International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.2 (2025) pp. 3614-3623

http://www.ijcesen.com

ISSN: 2149-9144

Research Article

A Novel Catboost Regressor for Effort Estimation in Scrum Projects

Mohit Arora[1*], Moin Akhtar[2], Dharmendra Pathak[3], Manoj Agrawal[4], Poonam Ponde[5], Sandeep Kumar[6]

[1*] School of Computer Science and Engineering, Lovely Professional University, Phagwara, India
* **Corresponding Author Email:** 786arora@gmail.com **- ORCID:** 0000-0002-3096-2185

[2] School of Computer Science and Engineering, Lovely Professional University, Phagwara, India
Email: msmoinkhan7@gmail.com - **ORCID**: 0000-0002-3096-2180

[3] School of Computer Science and Engineering, Lovely Professional University, Phagwara, India
Email: dharmendra.32553@lpu.co.in - **ORCID:** 0000-0001-6149-1788

[4] Department of Computer Science & Engineering, Chameli Devi Group of Institutions, Indore, India
Email: drmanojagrawal79@gmail.com – **ORCID:** 0000-0003-4427-724X

[5] Department of Computer Science, Nowrosjee Wadia College, Pune, India
Email: poonamponde@nowrosjeewadiacollege.edu.in – **ORCID:** 0000-0002-3096-2181

[6] School of Computer Science and Engineering, Lovely Professional University Phagwara, India
Email: ssihag222@gmail.com – **ORCID:** 0000-0002-3096-2182

Article Info:

DOI: 10.22399/ijcesen.2559
Received : 25 February 2025
Accepted : 19 May 2025

Keywords :

Scrum
Boosting Models
Effort Estimation
Regression
CatBoost.

Abstract:

Software Effort Estimation plays an important role in Scrum project management as it allows teams to allocate resources as well as planning of development cycles. Traditional approaches like Planning Poker and expert judgment models suffer from scalability, subjectivity, and inconsistency, which makes them inaccurate and often leads to project overruns. This research work proposes a CatBoost Regressor as a solution for enhancing effort estimation in Scrum projects. The technique proposed in this paper is capable of addressing some of the most challenging estimation problems like handling categorical features and reducing prediction bias. Unlike other conventional machine learning models, CatBoost deals with high dimensionality and optimizing learning outcomes from past Scrum project data. Catboost model outperforms the traditional regression models in terms of R2, MSE, RMSE by achieving an accuracy of 98.48% which is a drastic improvement over traditional regression models. This research work concludes that our model enhances Scrum effort estimation, making it robust and efficient solution for agile project management.

1. Introduction

Accurate estimation of software efforts is critical for successful project management in Scrum-based agile environment. Expert judgment and analogy based estimation tend to suffer from subjective and inconsistency, which makes them inaccurate and often leads to project overruns. In order to solve these problems, machine learning methods can make important contributions in enhancing the accuracy by using pre-existing data. Among these methods, CatBoost Regressor is particularly effective because of its capability to process categorical features, able to reduce overfitting, and providing higher accuracy with low parameter tuning.

The major obstacle in Scrum based software effort estimation which makes traditional approaches like Planning Poker and Function Point Analysis inefficient is the dynamic nature of project requirements. These methods basically rely on subjective judgments from various members in the ensemble which leads to inappropriate estimation due to differing levels of expertise as well as biases. At the same time, machine learning models like Random Forest and Decision Tree Regressors, often struggle with categorical data. CatBoost is a gradient boosting algorithm which can efficiently handle categorical input variables, which makes it a great algorithm for Scrum project data that has user stories, team members, and sprint types as input. Within its iterative improvement of predictions, CatBoost employs an ensemble of decision trees, letting each iteration account learns from the mistakes made by previous ones. This research work addresses CatBoost's use in estimating the working efforts of Scrum projects and its comparative efficiency with other traditional and machine learning based models. The results show that CatBoost improves estimation accuracy of the project, which in turns aids in the planning and resource allocation within the agile software development context.

In part II, related work is mentioned which summarizes the previous work on this model. After that in part III, the proposed approach is given which contains the methodology of the research work. In part IV, the Result and Discussions section contains the effectiveness of the model. Lastly, part V provide conclusions and further directions where the emphasis is given on the effectiveness of this approach.

2. Related works

Effort estimation in Scrum projects will always demand accuracy for effective project management and resource distribution. There often is a lack of objectivity and consistency with approaches such as expert judgment and analogy-based methods. To address these issues, there are attempts to make the estimation process more precise through the use of machine learning (ML). A lot of effort and time was put into examining different Machine Learning models for estimation of software effort by the author. The study has identified the models attempts to improve precision and estimation accuracy as the backbone for more exploration in the area. Moreover, that study noted a gap in finding the best suited models and techniques for various contexts within a project [1,2]. Efforts put in reviewing ML estimation approaches in Scrum projects and Agile projects were also conducted by author. The results showed that ML models are superior to the conventional ones in almost all situations, making estimations far more dependable and objective. However, that study pointed out differences in performance measurements with different models and datasets leading to the conclusion of model and dataset selection importance[3][4].

