Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 173 (2020) 335–344

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on *Smart Sustainable Intelligent Computing and Applications* under ICITETM2020

Predictive Analysis of Manpower Requirements in Scrum Projects Using Regression Techniques

Kamna Vaid[a*], Udayan Ghose[b]

[a,b]*USICT, Guru Gobind Singh Indraprastha University, Dwarka,New Delhi- 110078, India*

Abstract

Flexible iterative development life cycle, adaptive nature and fast delivery has given Agile an upper edge as compared to all other software development frameworks. In the current industry scenario agile methods are gaining popularity, owing to its people centric approach, hence organizations are adopting agile development methodologies at a large scale. Agile projects work in self-organizing small collaborative teams. Team size varies according to the project requirement however, agile development focus on smaller team size. Supervised machine learning is applied in this study to provide optimum prediction model. All the available regression models in Matlab R2019b are used to predict number of team members required for an agile project. Iterations from five different open source projects are considered for this study. The results after training all the variants of each regression model, namely Linear Regression models, Support Vector Machine models, Tree models, Ensemble models and Gaussian Process Regression models are compared using Root Mean Square Error (RMSE) score and R-squared values. On the basis of evaluative and comprehensive analysis, the most significant model to predict manpower requirement for an agile project has been chosen.

Keywords: Scrum; Agile projects; Team size; Regression models

1. Introduction

Software project management deals with all the aspects of project development with lots of challenges and issues. Planning of whole project gets effected by many uncertainties [13]. One of the major key drivers in any type of development is Human resource. Resource allocation and collaboration between the team members is a major

*Corresponding author. E-mail address: kamnavaid@gmail.com

10.1016/j.procs.2020.06.039

challenging factor in planning and management of a software project especially in case of a large project. The widely known agile practice, Scrum is capable of handling such challenges. In this paper we have given Regression models which describes the manpower requirements for a project depending upon the information provided in the data set taken from five Scrum projects.

Sutherland et al. [17] explained Scrum as a process which adds life to team of agile project. It provides accuracy, energy, focus and transparency to the software development team. Agile is established on principles that lay emphasis on self-organization with regular improvements, customer benefits, periodic incremental and quick delivery, profound and unified associations within small teams, stated by the Agile Manifesto (2011) [1].

Scrum is a people centric framework; people have utmost importance in agile project over process and technology [4]. Project team has three roles- Scrum master, Product owner and development team [15]. In Scrum projects smaller team size is considered. Continuous interaction and collaboration amongst the team members is encouraged through everyday Scrum meetings. Though how to fix the team size is a major challenge.

The main aim for this study is to find best prediction model that will help to predict team size of a scrum project iteration wise, based on the team's velocity. Velocity is defined as the amount of work done by a team in a Sprint. Velocity depends on various factors such as team expertise and experience, communication amongst peers, complexity of the project etc. Each sprint has a small team to do the assigned task. We have used supervised machine learning approach to choose the best prediction model by using Regression methods. All the regression models are compared on the basis of evaluation metrics.

The paper is organized as follows. Section 2 describes the literature review done on agile projects. Section 3 explains about the dataset and its source. Section 4 represents the research methodology and about performance measures considered for this study. Section 5 illustrates the results that we obtained after training our data set in a tabulated format with their brief explanation. Section 6 pertains to conclusion of the study.

2. Literature review

Teams are a most important building block of any software project. Generally, team size in agile project is small, independent, self-organizing and have the decision-making power in deciding how to proceed for work and how work will be allocated. Agile processes need to build backlogs, to control and measure every task to be done. They create a project team in a certain manner [23].

Scrum master has a significant role in building agile team. A suitable Scrum team should comprise of five to nine team members concluded by Cohn [10]. Owing to their work experience on such projects most of the professionals suggested that large team size should be avoided. With large team size it is difficult to have successful agile project development. While with small team size it is more efficient to manage inter team communication as well as coordination. Generally, nine people are considered as a good number for basic agile process.

