Effort and cost Estimation of Agile software development

By:

- Sandeepak Singh.
- Harishita Rana.
- Tarang Sharma.

Under Supervision of:

- Mr. Prateek Srivastava

Agile Software Development

Introduction

Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it.

When you approach software development in a particular manner, it’s generally good to live by these values and principles and use them to help figure out the right things to do given your particular context.

12 Agile Principles

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

- Continuous attention to technical excellence and good design enhances agility.

- Simplicity–the art of maximizing the amount of work not done–is essential.

- The best architectures, requirements, and designs emerge from self-organizing teams.

- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Effort and Cost Estimation

Effort Estimation in Agile

Estimated effort is a practice in agile software development where teams estimate the relative size of a task or product backlog item based on how much effort it will take to complete it.

Cost Estimation in Agile

Estimating costs in an Agile environment requires a more iterative, integrated, and collaborative approach than in traditional acquisition programs. Contrary to the myth that Agile is an undisciplined approach that downplays cost aspects, cost estimation is a critical activity in programs that use Agile practices.

Effort and Cost Estimation Techniques

There are multiple techniques used for effort and cost estimation .

1. **Expert-based:**
   - Planning Poker
   - Expert Judgement
   - Wideband Delphi

2.**Data-based:**
   - Machine Learning
   - Neural Network
   - Functional Size Measurement Regression
   - Algorithmic Methods
   - Fuzzy Logic
   - Swarm Intelligence
   - Bayesian Network
   - Monte Carlo
   - Statistical Combination
   - Principal Component Analysis COCOMO II

3. **Combination-based:**
   - Use Case Point
   - Change Effort Prediction
   - Ontology Model
   - Experience Factory
   - Prioritisation of Stories

Techniques used in Research

1. Ada Boost
2. Decision Tree
3. Random Forest
4. Ensemble Learning
5. Ensemble Learning with Grid Search CV

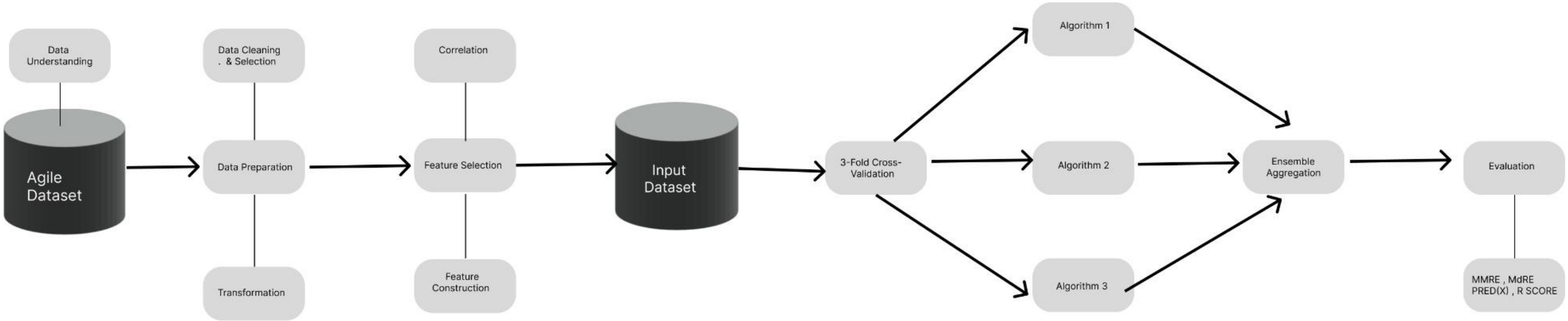
Dataset used in Research

Zia Dataset(21 agile projects)

from

Ziauddin, S., Tipu, K. & Zia, S. An effort estimation model for agile software development. Adv. Computer. Sci. Appl. 2(1), 314–324 (2012)

Methodology :



