NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES

RESEARCH PAPER

Predictive Financial Modelling for Software Project Success: A Comparative Analysis of Machine Learning Techniques and Traditional Methods

Software Engineering Economics

Course Instructor: Iqra Fahad

Rahima Irfan (21K-3107), Muhammad Rahim Khan (21K-4536), Parshant Vijay (21K-4613)

May 8, 2025

Abstract

This project provides an enhanced comparative analysis of traditional software project estimation methods, the Constructive Cost Model (COCOMO), and modern machine learning techniques to predict software success. Based on previous research, we discuss how these methodologies work when assessing project outcomes, especially when it comes to their predictive accuracy and practical use for project management. Our results show that machine learning models, particularly Support Vector Machine (SVM), are much superior to the traditional COCOMO estimates that are 91–97% accurate versus 76%. We suggest a new decision framework for resolving prediction disagreements and show its application through a detailed case study. This research helps in the developing field of software project management by offering empirical proof of the superiority of hybrid approaches that combine traditional domain knowledge with machine learning capabilities.

1 Introduction

Failure of software projects is still a recurring problem in the industry, where projects tend to exceed budgets, miss deadlines, and fail to meet stakeholder expectations. The ability to accurately predict outcomes of a project early in the development lifecycle is of significant importance for effective allocation of resources and risk management, and strategic planning. Industry has been using traditional methods of estimation, such as COCOMO. Traditional standards have been used for decades, whereas machine learning approaches are relatively new innovations in project success prediction.

This research aims to:

1. Compare the accuracy of predictions from traditional COCOMO modelling and machine learning methods.
2. Identify primary factors towards project success and failure.
3. Create an all-encompassing framework for solving issues in predictions.
4. Give evidence-based recommendations for implementation in practical terms in project management.

1.1 Problem Statement

The complexity of modern software development projects, defined by dynamic requirements, the combination of diverse technologies and global teams, has increasingly made traditional estimation models insufficient. Although COCOMO delivers structured estimation mainly in the form of technical parameters, it cannot portray the complexity of success in a project, which is not merely concerned with financial outcomes and stakeholder satisfaction, in addition to schedules.

1.2 Research Significance

This research fills the gap in the literature of project management by offering comparative analysis between traditional and machine learning approaches to the prediction of project success. By comparing both methodologies from the perspective of the financial results, we provide tips to dramatically enhance the project planning, allocation of resources, and risk management in software development.

2 Background

2.1 Traditional Estimation Methods

The Constructive Cost Model (COCOMO), developed by Barry Boehm in 1981, has been a foundation of software project estimation for decades. It uses regression formulas to estimate effort and duration, mainly based on software size in terms of thousands of lines of code (KLOC).

- Effort = $a \times (\text{KLOC})^b$
- Duration = $c \times (\text{Effort})^d$

Where a, b, c, and d are parameters whose values depend on the type of project (Organic, Semi-detached, or Embedded). Although COCOMO has been widely adopted, it has significant limitations, including a static nature, interrelationships between variables, and the inability to adjust to the changing nature of software development methodologies.

2.2 Machine Learning in Project Prediction

Recent developments in machine learning have provided new opportunities for the prediction of success in projects. Jorgensen and Shepperd (2007) did a systematic review on the limitations of traditional approaches and the need for data-driven models. Idri et al. (2005) suggested the use of fuzzy reasoning to overcome some of the problems with crisp models such as COCOMO. Lately, Huang et al. (2018) showed that Random Forest algorithms could be used for predicting software defects, while Kumar et al. (2019) included both technical and non-technical factors in their ML-based estimation models. The machine learning capabilities have developed at an unprecedented rate, which has made it easier for more complex ways of project success prediction that can consider complicated interactions between project variables.

2.3 Research Gap

Although previous research has already considered the use of machine learning for effort estimation and defect prediction, there is still a gap in comparative analysis between the traditional and machine learning approaches, specifically in terms of predicting the financial success of the project. This research closes this gap by directly comparing these methodologies and suggesting a framework for integrating them.