Jeff Sutherland recommended a group of seven people as a team in Scrum development project. Many successful agile projects with larger team size, with 120 or 250 team members are also there in the industry [3]. Fried [7] considers team size should not be greater than 10. Hence, in general we can say that team size for an average agile project should be between 8 to 10 but it may exceed according to the need of the project.

Effort estimation and Requirement Engineering helps in determining project requirements. There are two types of effort estimation methods namely, model-based methods and expert-based methods. Statistical regression models are widely referred in model-based methods while in expert-based methods, machine learning techniques like Support Vector Machines (SVM), Decision Trees (DT), Artificial Neural Networks (ANN), Bayesian Networks (BN), Genetic Algorithms (GA) etc. are used. Multiple studies have been conducted on the team's psychological perspective and team work as well [20].

Gren et al. [5] has recommended iterative development and their retrospectives has a strong correlation with levels of group maturity. They concluded that the behavior of agile teams varies at different development stages. Zia et al. [21] has focused on number of individuals rather than complete team. With the help of a survey, author has given single person overall effect on the agile project and many factors when team size become larger. They suggested that team size should be less than 25. There are different views of different practitioners according to their work experience on agile projects. Various studies done earlier have focused at project level rather at each iteration level. This study has focused on projects at four different iterations level. At each iteration a regression model can be analyzed for predicting manpower of a project.

3. Dataset

Data for this study is collected from the publicly available dataset online at https://github.com/SEAnalytics/datasets/tree/master/agile%20sprints/IEEE%20TSE2017/dataset [22] created by Choetkiertikul. M [2]. This dataset has iterations (also known as Sprints) and their associated issues from Scrum projects only. It comprises of iterations from five large open source projects - Apache, JBoss, JIRA, MongoDB and Spring. Every iteration contains different number of issues which needs to be resolved in that iteration. In total it has 3,834 iterations and 56,687 issues are taken from these projects. Data for each iteration of a project is collected by the author at a prediction time, which is taken as reference point for identification of features of an iteration during different prediction times i.e. when the iteration commences and moving to 30%, 50% and 80% of its schedule according to the planning of the project.

The data set for each iteration level from all the five open source projects, are downloaded as comma separated values (csv) files. Four iterations files are taken from each project. All these files contain features of iterations from four different prediction time 0%, 30%, 50% and 80% of the iteration's planned duration. These files are named as project_iteration_prediction time e.g. for Apache project, files extracted from the dataset are apache_iteration_0.csv, apache_iteration_30.csv, apache_iteration_50.csv, apache_iteration_80.csv. In our study we have conducted experiment on the features of an iteration. These characteristics describes three major aspects of an iteration, i.e. the team, the elapsed time and amount of work [2]. All the features given below are taken by the author [2] according to the prediction time taken by him. Features of an iteration includes duration of an iteration, number of issues at start time, velocity at start time, added velocity, to-do velocity, number of issues added, number of issues removed, removed velocity, number of to-do issues, number of in-progress issues, number of done issues, in-progress velocity, done velocity, scrum master and scrum team members. Given Table 1 summaries the data set from all the five projects in brief.

Table 1. Summary of dataset from all the five projects.

Project	Number of Iterations	Number of Issues	Number of team members Min/Max	Repository
Apache	348	5826	3/21	[24]
JBoss	372	4984	2/20	[25]
JIRA	1,873	10,852	2/12	[26]
MongoDB	765	17,528	2/30	[27]
Spring	476	17,497	3/15	[28]

4. Methodology

Machine learning methods and techniques have been applied to train our dataset for finding most significant regression model for predicting number of team members for a Scrum project. Instead of choosing one or two models we have trained our dataset to all available regression models, in parallel, with their default settings using Regression learner app in Matlab 2019b. In this supervised learning, Number of team members is taken as Response variable. Since Scrum is an incremental and iterative agile software development method [9], so prediction is done at different levels in an iteration.

