|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Study** | **Confounding Variable(s)** | **Data Set** | **Analysis** | **Results** |
| **Basili et al, 1996** | None factored in | Eight student projects written in C++ | Univariate linear regression. Multivariate logistic regression | Finds that LCOM is not a significant predictor of fault-proneness while the remainder of the CK metrics are. |
| **Tang et al., 1999** | None factored in | Three small/medium commercial systems written in C++ | Logistic regression | RFC and WMC strong predictors for fault-proneness |
| **Emam et al., 1999** | Class size (LOC) | One medium-sized commercial project | Logistic regression | After controlling for size, only CBO was an indicator of fault-proneness |
| **Cartwright and Shepperd, 2000** | None factored in | One large commercial project | Linear regression | Found the inheriting classes were more defect prone (identified as classes having a DIT or NOC > 0) |
| **Subramanyam and Krishnan, 2003** | None factored in | One large commercial project written in C++ and Java | Linear regression | CBO, DIT, WMC predictive of fault-proneness |
| **Xu et al., 2008** | Class size (LOC) | One medium-sized government project written in C++ | Neural networks | CBO, RFC and WMC are reliable metrics for defect estimation finding that overall |
| **Malhotra and Jain, 2012** | None factored in | One medium/large-sized FLOSS project written in Java | Logistic regression and machine learning techniques | Machine learning models comparable in performance to linear models. Found that CBO, LCOM, RFC and WMC not to be significant predictors of fault-proneness. The rest of the CK metrics were indicators. |