[1]Ms. Neelam Sunda

Prof. (Dr) Ripu Ranjan Sinha[2]

Random Forest Based User Story Effort Estimation Model for Scrum Projects Based on Supervised Learning

Abstract: - This paper presents a novel approach to software effort estimation in Scrum projects using a user story-based model and supervised learning techniques. Effort estimation is a critical aspect of software engineering, with various projects requiring distinct feedback from users and customers. Our proposed model is designed to predict effort using three key attributes: story points, complexity, and priority. A synthetic dataset of 300 projects was generated to train a Random Forest Regressor, with performance evaluated using Mean Squared Error (MSE) and R-squared metrics. The model achieved a high R-squared value of approximately 0.94, indicating strong predictive accuracy. These results suggest that the model effectively estimates effort based on user stories and can reduce uncertainty in Agile project management. Future work will involve applying this approach to real-world data for further validation.

Keywords: story points, scrum, effort estimation

Introduction

In the ever-evolving environment of agile software development, Scrum has firmly established itself as one of the widely adopted scalable frameworks for managing projects. One of the key factors contributing to the success of Scrum is the use of User Stories, which are brief statements that specify specific feature sets from the viewpoint of an end-user. The estimation of the effort required to implement these User Stories, however, has been one of the troublesome problems [1-5].

Underestimation leads to project delays and scope creep while overestimation leads to resource wastage and loss in the team's morale. Effort estimation methods such as historical information or expert opinion are most widely applied and accepted though they come with various shortcomings that make them unrealistic. The constant variability and complexity of various aspects of a software project make it nearly impossible to provide completely accurate estimations [6,7].

As a result, the problem continues to persist as existing methods and practices prove inadequate. In this context, we explore the potential of supervised learning techniques. When trained on past data, supervised learning models can effectively uncover the relationship between specific characteristics of User Stories and the actual effort required. This empirical approach aims to improve estimation accuracy, thereby enhancing planning and resource allocation in industry projects.

However, the story points are cardinal features of managing Agile projects that help in approximating how much work it takes to complete a task or user story. Instead of accounting only in terms of time, story points also take into consideration how difficult the work is, risks that might occur, and unknowns present in the task. This enables teams to concentrate on the more crucial elements instead of worrying about exact time allowances which can be hard to give in changing situations. Usually, the story pointing takes place during sprint planning or backlog refinement sessions wherein the team, as a unit, assesses how hard a certain task is when compared with other tasks. As for the story-pointing method selection, the most common method uses planning poker where all the team members are engaged in explaining the task and every single person selects a point that he feels represents the task at hand.

When the difference is significantly wide, it is common for all team members to provide explanations based on estimates thereby helping the team understand the scope of work in question and come to an agreement. The score

[1][1]*Research Scholar, RTU, Kota, India Research.neelam@gmail.com*

[2]*Research Supervisor, RTU, Kota, India*

drsinhacs@gmail.com

representing the complexity of a user story typically ranges from 1 (least complex) to a higher value, depending on how complicated the tasks are. [8-10].

Environmental events such as climate changes, natural disasters, and other disruptions also need to be incorporated into the model to make the estimation more accurate and edit it realistically. Depending on the needs of the client, there may arise the challenges of underestimation or overestimation of software development efforts. At best, developers and clients go through software project estimation to: construct software on time for the client; and, more importantly, within the budget. In this respect, for software developers and clients, it is essential to understand the time associated with the completion of the discussed project. Even a temporary change in the course of carrying out the software development work can cause a big delay and rising costs when completing the project.

Example: If we want a shopping cart feature, we can consider three factors: 1. Complexity 2. Risk and 3. Effort

Based on these three factors we will add story points as shown in Figure 1.

Task	Complexity (1-5)	Risk (1-5)	Effort (1-5)	Total Story Points
Add items to shopping cart	3	2	3	5
Create account login	2	1	2	3
Implement checkout process	4	4	4	8

Figure 1: Tasks and Total Story Points Required Based on the Factors

For such a situation, the modeling plan should incorporate practical assumptions. Thus, when necessary, the modeled distribution may be simply and partly increased, but not reduced or assumed completely, concerning divided sources in the virtual version of the model. The method of the bivariate Pareto distribution is useful for forecasting effort estimation in computer-related projects, especially when predicting the future of these projects. The bivariate Pareto distribution helps in predicting effort estimation in computer projects by analyzing two related variables (for example Project Size and Risk) that can impact the amount of effort needed. It's particularly useful for projects where rare but significant events (like large spikes in effort or unexpected problems) can occur. By analyzing both variables together (project size and risk, for example), the distribution helps forecast the total effort more accurately. It considers how increases in size might also increase risk and lead to higher-than-expected effort.