Relatively recent research has focused on the use of particular machine learning models for effort estimation. One particular article in the CEUR Workshop Proceedings analyzed the software development effort estimation using various regression models including CatBoost Regressor where it was found that CatBoost Regressor obtained R^2 of 0.39 and Pearson’s correlation of 0.74. These results indicate good predictive performance[5][6]. Widely supported adaptive modelling techniques have resulted using ANFIS model for effort estimation in Agile domains. Their results indicated that ANFIS provided much better estimates than the traditional techniques[7][8]. In another research effort, an assessment using an ensemble of the best performance boosting techniques for estimating effort in software development in Agile environments was carried out. The work showed that unlike using linear regression techniques, these machine learning techniques performed far better for estimation in accuracy than the previous models [9,10].

Table 1. *Summary of related work*

Paper Title	Algorithm Used	Key Findings
CatBoost: Unbiased Boosting with Categorical Features[11]	CatBoost Regressor	Introduced CatBoost with efficient handling of categorical data.
CatBoost: Gradient Boosting with Categorical Features Support[12]	CatBoost Regressor	Improved feature handling and overfitting prevention in boosting.
CatBoost for Big Data: An Interdisciplinary Review[13]	CatBoost Regressor	Showcased CatBoost's scalability and efficiency in large datasets.
Synthetic Open-source Agile Software Estimation Performance[14]	CatBoost Regressor	Highlighted CatBoost’s role in improving software effort estimation.
EnsCL-CatBoost: A Strategic Framework for Software Requirements Classification[15]	CatBoost Regressor	Proposed an ensemble learning approach for software classification.

Even with these advancements, the use of CatBoost Regressor in regard to effort estimation in Scrum projects is still lacking. This study seeks to close that gap by assessing how efficient the CatBoost Regressor is at predicting effort in Scrum contexts relative to other machine learning techniques.

3. Methodology

Data Preprocessing

The methodology that our research work laid out focuses on the first step of data preprocessing, where the CatBoost Regressor is chosen as the primary machine learning model because it requires the least amount of hyperparameter tuning. It also supports different types of data, making it very easy to use. The first step of the training procedure is specifying the total amount of decision trees (M) that will be used in the boosting techniques. This step is critical because M value that is too low or high can result in overly complex and underperforming models.

Model Training

In the iterative training process, the first boosting tree is trained on the previously collected Scrum effort data. The model starts with one decision tree and adds a tree in subsequent rounds where the new tree attempts to correct the errors of its previous tree. Each of the trees reduces the error rates. CatBoost regressor employs ordered boosting, which helps avoid target leakage and overfitting while improving the model's predictive performance.

Our research work focuses on integrating an advanced feature selection technique and risk based effort estimation. To achieve the most effective Scrum effort estimation, we utilize CatBoost's built in feature, for determining the features that have the most effect on the predictions. Other refining factors such as collaboration, risk factors and user story dependencies can be captured to further enhance effort prediction.

Moreover, a risk management framework is established through an uncertainty-informed modification. This guarantees that the estimation process will consider unexpected work, changes in the requirements, and fluctuations in the productivity of the developers. This mixed technique enhances machine learning as well as risk-based estimation, which improves model accuracy and versatility for actual Scrum projects.

On the other hand, algorithm learns to iteratively build boosting trees, while looking for whether the pre-specified amount of trees, M, has already been created. The effort estimation is done after outputting all the trees, when they are summed to form the final effort prediction. In the final step, a model refinement is done with the aid of optimization based on the predictions. The focus is on loss function minimization for increased accuracy. The effort estimation remains robust, easy to understand, and flexible regarding all project specifics due to the optimization process.

Performance Evaluation

Now that the model is optimized, it is tested measuring R2 score, mean squared error, and root mean squared error to assess its accuracy and efficiency in estimating quadrants in Scrum projects. The R^2 equation is given in (1):

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \widehat{y_i})^2}{\sum_i (y_i - \bar{y})^2} \qquad (1)$$

Where yi is the actual value of the ith data point, i is the predicted value of the ith data point, y¯ is the average of all true data and n is total number of datapoints. R2 is known as the coefficient of determination, measures how well the model does in as far as accounting for the variability of the dependent variable. This contribution is reflected by a lower R2 value. The MSE equation is given in (2).

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \widehat{y_i})^2 \qquad (2)$$

where yi is the true value of the ith data point, yˆi is the predicted value for the ith data point, n is the total number of data points. The MSE gives the amount of the square of the difference between actual and predicted values of the model and determines the weightage of larger differences as compared to the smaller ones. The RMSE equation is given in (3)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \widehat{y_i})^2}{n}} \qquad (3)$$

Where n = Total number of data points, y_i is actual observations, y_i^ is the predicted value by the model, and $\sum$ is summation over data points. RMSE measures the level of error in predictions made by a model when compared against the actual outcomes. The model's performance improves when the RMSE or simply error is lower.