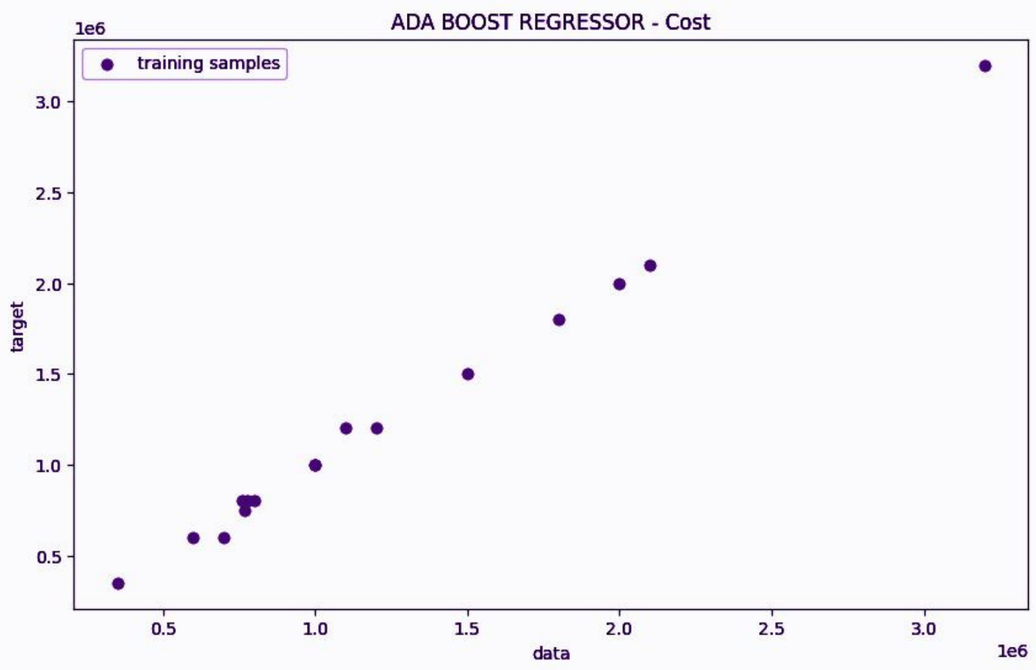
Results

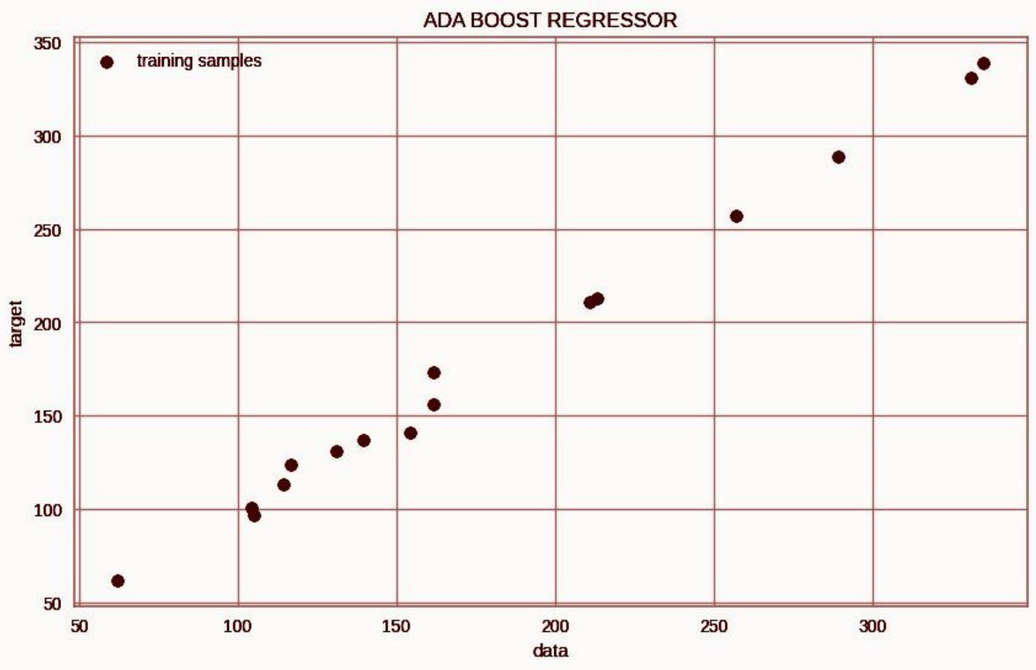
The execution of the experiments occurs independently, which means that first estimates are gathered from the decision tree, then estimates from the random forest and AdaBoost, finally, Ensemble Learning. All algorithms used the same data and the same configuration. Every trained model gives an accuracy greater than 80% and a coefficient of determination of at least 0.90.