Our focus is restricted to predicting number of team members required at a time in a particular iteration according to the to-do velocity at that iteration level. Velocity is the summation of issues to be done by the prediction time. The number of team members varies across the five projects taken. For each project all the iteration files are extracted from the data set.

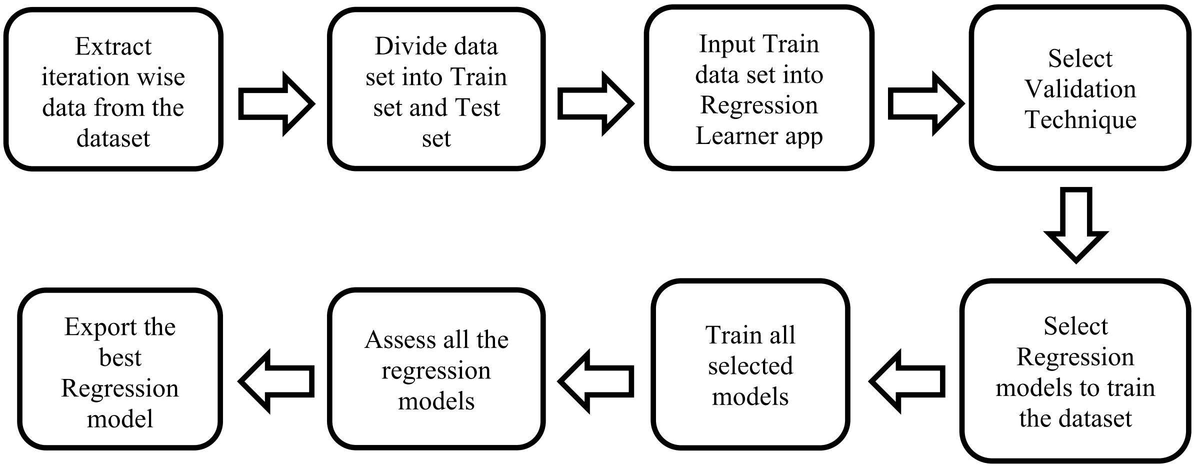
Extract iteration wise data from the dataset
Divide data set into Train set and Test set
Input Train data set into Regression Learner app
Select Validation Technique
Select Regression models to train the dataset
Train all selected models
Assess all the regression models
Export the best Regression model

Figure 1. Workflow of choosing the best regression model.

The figure 1 explains workflow of our study. Iteration file from four different times of each of the project was extracted from the dataset. Then this dataset is split into train set and test set. From each Iteration file 70% of the data set is used for training our predictive model. Before training our model, the most common statistical method K-fold Cross validation approach is applied to protect our model from the problem of overfitting. This technique splits the data into disjoint sets or folds. For each set it trains the model and assesses its performance using in-fold data. For all the folds or sets average test error is calculated. For smaller data sets this approach is very good. All above steps are executed K times in this approach.

In Regression Learner app by default 5- Fold Cross Validation is used. Hence, our data set is partitioned into 5 subsets. This data set is trained using five Regression models in the Regression Learner app using all its 18 features. Most dominant models are given in Table 3. These models are evaluated on the basis of 4 performance measures namely, RMSE, R-squared, MSE and MAE. Model with the best performance is chosen in each iteration for evaluation on the test data set. The best regression model is exported for further analysis.

4.1 Performance measures*: (a) Root Mean Square Error (RMSE)* is an approach for calculating the error of a model in prediction of data. It measures the sample standard deviation of differences between predicted and observed values. Formally, it is defined as follows:

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

where,
n = number of observations
$\widehat{y_1}, \widehat{y_2}, \ldots\ldots, \widehat{y_n}$ are predicted values
$y_1, y_2, \ldots\ldots y_n$ are observed values

RMSE is an effective measure to determine how accurately the model predicts the response. It is the most significant criterion for fit, of any prediction model. RMSE values are positive values as these depicts the root of Mean Squared Error (MSE).