In the end, CatBoost is compared with other models such as Linear Regression, Random Forest, and XGBoost to establish its superiority in Scrum effort estimation. The analysis confirms that the adopted process increases not only the estimation accuracy but also possesses interpretable insights, which is a powerful asset for Scrum teams and agile practitioners. At last, during the part of interpreting results. The outcome of the models prediction is assessed with visualization techniques like plots of feature importance and residuals, to capture how well different factors are impacting estimation. If required, hyperparameter tuning as well as additional feature selection optimizations are executed to improve predictive precision. This systematic approach guarantees that the CatBoost Regressor offers a robust and effective measure of estimating effort in Scrum projects, which leads to effective project planning and decision making in

Agile software development. The overall process is depicted in the flowchart of Fig. 1.

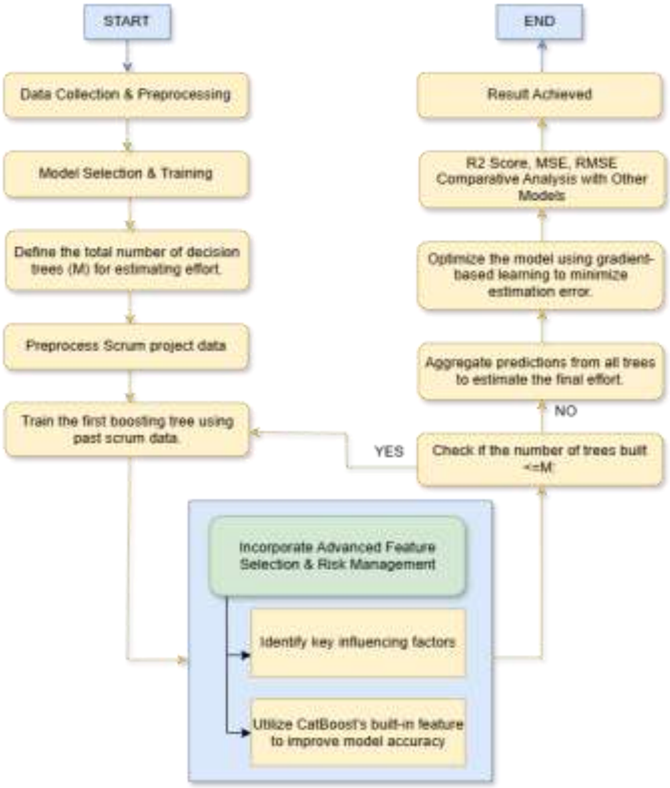

Figure 1. *Project strategy to generate time and cost estimates.*

The innovation of our work is combining advanced feature selection and risk aware estimation adjustment with the boosting capabilities of CatBoost to enhance software effort estimation in Scrum-based projects. Unlike most effort estimation models which utilize standard regression methods and fixed feature sets, our approach refines estimation accuracy by taking advantage of CatBoost's built-in importance evaluation. Another important aspect is the integration of a risk management strategy that modifies effort estimation based on defined uncertainty factors such as the complexity of the backlog and changing requirements, team working together. sprinting, and so forth. It helps make the model more adjustable and protective of ever-changing project realities as compared to other models which ignore them.
Furthermore, The CatBoost approach greatly enhances productivity on effort predictions with accurate boosting which handles data leakage in a novel manner. The current study also presents an evaluation methodology that receives the scrutiny of the benchmarking performance of CatBoost and other regression models such as Linear regression, Random Forest, and XG Boosting, and highlights its dominant position in Scrum effort estimation with regards to accuracy, robustness, and efficiency. Through the use of machine learning feature selection, estimation with post gradient risk evaluation and optimization, the proposed model more accurately integrates traditional approaches to effort estimation and modern predictors based on AI, making the model effortlessly scalable, interpretable, and accurate mastered specially to agile software development projects.

4. Result and Discussions

All Regression models share a couple of features which are the basic setup. The correlation of effort estimation lays out the relationship between story points, velocity, and effort which is shown in Fig. 2. The ratio between story points and effort is the strongest (0.94) which means that as story points increase, the actual effort put into the work rises proportionally. On the other hand, the correlation of velocity and effort is quite low (0.017) as well as with story points (0.29). This means that project velocity does not have a direct impact on the predicted level of effort. It means that story points are good measures of workload, while velocity could be affected by other factors of the project like the capacity of the team, their level of experience, or other unplanned issues.