Model training and evaluation comes in three parts. The first one takes the training set created from data augmentation and runs the cross-validation. It splits the dataset into a training set and a validation set, which helps to check the accuracy of training with the subset taken for validation.

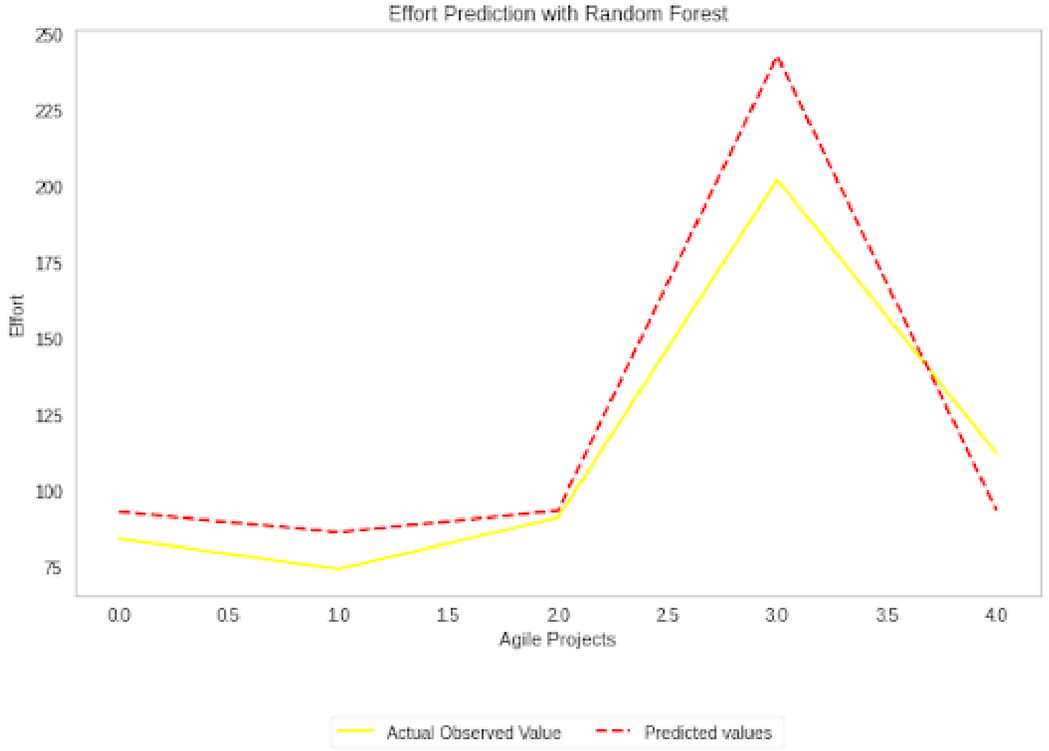
Once the model is trained, the set of 21 real projects is fed as an input and the model generates new estimates. This is the testing set, and it produces the final predictions for the trained model, which will be used to create the final estimations. The training phase, testing were assessed with the accuracy, coefficient of determination, mean squared error, root of mean squared error, mean relative error and variance. This resulted in tables for every technique that group the metrics for each phase.