(b) Mean Squared Error (MSE) describes how close is the regression line to our data set points. The distance between regression line and data points are called errors. These errors are squared and then average of these errors is calculated. Smaller value of MSE is considered better as it gives the line of best fit.
Formally, MSE is defined as,

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \tag{2}$$

where,
n = number of observations
$\widehat{y_1},\ \widehat{y_2}, \ldots\ldots, \widehat{y_n}$ are predicted values
$y_1, y_2, \ldots\ldots\ y_n$ are observed values

(c) R-squared is another statistical measure used to show how close our data points are to the fitted regression line. It is the coefficient of determination. R-squared represents variation in response variable given by the linear model. Its percentage lies between 0% and 100%. Higher value of R-squared is considered significant, as higher value indicates goodness of fit for our data set. Regression predictions perfectly fit the data when R-squared is equal to 1. R^2 is calculated as:

$$R^2 = 1 - \frac{\Sigma(y_i-\hat{y})^2}{\Sigma(y_i-\bar{y})^2} \tag{3}$$

where,
$\hat{y}$ = predicted value of y
$\bar{y}$ = mean value of y

(d) Mean Absolute Error (MAE) is an assessment metric used with regression models. It gives the closeness of the predicted and eventual outcomes. MAE is calculated as:

$$\text{MAE} = \sum_{i=1}^{n}\frac{|\ \widehat{y_i} - y_i\ |}{n} \tag{4}$$

where,
n = number of observations
$\widehat{y_1},\ \widehat{y_2}, \ldots\ldots, \widehat{y_n}$ are predicted values
$y_1, y_2, \ldots\ldots\ y_n$ are observed values

5. Results

Best Performance measures of fitted models using various supervised learning techniques, iteration wise for each of the five software are given in Table 3. These techniques are chosen based on the results of RMSE and R-squared values. RMSE and R-squared values are taken for comparing all the trained regression models. There are large variations in the values of RMSE. The models with too high RMSE values are not considered favourable for a significant prediction model. Always smaller values of RMSE are considered better. In all the following tables, models with the smallest RMSE scores are chosen for each software. R-squared which is coefficient of determination is taken as a second measure for choosing the best model. R-squared values closer to 1 are considered ideal, hence models with such values are taken. Few models have given negative R-squared values such models are considered worst.

The following table 2 has given different ranges of RMSE score amongst all the five software projects iteration wise. For example, for JIRA project values of RMSE lies in the range of 0.59 to 7.73 for Iteration 0, 0.61 to 1.66 for Iteration 30, 0.73 to 4.30 for Iteration 50 and 0.77 to 3.69 for Iteration 80. For Spring projects values of RMSE lies in the range of 1.25 to 42.15 for Iteration 0, 1.21 to 20.91 for Iteration 30, 1.331 to 10.23 for Iteration 50 and 1.454 to 24.15 for Iteration 80.

Iteration wise range of RMSE values for each of the software are:

Table 2. Summary of range of RMSE from all the five projects Iteration wise.

Project	Iteration 0	Iteration 30	Iteration 50	Iteration 80
Apache	1.76 to 42.2	1.99 to 48.9	1.72 to 4206	1.86 to 167.37
JBoss	1.69 to 13.83	2.31 to 22.35	2.27 to 30.01	1.84 to 29.14
JIRA	0.59 to 7.73	0.61 to 1.66	0.73 to 4.30	0.77 to 3.69
MongoDB	1.52 to 223.9	1.28 to 111.8	1.29 to 77.59	1.072 to 50.13
Spring	1.25 to 42.15	1.21 to 20.91	1.331 to 10.23	1.454 to 24.15

On our dataset we have applied all 4 Linear regression models namely, Linear Regression, Interactions linear regression, Robust linear regression and Stepwise linear regression to all the 5 different software. The Table 4 given below represents the best Linear Regression model for Apache, JBoss, JIRA, MongoDB and Spring projects. Except Spring which has Stepwise linear regression as the best linear regression method, all other projects have Linear Regression model with smallest RMSE values. R-squared values ranges from 0.15 to 0.67 for all the iterations of Linear Regression model.