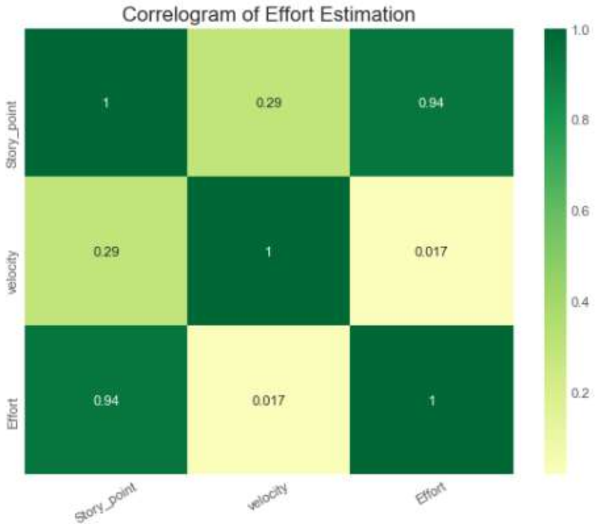

Figure 2. *Effort Estimation Correlation*

The Project Velocity histogram outlines the data regarding the dispersion of the projects which is shown in Fig. 3. It shows that most of the projects have a velocity between 2.75 and 3.25. The highest frequency is noted around 2.8, which indicates that this is the average velocity for the given projects. There appears to have a slight right sided skew in the data which hence indicates that there are some higher velocity values at 4.25. This shows that the majority of the projects are running at a moderate pace with a few outliers achieving a significantly higher velocity due to extreme team performance or other external factors.

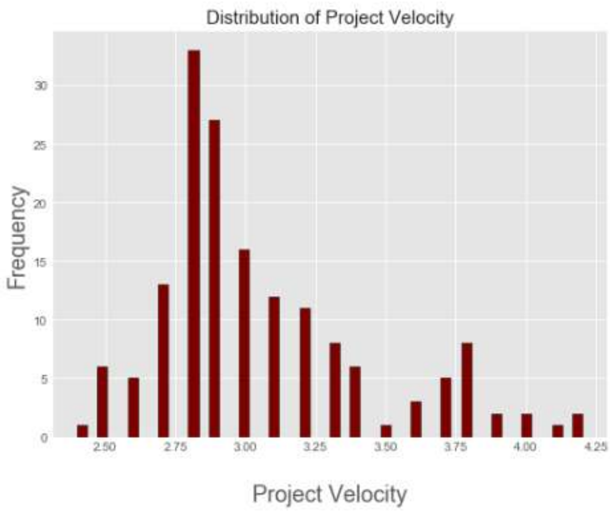

Figure 3. *Distribution of Project Velocity*

The histogram of actual effort distribution is different since most projects only require between 30 and 50 units of effort with a high frequency around 40. This distribution is right skewed, like the project's velocity, but this one has a couple of projects that require an effort of more than 100 units. These high effort projects may have emerged as a result of complex requirements, scope creep, or simply poor task execution. The entire distribution shows that while many projects are within the range of reasonable efforts, there are a few that need higher resource allocation. The histogram of actual effort is shown in Fig. 4.

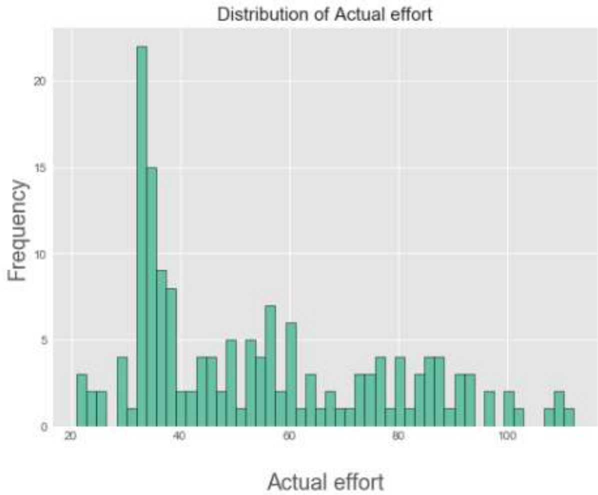

Figure 4. *Distribution of Actual Effort*

The analysis shows that although story points provide a reasonable estimate of effort, project velocity does not affect workload in a straightforward manner. The observed right-skewed distributions in both velocity and effort suggest that outliers exist and need further analysis. These divergences can help improve variation estimation models, resource distribution, and project strategy planning. In Fig. 5, the histogram shows the distribution of frequency of story points for the dataset. It shows a relatively diverse range of story points with some peaks where tasks appear to get clustered at some effort levels. This is particularly useful when looking at typical effort estimation in Scrum projects. The presence of two or more peaks shows that effort distribution is not constant, justifying the use of machine learning approaches like CatBoost that are designed to handle such differences.