3 Research Methodology

3.1 Data Collection

This study utilizes comprehensive project data including: Budget (Estimated and Actual), Revenue Generated, Project Duration (Estimated and Actual), Lines of Code, Bug Reports, Team Size, Change Requests, Industry Type, Cost and Schedule Overrun percentages, Project Success (Target: 1 for success, 0 for failure).

3.2 Machine Learning Models

The research uses four classification algorithms:

1. Logistic Regression (LR): A statistical model that utilises logistic function to model a binary dependent variable.
2. Support Vector Machine (SVM): A supervised learning model used in analysing data for classification.
3. K-Nearest Neighbours (KNN): A non-parametric procedure for classification and regression.
4. Naive Bayes (NB): A probabilistic classifier that uses the application of Bayes' theorem.

These models were trained based on historical data of projects and tested with various performance metrics, including accuracy, precision, recall, and F1-score.

3.3 COCOMO Implementation

For COCOMO estimation, we created an implementation of the standard formula for three project modes:

- Organic (KLOC $\leq$ 50): $a = 2.4$, $b = 1.05$, $c = 2.5$, $d = 0.38$
- Semi-detached (50 < KLOC $\leq$ 300): $a = 3.0$, $b = 1.12$, $c = 2.5$, $d = 0.35$
- Embedded (KLOC > 300): $a = 3.6$, $b = 1.20$, $c = 2.5$, $d = 0.32$

The success of a project in COCOMO was measured by comparing the estimated duration with the actual duration, with a 15% tolerance threshold.

3.4 Feature Selection

Feature selection was done using Pearson correlation to determine the best predictive variables. Features having a correlation coefficient of more than 0.1 with the target variable (Project Success) were retained. This assisted in dimensionality reduction without losing predictive power.

3.5 Model Evaluation

An 80/20 split of train-test was used for training and testing of the model. Each model was evaluated using accuracy, precision, recall, and F1-score among multiple metrics. Hyperparameter tuning was performed using GridSearchCV to optimise model performance.

4 Results and Analysis

4.1 Model Performance Comparison

Table 1 presents a comparative performance of the four machine learning models.

Table 1: Performance Comparison of Machine Learning Models

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	91%	0.92	0.92	0.90
SVM	97%	0.97	0.97	0.97
KNN	94%	0.94	0.94	0.94
Naive Bayes	95%	0.95	0.95	0.95

SVM showed the best overall performance of 97% accuracy, followed by Naive Bayes (95%), KNN (94%), and Logistic Regression (91%). SVM demonstrated high balance on all metrics, thus it was especially appropriate in financial prediction tasks where both false positives and false negatives can have significant consequences.

4.2 ML vs. COCOMO Performance

Table 2 presents a comparative analysis between ML models and COCOMO:

Table 2: Comparative Analysis of ML Models vs. COCOMO

Comparison Dimension	ML Models	COCOMO
Accuracy	91%–97%	76%
Adaptability to new data	High	Low
Interpretability	Moderate	High
Sensitivity to project dynamics	High	Low
Implementation complexity	Moderate	Low
Computational Requirements	Moderate	Low

ML models, in particular SVM, outperformed COCOMO by a large margin in prediction accuracy. COCOMO's rule-based approach cannot account for the complex, nonlinear relationships between project variables that can be identified by ML models. Moreover, ML models are more flexible to new data and can learn from the patterns of history, while COCOMO uses constant parameters.

4.3 Feature Importance Analysis

The most predictive features for project success identified in our analysis were:

1. Revenue Generated
2. Cost Overrun Percentage
3. Actual Budget Spent
4. Schedule Overrun Percentage

Surprisingly, even traditional metrics, such as Lines of Code, did not correlate as well with the project's success as economic indicators, which might mean that economic indicators are more indicative of the outcome of a project than purely technical measures.

4.4 Exploratory Data Analysis Insights

Key insights from the exploratory data analysis include:

- Regular gap between predicted and actual budgets, suggesting systematic underestimation.
- High correlation between the overrun of the budget and the failure of projects.
- Revenue-to-cost ratio is a material determinant of the success of a project.
- Different success rates in various industries.
- The relationship between team size and project success is not linear.
- Change requests versus project failure correlation is positive.