Using Support Vector Machines (SVM), we have evaluated our data set by training all the SVM regression models available in the Matlab i.e. Linear, Quadratic, Cubic, Fine Gaussian, Medium Gaussian and Coarse Gaussian. The best models for each of the 5 projects are given in Table 5. There is a variation of SVM models amongst different software. Apache has Quadratic SVM model, JBoss and MongoDB has Linear SVM model whereas JIRA and Spring has Medium Gaussian as the prime regression variant. R-squared values ranges from 0.19 to 0.85 for all the iterations of SVM model. For spring software, Medium Gaussian SVM has given best R-squared value of 0.85.

In the similar manner, Table 6, Table 7 and Table 8 represents the best Tree, Ensemble and Gaussian Process Regression models. All the three variants of tree namely, fine tree, medium tree and coarse tree are used for training our dataset. Fine tree has given best values as compared to others. But for Apache project, all the three categories have given a negative R-squared values. Hence, tree model is not at all significant for Apache projects prediction on this dataset. For spring software, Fine tree has given best R-squared value of 0.82.

Both Ensemble bagged and boosted tree regression techniques have been applied on the dataset and results have shown that Boosted trees are better in terms of performance. None amongst the five software has significant results with bagged tree regression method. All the performance measures of boosted trees are given in Table 7. Using Gaussian Process Regression (GPR), data set is trained for Squared Exponential Gaussian Process Regression, Matern 5/2 Gaussian Process Regression, Exponential Gaussian Process Regression and Rational Quadratic Gaussian Process Regression (RQGPR). Amongst all Rational Quadratic Gaussian Process Regression have given the significant RMSE and R-squared values as compared to all the regression techniques applied on this dataset. R-squared values ranges from 0.55 to 0.96 for all the iterations. 0.96 signifies that this model explains the best variability of the response data around its mean. It explains the fitted regression line has maximum data points on it. For MongoDB software Rational Quadratic Gaussian Process Regression has given best R-squared values between 0.9 to 0.96.

Table 3 - BEST MODELS for Apache, JBoss, JIRA, MongoDB, Spring Projects

		Iteration 0				Iteration 30				Iteration 50				Iteration 80			
Software Projects	Regression models	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE
Apache	RQGPR*	1.76	0.63	3.098	1.175	1.999	0.55	3.999	1.38	1.814	0.61	3.292	1.226	1.867	0.59	3.487	1.247
JBoss	SVM+	1.782	0.62	3.178	1.111	2.316	0.66	5.366	1.169	2.538	0.6	6.445	1.368	1.846	0.75	3.41	1.058
JIRA	RQGPR	0.595	0.82	0.355	0.284	0.634	0.83	0.402	0.297	0.73	0.8	0.533	0.326	0.778	0.8	0.605	0.343
MongoDB	RQGPR	1.557	0.9	2.424	0.542	1.283	0.94	1.646	0.458	1.295	0.93	1.679	0.539	1.071	0.96	1.148	0.488
Spring	RQGPR	1.346	0.85	1.813	0.863	1.214	0.91	1.474	0.822	1.33	0.9	1.77	0.871	1.453	0.89	2.113	0.883

*RQGPR - Rational Quadratic Gaussian Process Regression
+SVM - Support Vector Machine

Table 4 - LINEAR REGRESSION MODELS

		Iteration 0				Iteration 30				Iteration 50				Iteration 80			
Software Projects	Linear regression model	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE
Apache	Linear	2.583	0.20	6.673	1.797	2.764	0.15	7.638	2.026	2.298	0.38	5.281	1.780	2.396	0.33	5.740	1.741
JBoss	Linear	1.864	0.58	3.476	1.143	2.882	0.48	8.305	1.514	3.221	0.36	10.377	1.428	2.480	0.56	6.149	1.334
JIRA	Linear	0.830	0.65	0.689	0.484	0.923	0.65	0.851	0.516	0.951	0.66	0.905	0.555	1.010	0.67	1.021	0.591
MongoDB	Linear	3.555	0.49	12.636	0.687	3.227	0.61	10.415	0.646	2.968	0.64	8.812	0.669	4.036	0.37	16.285	0.696
Spring	Stepwise linear	2.042	0.66	4.171	1.438	2.697	0.56	7.271	1.509	2.400	0.67	5.762	1.560	2.748	0.61	7.554	1.648