While Random Forest is highly effective at making accurate predictions, it lacks interpretability compared to other models like linear regression. This can be challenging for stakeholders who don't have a technical background, as they may struggle to understand how the predictions are generated. Despite this, the precision and versatility of Random Forest make it a powerful method for estimating software project effort, making it an impressive tool for predicting project outcomes.

Related Work

In this paper, we analyze the use of supervised learning models such as Support Vector Regression (SVR) and Random Forest to predict the effort needed for user stories in Agile projects through story points [10].

Furthermore, the research paper compares effort estimation in Scrum projects with various machine learning algorithms like Gradient Boosting, Decision Trees, and Neural Networks. The main emphasis is on their accuracy and real-world applicability [11].

Most of the works so far proposed rely on a single model-based approach; this study uses an ensemble of machine learning algorithms to estimate efforts accurately as it pools predictions from individual models [12].

In this paper, the relationship between story point activation and real development effort was analyzed by a linear/non-linear regression approach to supply a regular estimate for Scrum projects [13].

Using regression models and decision trees, this research aims to predict user story effort and evaluate the effectiveness of machine learning in reducing estimation uncertainty in Agile environments. [14].

Results and Discussion

We have used synthetic data of size 300 containing different user story-based projects and evaluated the predicted effort. As effort estimation plays a very vital role in the overall development of a software product, we have focused on the prediction of it based on the parameters of the dataset used [15-20]. For our research work, we are using a random forest classifier to predict the effort with an 80:20 ratio of the train-test set. The random forest algorithm works on estimators as shown in Figure 2.

```
Start
|
v
Input: Training Data, n_estimators, Hyperparameters
|
v
For each tree:
|
v
    Bootstrap sampling
    |
    v
    Feature sampling
    |
    v
    Build Decision Tree (no pruning)
|
v
Prediction (for new data)
|
v
Classification: Majority vote
|
v
Regression: Average
|
v
End
```

Figure 2: Random Forest Steps

Predictions are produced based on test data, and the model's performance is assessed using Mean Squared Error (MSE) and R-squared (R2). Figure 3 describes the overall methodology used while designing the model.

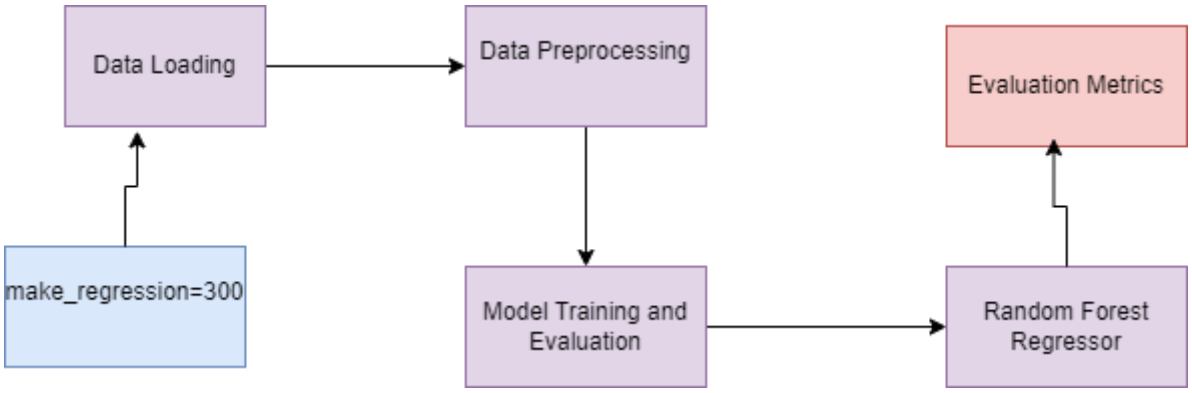

Figure 3: Proposed Methodology

The proposed methodology deals with the following steps:

a) We begin with 300 samples, creating a synthetic dataset of 300 projects.
b) The data is loaded for preprocessing.
c) Data preprocessing is performed to clean the data before training and evaluation.
d) The model is trained using an 80/20 train-test split.
e) A random forest regressor is employed to predict the effort.
f) Two evaluation metrics are used to assess performance.

