CSE422 Lab Project Report

Software Quality Prediction

Submitted by:
Student ID: 22301565
Date: January 2, 2026

1. Introduction

This project aims to predict the quality of software modules based on various code metrics. In software engineering, maintaining high quality and low bug density is crucial. By using machine learning models to analyze features like Lines of Code (LoC), Cyclomatic Complexity, and Code Churn, we can proactively identify modules that might require more testing or refactoring. The project explores both supervised learning (for classification) and unsupervised learning (for pattern discovery).

2. Dataset Description

The dataset used is software_quality_dataset.csv, containing 500 records.
- How many features?: 5 input features (LoC, Complexity, Churn, Bugs, Unit Tests).
- Classification or regression?: Classification (Target: Software_Quality).
- Feature Types: Quantitative (LoC, Complexity, Churn, Bugs) and Categorical (Unit Tests).
- Encoding: Label Encoding was used for 'Has_Unit_Tests'.

Correlation Analysis:
Imbalanced Dataset Check:

3. Dataset Pre-processing

Faults & Solutions:
- Null Values: Dropped rows with missing data using dropna().
- Categorical Values: Encoded 'Has_Unit_Tests' into 0/1.
- Feature Scaling: Applied StandardScaler to normalize numerical features.

4. Dataset Splitting

Split into 80% Training and 20% Testing sets using random state 42 for reproducibility.

5. Model Training & Testing (Supervised)

Models trained: KNN, Decision Tree, Logistic Regression, Naive Bayes, Neural Network, Random Forest, SVM, AdaBoost.

Unsupervised Learning (K-Means):
The data points were grouped into 3 clusters representing different code complexity regimes.

6. Model Selection / Comparison Analysis

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.81	0.81	0.81	0.81
Decision Tree	0.84	0.84	0.84	0.84
Random Forest	0.89	0.89	0.89	0.89
SVM	0.87	0.87	0.87	0.87
KNN	0.82	0.82	0.82	0.82
AdaBoost	0.77	0.77	0.77	0.77
Naive Bayes	0.85	0.85	0.85	0.85
Neural Network	0.91	0.91	0.91	0.91

Best Model Confusion Matrix:

7. Conclusion

The Neural Network (MLPClassifier) achieved the highest accuracy of 91.4%. Feature scaling and proper handling of categorical variables were essential for these results. The primary challenge was managing feature ranges and interpreting the K-Means clusters.