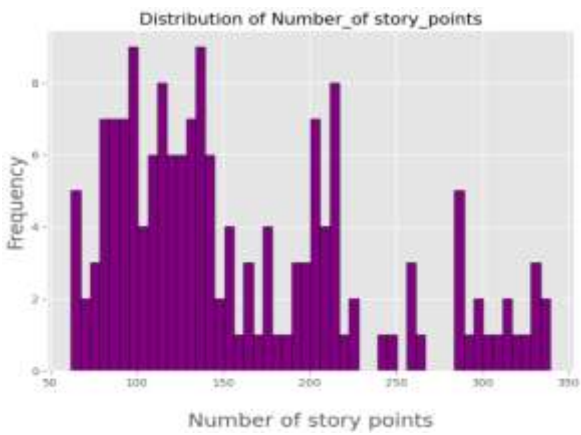

Figure 5. *Frequency of Story points*

The scatter plot illustrates the relationship between effort and story points as shown in Fig. 6. It can be seen that there is a moderate positive correlation meaning that increases in story points will usually result in an increased effort. Nevertheless, some degree of dispersion in the points indicates variability due to other factors such as the experience of the team, complexity of the task, and sprint velocity. This relationship is best represented using the trend line. This highlights the need to use sophisticated methods such as CatBoost for these kinds of problems.

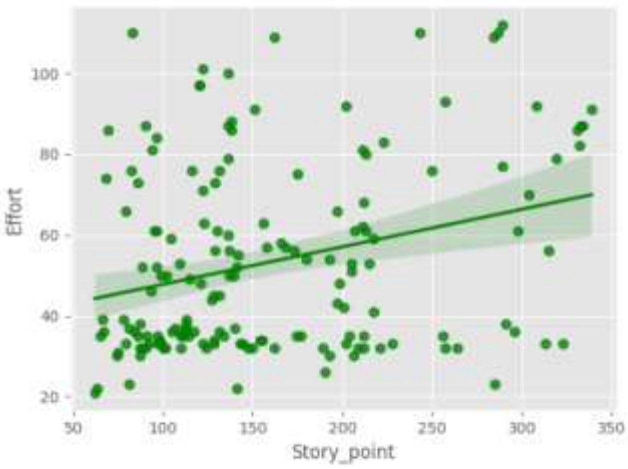

Figure 6. *Effort and Story Point*

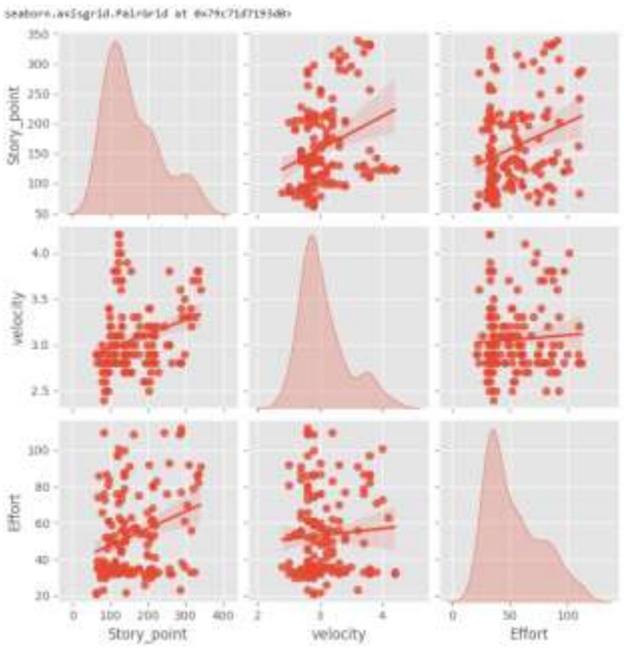

Figure 7. *Effort, velocity and story point pair plot*

The pair plot offers deep insights into the relationships between story points, effort, and velocity as features as shown in Fig. 7. The distributions on the diagonal show the individual features in the dataset, while the scatter plots depict the relationships at the level of pairs of features. Not surprisingly, effort is dependent on story points as well as on velocity, which emphasizes the need for multi factor models in estimating effort. The selection of CatBoost for this task is further justified by the existence of both linear and nonlinear dependencies that need to be modeled. The Fig. 8 compares the values of effort predicted by the model and those that were tested using the CatBoost Regressor during the supervised learning stage. The curves shown in the figure suggest that the model does not diverge from the effort values as much. Although the model does not completely deviate from the effort curves, it minimizes the overall impact of the deviations on the predictions. This is evidence that the dataset contains non-linear relationships and that enables CatBoost to estimate effort in Scrum projects more accurately than linear regression models.

