📄 ML Model Evaluation Report

1. Introduction

This report evaluates three supervised learning models: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression. The objective is to identify the most effective model based on classification accuracy, robustness, and generalization capabilities for the given dataset.

2. Model Performance Summary

Model	Accuracy	Key Characteristics
K-Nearest Neighbors (KNN)	48%	Simple, instance-based, sensitive to feature scaling, computationally expensive during inference.
Support Vector Machine (SVM)	85%	Strong generalization, margin-based classifier, robust.
Logistic Regression	96%	Highly robust, interpretable, strong convergence on dataset, balanced accuracy and interpretability.

3. Error Analysis & Major Failure Modes

3.1 KNN Failure Modes

- High misclassification rates due to noisy neighbors and struggles with overlapping classes.

3.2 SVM Failure Modes

- Errors observed in borderline and overlapping cases.
- Slight performance degradation on imbalanced subsets.

3.3 Logistic Regression Failure Modes

- Occasional misclassification in rare or minority classes.
- Limited capability in capturing complex non-linear relationships within the data.

4. Lessons Learned

- **Baseline Importance**: KNN served as a valuable baseline model for comparison.
- **Feature Scaling**: Demonstrated to be crucial for the successful performance of both SVM and Logistic Regression.
- **Model Selection**: Logistic Regression generalized best for this specific dataset, exhibiting superior performance.
- **Trade-offs**: Highlighted the inherent trade-offs between models: KNN offers simplicity but poor generalization; SVM provides strong performance but is computationally heavier; Logistic Regression achieves a good balance of interpretability and accuracy.

5. Conclusion

- **Final Model**: **Logistic Regression (96% accuracy)** is selected as the top-performing model due to its superior accuracy and robustness.

5.1 Recommendations

- Deploy Logistic Regression for production.
- Explore ensemble methods or non-linear kernels to further improve model performance.
- Investigate strategies for handling class imbalance to reduce misclassifications in rare-case scenarios.