5 Case Study: Resolving Prediction Disagreements

To demonstrate the practical use of our results, we give a detailed case study of a sample software project where ML and COCOMO predictions diverge.

- Profit = Revenue - Actual Budget = \$650,000 - \$500,000 = \$150,000
- Profitability = (Profit/Budget) $\times$ 100 = (\$150,000/\$500,000) $\times$ 100 = 30%

Table 3: Project Parameters

Parameter	Value
Actual Budget Spent	\$500,000
Cost Overrun	12%
Actual Duration	24 months
Schedule Overrun	20%
Revenue Generated	\$650,000
Estimated Duration	20 Months
Lines of Code	200,000

5.1 COCOMO Analysis

- KLOC = 200,000/1,000 = 200
- Project Mode: Semi-detached (50 < KLOC $\leq$ 300)
- Parameters: $a = 3.0$, $b = 1.12$, $c = 2.5$, $d = 0.35$
- Effort = $3.0 \times (200)^{1.12} \approx 1133.11$ person-months
- Duration = $2.5 \times (1133.11)^{0.35} \approx 24.3$ months

COCOMO Prediction:

- Success if: Duration $\leq$ Estimated Duration $\times$ 1.15
- 24 months $\leq$ 20 months $\times$ 1.15
- 24 months $\leq$ 23 months
- Result: Failure

5.2 Model Analysis

Table 4 presents the performance of the machine learning models for the case study project.

Table 4: Model Performance for Case Study Project

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	91%	0.92	0.92	0.90
SVM	97%	0.97	0.97	0.97
KNN	94%	0.94	0.94	0.94
Naive Bayes	95%	0.95	0.95	0.95

- Success Score = SVM + KNN = 97% + 94% = 191%
- Total Accuracy = LR + SVM + KNN + NB = 91% + 97% + 94% + 95% = 377%
- Weighted Success Score = 191/377 = 0.507 (> 0.5)

ML Consensus: Success

5.3 Novel Decision Framework for Handling Model Disagreement

We suggest a new framework for comparison between COCOMO and ML's disagreement.

Condition 1: Profitability $> 50\%$ AND Cost Overrun $< 10\%$.

- In case both criteria are satisfied, prefer ML prediction.
- Great profitability and low cost overrun indicate that the ML prediction could be more accurate.
- Financial health indicators are good, even with possible scheduling problems.

Condition 2: $|$COCOMO Duration - Estimated Duration$| >$ Estimated Duration $\times$ 0.25

- If met, favour COCOMO prediction.
- Big schedule deviation indicates that COCOMO prediction may be more accurate.
- Technical constraints probably have a great effect on the result of project.

Condition 3: If neither condition is met

- No clear signal; consider both perspectives.
- Other project-specific variables should be considered.

Application to Case Study:

- Profitability = 30% ($< 50\%$)
- Cost Overrun = 12% ($> 10\%$)
- $|24.3 - 20| = 4.3$ ($< 20 \times 0.25 = 5$)
- Neither condition is clearly met.

Resolution: In this case, there is no dominant signal to favour one prediction over the other. Both models have valid points depending on interpretation. The project demonstrates good financial indicators (30% profitability) even though schedule and cost overruns occur, which is why ML models predict success, while COCOMO predicts failure.

6 Discussion

6.1 Strengths of Machine Learning Approaches

1. Higher prediction accuracy: ML models performed more effectively (91–97% better accuracy) compared to COCOMO (76%).
2. Adaptability: ML models are capable of learning from new data and making predictions based on such data.
3. Feature importance analysis: ML models can determine the strongest factors that affect the project success.
4. Non-linear relationships: ML models can capture complex, non-linear relationships between variables.
5. Integration of multiple variables: ML methods can include technical and non-technical factors.

