian Elimination and Row Echelon Form
6. Matrix Inverse
7. Solving a Matrix-Vector Equation (using Gaussian elimination and the Inverse as two different methods)
8. LU factorization (I will explore pivoting strategies for numerical stability as needed)

If I have time I will implement more matrix algorithms.

**Why This Is an Interesting Application**

Matrix operations are at the heart of numerous algorithms in machine learning and data science. By implementing these operations from scratch, I will gain a deeper understanding of their computational complexity and efficiency. This understanding is critical as it influences algorithm design and optimization, particularly when working with large datasets. Moreover, this project serves as an intersection between theoretical linear algebra and practical programming, providing a tangible way to apply mathematical concepts.

**Technical Challenges and Learning Opportunities**

1. Efficiency and Optimization: One of the main challenges will be ensuring that the matrix operations are efficient, particularly for large matrices. Understanding time complexity will be important.
2. Numerical Stability: Implementing methods like Gaussian elimination and LU factorization requires careful handling to maintain numerical stability and avoid errors due to floating-point precision issues. For LU factorization there are pivoting methods I will look into to help with stability
3. Error Handling and Edge Cases: Ensuring the matrix class can handle edge cases, such as singular matrices or incompatible dimensions, will be an important part of the implementation.