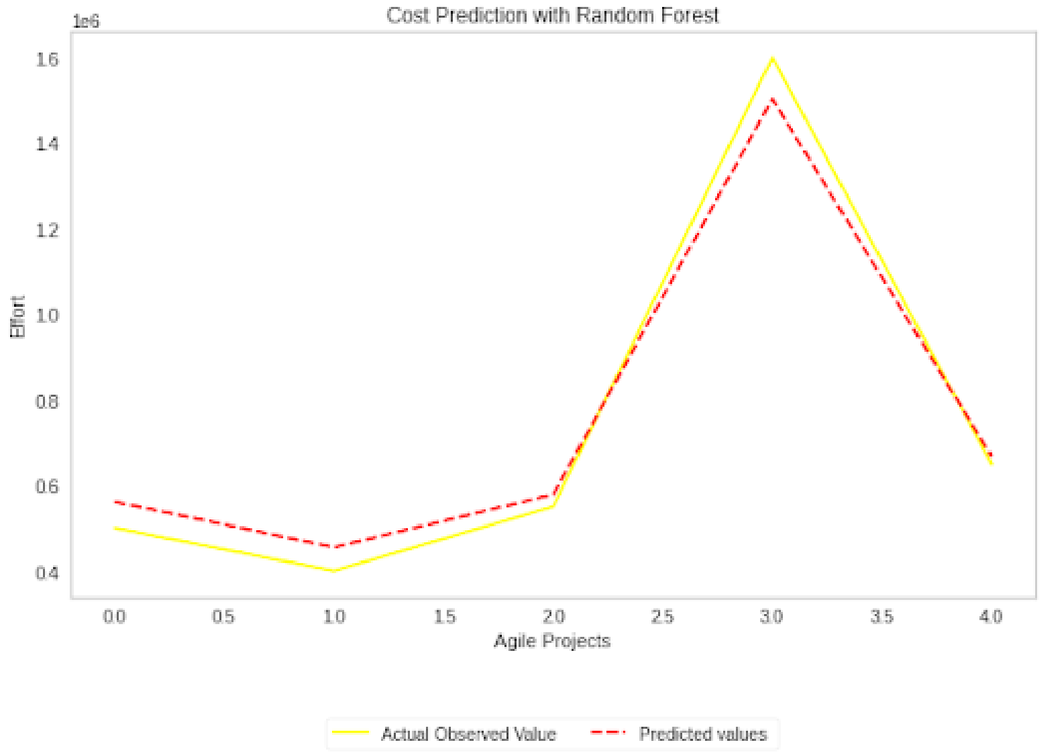
Ada boost


ADA BOOST REGRESSOR - Cost
1e6
training samples
target
data
1e6
0.5
1.0
1.5
2.0
2.5
3.0