6.2 Strengths of COCOMO

1. Simplicity: Easy to understand and implement.
2. Interpretability: Simplified relationship between inputs and outputs.
3. Low computational requirements: Can be computed manually or using simple devices.
4. Industry acceptance: Proven and commonly used in software engineering.
5. Minimal data requirements: Can use with scanty historical data.

6.3 Limitations of the Study

1. Data limitations: The historical project data quality and quantity impact ML model performance.
2. Industry specificity: Results are likely to differ from one industry sector and type of project to another.
3. Limited qualitative factors: Other critical success factors in any project, such as team dynamics and stakeholder satisfaction, are hard to measure.
4. Model complexity: More complex ML models may have improved accuracy but poorer interpretability.

6.4 Implications for Project Management

The results have several implications for software project management:

1. Hybrid approach: Organisations may want to implement both ML and traditional methods for more comprehensive project evaluation.
2. Early risk identification: ML models can detect possible risks at an earlier stage of the project cycle.
3. Resource allocation optimization: More precise predictions support efficient distribution of resources and planning.
4. Focus on key predictors: Project managers should give particular attention to revenue generation, cost overrun percentage, and schedule adherence.

5. Data-driven decision making: Organisations should engage in systematic collection of data to enhance ML model performance over time.

7 Recommendations and Practical Applications

Provided with our research findings, we make the following recommendations regarding software project management:

7.1 For Project Managers

1. Implement hybrid prediction systems: Utilise both ML and traditional methods for all-round project evaluation.

2. Focus on key predictors: Take special care of revenue generation, cost overrun percentage, and schedule adherence.

3. Continuous monitoring: Adjust predictions as projects progress, and new data becomes available.

4. Context-aware approach: When planning, consider industry type, project complexity, and business objectives when selecting prediction methods.

5. Data-driven decision making: Make project decisions based on quantitative analysis, not intuition alone.

7.2 For Organizations

1. Invest in data collection infrastructure: Collect the full set of project metrics to enhance ML model performance over time.

2. Develop prediction capabilities: Develop internal capability in ML-based project prediction.

3. Training and development: Make sure that project managers know both traditional and ML-based approaches.

4. Knowledge sharing: Develop mechanisms for sharing experiences and outcomes of the projects across teams.

5. Process integration: Embed prediction models into current project management procedures.

7.3 For Researchers

1. Expand data diversity: Add more varied types and contexts of projects to be studied in the future.

2. Explore ensemble methods: Explore the prospects of such ensemble methods as Random Forest and XGBoost.

3. Incorporate qualitative factors: Create ways to incorporate qualitative factors, such as team expertise and stakeholder alignment.

4. Time-series analysis: Discover time-series analysis for the project trajectory over the period.

5. Tool development: Develop publicly available tools for implementing hybrid prediction strategies.

8 Conclusion

This research supports the notion that machine learning practices can substantially improve financial forecasting in software projects. Unlike the traditional cost models such as COCOMO, ML algorithms adjust dynamically to complex project variables and make more accurate predictions of project success/failure. The Support Vector Machine (SVM) model was the most accurate, with 97% accuracy, thus especially suitable for the prediction of financial outcomes. We also discovered that metrics such as revenue, cost overrun, and actual budget spent were more predictive of the success of a project than technical metrics such as lines of code. Our new decision framework to resolve prediction disagreements offers a systematic method of combining the strengths of traditional and ML-based methodologies. This hybrid approach offers the most holistic view of the issue of software project success prediction, integrating the interpretability and simplicity of COCOMO and the accuracy and adaptability of machine learning. Although ML models show higher predictive accuracy, traditional methods such as COCOMO do still offer valuable complementary insights, in particular in terms of technical feasibility and schedule constraints. An approach combining both methodologies is the most comprehensive view of software project success prediction, which will help to make better decisions and eventually improve project outcomes.

References

[1] Various Authors, *Comparative Analysis of Machine Learning Techniques in Financial Risk Assessment*, ResearchGate, 2023, https://www.researchgate.net/publication/390756294_Comparative_Analysis_of_Machine_Learning_Techniques_in_Financial_Risk_Assessment.