*Semester Project: Credit-Card Fraud Detection*

**Problem Statement:**

Credit card fraud is a common problem that costs millions of dollars to financial institutions and cardholders. Fraudsters use various techniques to deceive banks and individuals, making it challenging to detect fraudulent transactions. Therefore, there is a need to develop a reliable system that can identify fraudulent transactions and minimize the losses incurred.

**Goals:**

The primary goal of this project is to build a credit card fraud detection system that can accurately identify fraudulent transactions. This system will use logistic regression as a machine learning algorithm to detect fraudulent transactions. The project's success will be measured by its ability to detect fraudulent transactions with high accuracy and reduce the number of false positives and false negatives.

**Objectives:** The objectives of this project are:-

1. Collect and preprocess the credit card transaction dataset.
2. Build a logistic regression model to detect fraudulent transactions.
3. Train and evaluate the model's performance on the credit card transaction dataset.
4. Improve the model's accuracy by tuning the hyperparameters.
5. Implement the final model as a Command-Line Interface (CLI) for real-time fraud detection.

**Methodology**: The methodology of our project is in the following order:-

1. Data collection: The credit card transaction dataset will be obtained from public sources such as Kaggle.
2. Data preprocessing: The dataset will be prepared for model testing and training, which includes processing the source file into a format which can be worked on, preparing subsets of data for test and train data, and to account for imbalanced data and any other challenges.
3. Model building: The logistic regression algorithm will be implemented using Python's Scikit-learn library to build a fraud detection model. Other libraries may also be used for implementation of various other models for evaluation purposes.
4. Model evaluation: The performance of the model will be evaluated using metrics like the precision score, recall, F1 score, and ROC-AUC. We will also compare the model's performance with other machine learning models such as Decision Trees, Random Forests, and Neural Networks for comparison (Exact models TBD).
5. Model tuning: The model's hyperparameters will be tuned to further improve its accuracy.
6. Implementation: Implement the final model as a command-line interface (CLI) for simplicity and testing purposes.

**Implementation Plan:**

Week 1: Data collection and preprocessing.

Week 2: Model building.

Week 3: Model testing and tuning.

Week 4: Model evaluation and deliverables documentation.

Week 5: Project Delivery.