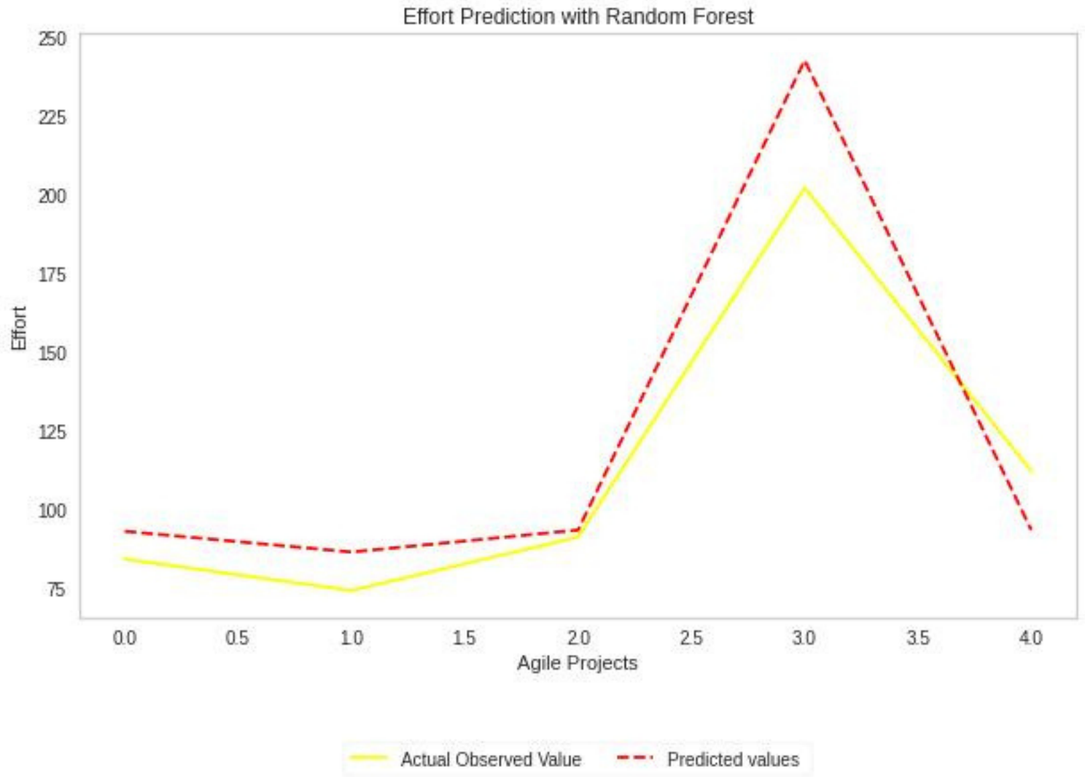

ADA BOOST REGRESSOR
training samples
target
data
50
100
150
200
250
300
350

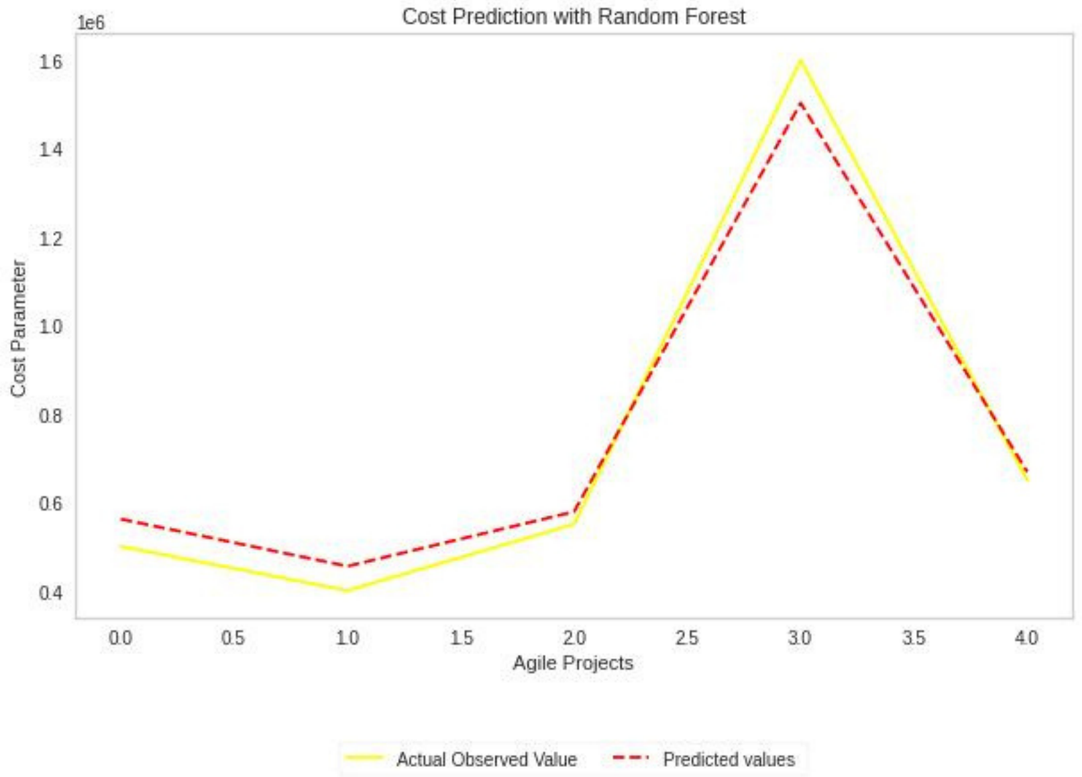
Decision Tree



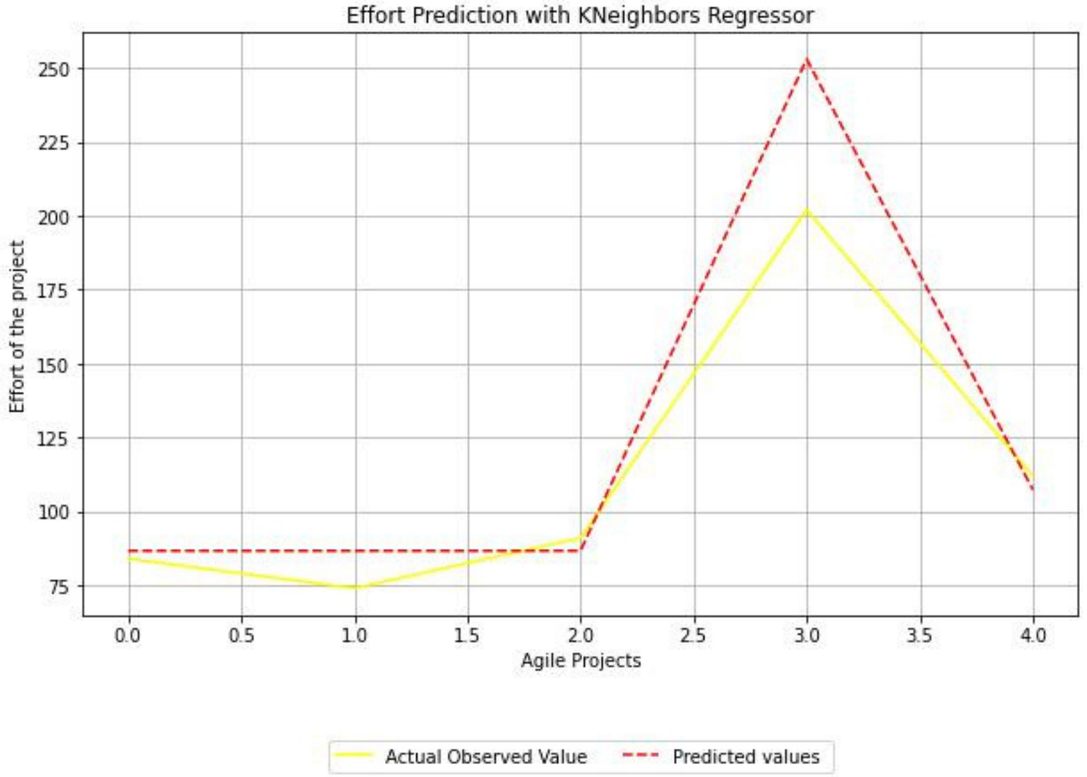


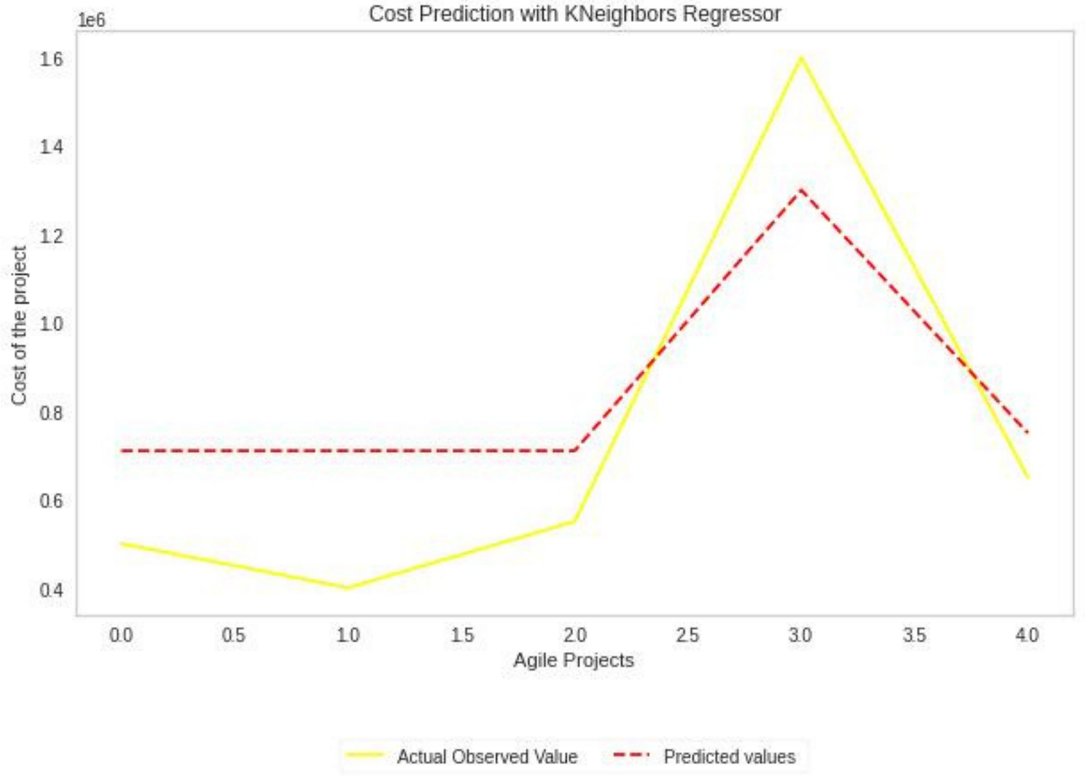
Random Forest





Ensemble With Grid Search





Future Work

These studies have been analysed with the purpose of understanding their findings in an attempt to shed light on various issues, and particularly on the set of issues that have arisen
during the period selected. With regard to the integration of the findings, we considered that comparing the results of both subsets of primary studies was more interesting than merging
them to present an overall result regarding all the papers on this subject published from 2018 to the present.

Our results show that effort estimation methods have been used in four different agile methods (i.e., Ada Boost ,Decision Tree, Random Forest , Ensemble Learning)

