**Data Science - Fraud Detection**

**Project Report**

**Introduction**

In this project, I focused on developing a fraud detection system to identify fraudulent transactions within a dataset. The primary objective was to analyze transaction data and build predictive models to accurately detect and flag fraudulent activities. This report outlines the methodologies used, the challenges encountered, and the key findings from the project.

**Methodology**

**Data Preparation**

- I began by importing necessary libraries such as pandas, seaborn, and matplotlib for data manipulation and visualization.

- The dataset was loaded, and initial exploration revealed the presence of outliers in the transaction amounts.

- I decided to keep fraudulent outliers as they represent genuine fraud cases, while non-fraudulent outliers were initially retained to assess their impact on model performance.

- Irrelevant columns, such as `nameOrig` and `nameDest`, were removed as they did not add predictive value to the fraud detection models.

**Algorithms Used**

- Throughout the project, I explored various machine learning algorithms to determine which would be most effective for fraud detection.

- Initially, I considered using Support Vector Classifier (SVC), Random Forest, and Multi-Layer Perceptron (MLP). However, these algorithms proved challenging due to their long training times and significant RAM usage.

- [Specific algorithms used, e.g., Logistic Regression, Decision Trees, Gradient Boosting, etc.]

- Each algorithm was evaluated based on performance metrics such as accuracy, precision, recall, and F1-score.

**Model Training and Evaluation**

- Models were trained using the preprocessed data, and their performance was evaluated using appropriate techniques.

- The evaluation metrics provided insights into which models performed best in detecting fraudulent transactions.

**Challenges Encountered**

**Data Issues**

- One of the primary challenges was dealing with data quality issues, such as outliers and irrelevant columns.

- These issues required careful handling to ensure that the models could learn effectively from the data.

**Algorithm Limitations**

- Long Training Times: Algorithms like SVC, Random Forest, and MLP took a considerable amount of time to train, which was not feasible given the project constraints.

- RAM Issues: These algorithms also required significant memory resources, leading to RAM issues during training.

- Balancing the need for accuracy with computational efficiency was a key consideration throughout the project.

**Findings**

**Results**

- The evaluation results indicated that certain algorithms performed better than others in detecting fraudulent transactions.

- [Specific performance metrics for each algorithm, e.g., "Logistic Regression achieved an accuracy of 95% with a precision of 88% and recall of 90%."]

- A detailed comparison of the models was conducted, revealing which algorithms were most suitable for the task at hand.

**Comparison**

- The comparison highlighted the strengths and weaknesses of each algorithm in the context of fraud detection. For detailed analysis refer -> <https://colab.research.google.com/drive/1Qn-gXmkj6x1qMaYjOH161jO49QPM17LZ?usp=sharing>

**Conclusion**

This project provided valuable insights into the effectiveness of different machine-learning algorithms for fraud detection. The findings will inform future work and help in developing more robust fraud detection systems. The experience gained from this project will be invaluable in addressing similar challenges in the future.