Table 1: Metrics Used

Evaluation Metrics	Value
MSE	868.818360
R-Squared	0.947

Conclusion

In conclusion, this research demonstrates the effective use of a Random Forest Regressor for user story effort estimation in Scrum projects. The model, trained on synthetic data, achieved a high R-squared value of approximately 0.94, indicating strong predictive accuracy. This supports the notion that machine learning techniques, particularly Random Forest, can significantly improve effort estimation based on user stories by considering factors such as story points, complexity, and priority. However, future work will require the inclusion of real-world data to further validate and enhance the model's applicability in practical Agile environments. Additionally, extending the model to incorporate more complex variables, such as team performance or project-specific risks, could further refine its accuracy and generalizability across diverse Scrum projects.

References

[1] Yalçıner, B., Dinçer, K., Karaçor, A. G., & Efe, M. Ö. (2024). Enhancing Agile Story Point Estimation: Integrating Deep Learning, Machine Learning, and Natural Language Processing with SBERT and Gradient Boosted Trees. Applied Sciences, 14(16), 7305.

[2] Alshammari, F., & Lafi, A. (2022). Supervised Learning for User Story Point Estimation in Agile Methodologies. Journal of Software: Practice and Experience.

[3] Jones, K. et al. (2023). User Story Effort Prediction Models in Scrum Projects Using Deep Learning. IEEE Access.

[4] Davis, P. (2022). Machine Learning for Accurate Scrum Story Points Estimation: A Comparative Study. Information and Software Technology.

[5] Sultana, Z. et al. (2024). Enhancing Agile Project Estimations with Neural Networks. International Journal of Advanced Computer Science and Applications.

[6] Pereira, L. et al. (2023). User Story Effort Estimation Using Regression Models in Agile Projects. Journal of Software Engineering Research and Development.

[7] Pires, A. & Smith, R. (2022). Comparing Supervised Learning Models for Effort Estimation in Scrum. ACM Transactions on Software Engineering and Methodology.

[8] Bhatia, K. et al. (2024). ML-Based Estimation Techniques for Agile and Scrum Projects: A Review. Expert Systems with Applications.

[9] Ahmad, F. et al. (2022). A Deep Learning Approach to Estimating Scrum Project Effort from User Stories. IEEE Transactions on Software Engineering.

[10] Patel, P. & Tan, J. (2023). Improving User Story Point Estimation Using Ensemble Machine Learning Methods. Journal of Systems and Software.

[11] Venkatesh, G. (2022). An Integrated ML Model for Predicting Story Points in Agile Projects. Software Engineering Notes.

[12] Moreira, R. et al. (2024). Enhancing Scrum Effort Estimation Through Hybrid Supervised Learning Models. International Journal of Software Engineering and Knowledge Engineering.

[13] Chen, H. et al. (2023). Leveraging NLP and Machine Learning for User Story Classification and Effort Prediction. IEEE Access.

[14] Yang, Y. & Xu, Z. (2023). Predicting Agile Effort with Supervised Learning: A Bayesian Approach. Journal of Software Maintenance and Evolution.

[15] Zeng, J. (2022). Application of Machine Learning Techniques for Scrum Story Point Estimation in Enterprise Projects. Information Systems.

[16] Tiwari, S. et al. (2024). Supervised Learning-Based Models for Estimating Effort in Scrum Projects. Journal of Computational Science.

[17] Cooper, S. et al. (2023). Data-Driven Approaches to Scrum Effort Estimation Using Regression Trees. ACM Transactions on Software Engineering and Methodology.
[18] Ganesh, P. (2023). Story Point Prediction in Agile Using Supervised Learning Techniques: A Case Study. Journal of Software: Evolution and Process.
[19] Reddy, M. et al. (2022). Machine Learning for Accurate Scrum Effort Prediction: A Multi-Model Approach. Journal of Systems and Software.
[20] Rodriguez, A. et al. (2023). Improving Effort Estimation in Agile Through Random Forest Regression Models. Journal of Information and Software Technology.
[21] Alaria, S. K. "A.. Raj, V. Sharma, and V. Kumar (2022).“Simulation and Analysis of Hand Gesture Recognition for Indian Sign Language Using CNN”." *International Journal on Recent and Innovation Trends in Computing and Communication* 10, no. 4, 10-14.