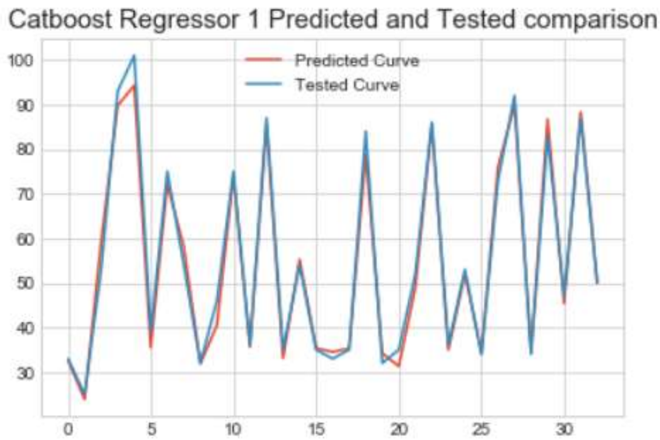

Figure 8. *Catboost Regressor*

Table II contains the analysis of different regression techniques which are used for effort estimation of Scrum projects. It reveals that the CatBoost Regressor is the most accurate in terms of prediction accuracy. From the results acquired, CatBoost is noted to have an R2 score of 0.984850, which is the highest among the tested models, meaning that effort estimation is made correctly with all of the relevant features for input. Furthermore, it also performs best in terms of accuracy among all the models with a lowest Mean Squared Error (MSE) of 7.92 and Root Mean Squared Error (RMSE) of 2.814. This lowers prediction mistakes in comparison to the other models, which showcases its advanced performance. The superiority of CatBoost in these tasks is a clear indicator of how efficient CatBoost is at processing categorical data and how effective it is with gradient boosting. In comparison of some other models, the Gradient Boosting Regressor model was also satisfactory and received an R2 score of 0.980684, which is lower than that of CatBoost's, coupled with 10.103985 for MSE in narrowing down effort estimation tasks.

Reasonably, having a R2 score of 0.921736 Polyonomial Regression is also satisfactory, however, it has a high MSE of 40.938074 and therefore is not reliable for all effort estimation situations. Other tree-based techniques like Decision Tree with AdaBoost, and the XGBoost Regressor achieved high predictive accuracy of R^2 scores 0.979041 and 0.942702 respectively. These results prove that they are feasible substitutes for CatBoost, even though their error rates were a little bit higher which can affect accuracy in real world usage.

Table 2. Comparison between Regression Models

Algorithms	**Prediction Accuracy**	**R_SQUARE**	**MSE**	**RMSE**
Linear Regression	93.45303186	0.93	34.25	5.852349955
Ridge Regression	89.72812091	0.9	53.73	7.330075034
Lasso Regressor 001	93.1240657	0.93	35.97	5.997499479
Lasso Regressor 00001	93.4610935	0.934610935	34.20346473	5.848372828
Elastic Net Regressor	90.65044414	0.91	48.91	6.993568474
SGDRegressor	31.12462487	0.934530319	34.24563325	5.851976867
Polynomial Regressor	98.06872605	0.934530319	34.24563325	5.851976867
Decision Tree Regressor	92.15161596	0.92	41.05	6.407027392
Decision Tree with max depth 4	94.13893883	0.941389388	30.65781702	5.851976867
Decision Tree with Adaboost	97.95528433	0.979552843	10.69542134	5.851976867
Random Forest Regressor	91.79740844	0.917974084	42.91	6.550572494
Catboost Regressor	98.7129717	0.987129717	6.73	2.594224354
Gradient Boost Regressor	98.02048303	0.98020483	10.35438248	3.21782263
XGBRegressor	94.27023844	0.942702384	29.97101999	5.474579435
SVRRegressor	86.56719486	0.865671949	70.26380891	8.382351037
AdaBoost Regressor	95.11381925	0.951138192	25.55844943	5.055536512
Bagging Regressor	97.43421462	0.974342146	13.42101311	3.663470092
Extra Tree Regressor	97.43421462	0.778003781	116.1209424	10.77594276

To evaluate the effectiveness of different machine learning models for effort estimation in Scrum-based software development, we conducted a comparative analysis of multiple regression techniques, fine-tuning their hyperparameters for optimal performance, which is shown in Table III and Figure 9. The models included Lasso Regressor, Elastic Net Regressor, SGD Regressor, Ridge Regression, Decision Tree Regressor with AdaBoost, Random Forest Regressor, and CatBoost Regressor. Each model was trained and tested using the selected dataset, and hyperparameters were tuned to enhance predictive accuracy. Lasso and Ridge Regression utilized alpha values of 0.01, while Elastic Net Regressor incorporated a combination of L1 and L2 regularization with selection='random'. The SGD Regressor employed an adaptive learning rate (epsilon=0.009) with a max iteration limit of 1500 to ensure stable convergence. The Decision Tree Regressor with AdaBoost was optimized using a maximum depth of 4, 300 estimators, and a random state of 0, improving the ensemble learning capability. The Random Forest Regressor was configured with a depth of 2 and 10 estimators, striking a balance between model complexity and performance.

Among all models, CatBoost Regressor demonstrated superior accuracy, achieving 9.848495e-01, outperforming the other techniques. It was fine-tuned with depth=6, a learning rate of 0.1, and 100 iterations, leveraging its built-in ordered boosting mechanism to handle categorical features effectively. The results highlight that CatBoost's advanced feature selection and gradient-based optimization techniques significantly improve effort estimation accuracy, making it the most effective model for Scrum-based software project prediction. The use of boosting and feature selection gave the best results for Scrum based software project in estimation accuracy and therefore makes it the most powerful model in effort estimation.