In spite of the vast number of approaches, the accuracy of software effort estimation models for agile development is still inconsistent. The results of our study highlighted that accuracy remains a challenge in most of the papers analysed. However, we have identified significant improvements in this respect. First, an increasing number of papers report adequate ranges of accuracy values. Second, of the papers that report the accuracy metric used, an increasing percentage also reflect aspects concerning the validation of the models. Third, those papers that report the statistical significance of the different accuracy values have also begun to report the effect size.

Future Work

In future work, we plan to apply the use of ANN Artificial Neural Networks , we plan to investigate the influence of cost factors on the accuracy of the estimation models used in Zia. We particularly wish to assess to what extent the use of specific cost factors may imply more accurate estimates in particular contexts. We also plan to perform incremental updates in order to follow the evolution of effort estimation methods and cost factors in Zia dataset. Extension to this procedure might be made by applying other machine learning techniques such as Extreme Learning Machine and Bayesian Networks on the Agile related dataset.

Conclusion

Software estimation helps for proper planning, management and estimation of the total efforts that will be used for implementing, testing and delivering the desired product to the customers in
terms of time within the specified deadlines.

Most of the research were done in software development estimation but there are only few of them which focuses on agile software development , thus our research try to find out some of the best approaches that can be used for estimation of agile software development .

Machine learning is the major way of estimating the agile software development project and thus we have focused on machine learning algorithms in our research and tried to find out some of the best algorithms for estimation after comparing their performances through various evaluation criteria.