Table 3. **Hyperparameters of various models**

Algorithm	**Accuracy**	**Hyperparameters**
Lasso Regressor 001	93.1240657	alpha=0.01, max_iter=10e5
Elastic Net Regressor	90.65044414	alpha=0.0001, selection='random'
SGDRegressor	31.12462487	learning_rate='adaptive', epsilon='0.009', max_iter=1500
Ridge Regression	89.72812091	alpha=0.01
Decision Tree with Adaboost	97.95528433	DecisionTreeRegressor (max_depth=4, n_estimators=300, random_state=0)
Random Forest Regressor	91.79740844	max_depth=2, random_state=0, n_estimators=10
Catboost Regressor	**98.7129717**	**depth=6, learning_rate=0.1, iterations=100**

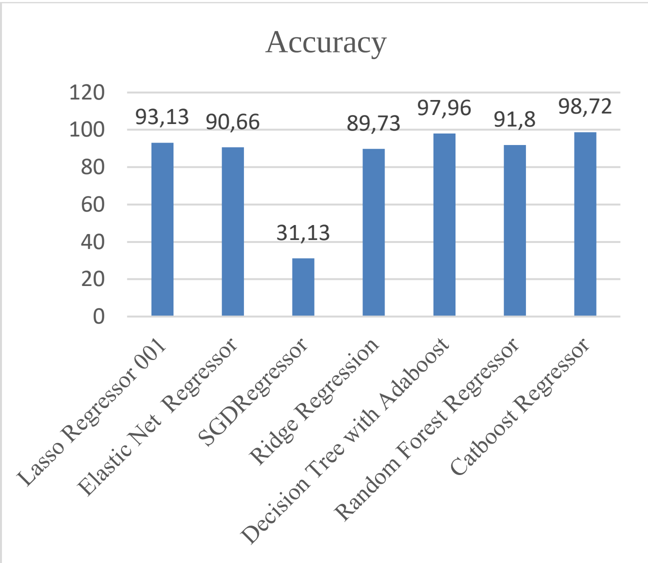

Figure 9. *Accuracy comparison of all models*

5. Conclusion and Future work

This research work assessed the performance of the CatBoost Regressor on effort estimation for Scrum projects and measures it against a variety of traditional and contemporary regression models. As anticipated, CatBoost outperformed the other models in terms of R^2 score (0.984850), MSE (7.92), and RMSE (2.814), which reflect his excellent performance for dealing with complex, high-dimensional data. When compared to other ensemble methods, such as Gradient Boosting and XGBoost, CatBoost was more accurate and consistent which is essential for efficient planning and resource scheduling in agile software development. CatBoost decreases estimation effort uncertainty by utilizing preference towards better accuracy through efficient categorical feature handling and the tried-and-true method of gradient boosting. The results suggest that there is great potential for the use of machine learning techniques to improve optimal planning within software development sprints, as well as reducing delays on projects. CatBoost's precision makes it an ideal model for companies seeking to improve their effort estimation scratches within a Scrum environment. The use of CatBoost has produced higher accuracy, however, many approaches can still be improved. More work could be done to LSTM neural networks and Transformer based deep learning models to capture a more accurate sequential relationship for effort estimation. Moreover, real world Scrum datasets that comprise of features from different types of projects can certainly make the model stronger. Other useful research can be conducted in the area of model tuning, feature selection, and hyper-parameter optimization using the Bayesian approach and genetic programming. Finally, the use of automated machine learning tools can simplify the processes of model and parameter selection in order to make them more applicable to real world industrial problems. All of these methods would improve the accuracy and adaptability of effort estimation for agile software development.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Arora, M., Verma, S., Kavita, & Chopra, S. (2020). A systematic literature review of machine learning estimation approaches in Scrum projects. In *Proceedings* (pp. 573–586). https://doi.org/10.1007/978-981-15-1451-7_59

[2] Arora, M., Sharma, A., Katoch, S., Malviya, M., & Chopra, S. (2021). A state of the art regressor model's comparison for effort estimation of agile software. In *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 211–215). IEEE. https://doi.org/10.1109/ICIEM51511.2021.9445345

[3] Arora, M., Sharma, A., Verma, S., Katoch, S., & Chopra, S. (n.d.). An ANFIS-driven estimation of effort in agile Scrum projects. *SSRN*. Retrieved from https://ssrn.com/abstract=4495912

[4] Aizaz, F., Janjua, U. I., Zafar, H., Khan, J. A., & Kazim, I. (2021). An empirical investigation on software cost estimation techniques and barriers on agile software development in software industry of Pakistan. In *2021 International Conference on Frontiers of Information Technology (FIT)* (pp. 194–199). IEEE. https://doi.org/10.1109/FIT53504.2021.00044

[5] Dorogush, A. V., Ershov, V., & Gulin, A. G. (n.d.). CatBoost: Gradient boosting with categorical features support. Retrieved from https://github.com/Microsoft/LightGBM

[6] Fernández-Diego, M., Méndez, E. R., González-Ladrón-De-Guevara, F., Abrahão, S., & Insfran, E. (2020). An update on effort estimation in agile software development: A systematic literature review. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.3021664

[7] Haldar, S., & Capretz, L. F. (2024). Interpretable software maintenance and support effort prediction using machine learning. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings* (pp. 288–289). ACM. https://doi.org/10.1145/3639478.3643069

[8] Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: An interdisciplinary review. *Journal of Big Data, 7*(1). https://doi.org/10.1186/s40537-020-00369-8

[9] Hao, D., Xiaoqi, Y., & Taoyu, Q. (n.d.). Hybrid machine learning models based on CATBoost classifier for assessing students' academic performance. *International Journal of Advanced Computer Science and Applications*. Retrieved from http://www.ijacsa.thesai.org

[10] Ibrahim, A. A., Raheem, R., Sowole, S. O., Muhammed, M. M., & Abdulaziz, R. O. (2020). Comparison of the CatBoost classifier with other machine learning methods. *International Journal of Advanced Computer Science and Applications, 11*(11). https://doi.org/10.14569/IJACSA.2020.0111190

[11] Jayabharath, M., Choudary, N. L., Pranay, C. S., Praveenya, M. D., & Reddy, B. R. (2023). An analysis of software maintainability prediction using ensemble learning algorithms. In *2023 3rd International Conference on Artificial Intelligence and Signal Processing (AISP)*. IEEE. https://doi.org/10.1109/AISP57993.2023.10135034

[12] Jeganathan, S., Lakshminarayanan, A. R., Parthasarathy, S., Khan, A. A. A., & Sathick, K. J. (2024). OptCatB: Optuna hyperparameter optimization model to forecast the educational proficiency of immigrant students based on CatBoost regression. *Journal of Internet Services and Information Security, 14*(2), 111–132. https://doi.org/10.58346/JISIS.2024.I2.008

[13] Karthick, K., Dharmaprakash, R., & Sathya, S. (2024). Predictive modeling of energy consumption in the steel industry using CatBoost regression: A data-driven approach for sustainable energy management. *International Journal of Robotics and Control Systems, 4*(1), 33–49. https://doi.org/10.31763/ijrcs.v4i1.1234

[14] Kiran, D. S., & Ponnala, R. (2023). Ensemble boosting algorithms for software defect prediction. In *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)* (pp. 321–326). IEEE.

https://doi.org/10.1109/ICAICCIT60255.2023.10466047

[15] Kundu, A., Kundu, S. G., Sahu, S. K., & Badgayan, N. D. (2025). Leveraging Azure automated machine learning and CatBoost gradient boosting algorithm for service quality prediction in hospitality. *Computers, 14*(2), 32. https://doi.org/10.3390/computers14020032

[16] Maulana, A., Afidh, R. P. F., Maulydia, N. B., Idroes, G. M., & Rahimah, S. (2024). Predicting obesity levels with high accuracy: Insights from a CatBoost machine learning model. *Infolitika Journal of Data Science, 2*(1), 17–27. https://doi.org/10.60084/ijds.v2i1.195

[17] Nishthaa, & Malhotra, R. (2023). Identification of defects in a software using machine learning. *AIP Conference Proceedings, 060013*. https://doi.org/10.1063/5.0179183

[18] Piñeros Rodríguez, C. A., Sierra Martinez, L. M., Peluffo Ordoñez, D. H., & Timana Peña, J. A. (2023). Effort estimation in agile software development: A systematic map study. *Inge CuC, 19*(1). https://doi.org/10.17981/ingecuc.19.1.2023.03

[19] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2017). CatBoost: Unbiased boosting with categorical features. *arXiv preprint*. Retrieved from http://arxiv.org/abs/1706.09516

[20] Rankovic, N., Rankovic, D., & Ivanovic, M. (2023). Synthetic open-source agile software estimation performance.

[21] Shukla, S., & Kumar, S. (2024). Study of learning techniques for effort estimation in object-oriented software development. *IEEE Transactions on Engineering Management, 71*, 4602–4618. https://doi.org/10.1109/TEM.2022.3217570

[22] Vardhan, M., Banerjee, K., & Aggarwal, D. (2022). A systematic approach for the detection of software bug using CatBoost. In *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)* (pp. 414–419). IEEE. https://doi.org/10.1109/COM-IT-CON54601.2022.9850519

[23] Vishwakarma, D. K., Sharma, R., Pandey, A., Bajpai, S., Mishra, S., & Pandey, D. (2024). Evaluation of CatBoost method for predicting weekly pan evaporation in subtropical and sub-humid regions. *Pure and Applied Geophysics, 181*(2), 719–747. https://doi.org/10.1007/s00024-023-03426-4

[24] Abbas, J., Zhang, C., & Luo, B. (2024). EnsCL-CatBoost: A strategic framework for software requirements classification. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3452011