Table 5 - SUPPORT VECTOR MACHINE MODELS (SVM)

		Iteration 0				Iteration 30				Iteration 50				Iteration 80			
Software Projects	SVM	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE
Apache	Quadratic	1.987	0.52	3.949	1.263	2.692	0.19	7.248	1.622	1.727	0.65	2.985	1.197	1.92	0.57	3.689	1.286
JBoss	Linear	1.782	0.62	3.178	1.111	2.316	0.66	5.366	1.169	2.538	0.6	6.445	1.368	1.846	0.75	3.41	1.058
JIRA	Medium Gaussian	0.926	0.56	0.857	0.515	1.024	0.56	1.05	0.597	1.113	0.53	1.24	0.647	1.129	0.59	1.276	0.639
MongoDB	Linear	3.762	0.42	14.157	1.676	3.126	0.63	9.775	1.481	3.326	0.55	11.07	1.617	3.683	0.47	13.57	1.706
Spring	Medium Gaussian	1.686	0.77	2.845	1.157	1.586	0.85	2.515	1.157	1.732	0.83	3.001	1.212	1.747	0.84	3.052	1.191

Table 6 - TREE MODELS

		Iteration 0				Iteration 30				Iteration 50				Iteration 80			
Software Projects	TREE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE
Apache	Fine Tree	2.392	0.17	5.719	1.683	2.626	0.03	6.896	1.826	2.311	0.26	5.342	1.689	2.755	-0.02	7.594	1.917
JBoss	Fine Tree	2.831	0.18	8.016	1.487	2.726	0.35	7.434	1.553	2.697	0.37	7.274	1.534	3.1	0.21	9.611	1.667
JIRA	Fine Tree	0.863	0.56	0.744	0.495	0.962	0.56	0.926	0.558	1.021	0.54	1.042	0.581	1.097	0.54	1.205	0.637
MongoDB	Fine Tree	2.367	0.76	5.601	1.035	2.446	0.76	5.986	1.115	2.767	0.7	7.658	1.143	2.89	0.69	8.353	1.2
Spring	Fine Tree	1.713	0.76	2.932	1.009	2.122	0.73	4.504	1.239	1.781	0.82	3.173	1.084	2.128	0.77	4.528	1.121

Table 7 - ENSEMBLE MODELS

		Iteration 0				Iteration 30				Iteration 50				Iteration 80			
Software Projects	ENSEMBLE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE
Apache	Boosted Trees	1.992	0.43	3.97	1.35	2.168	0.34	4.703	1.553	1.95	0.47	3.805	1.413	2.421	0.21	5.864	1.68
JBoss	Boosted Trees	2.683	0.26	7.199	1.389	2.428	0.48	5.898	1.401	2.271	0.55	5.159	1.355	2.558	0.46	6.546	1.449
JIRA	Boosted Trees	0.814	0.61	0.663	0.489	0.924	0.59	0.855	0.569	0.966	0.59	0.933	0.591	1.029	0.6	1.059	0.629
MongoDB	Boosted Trees	2.324	0.76	5.403	1.061	2.346	0.78	5.504	1.097	2.732	0.71	7.464	1.156	2.849	0.7	8.118	1.179
Spring	Boosted Trees	1.256	0.87	1.579	0.83	1.5	0.86	2.25	1.006	1.47	0.88	2.162	0.99	1.636	0.86	2.678	1.008

Table 8 - GAUSSIAN PROCESS REGRESSION (GPR) MODELS

		Iteration 0				Iteration 30				Iteration 50				Iteration 80			
Software Projects	GPR	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE	RMSE	R-Squared	MSE	MAE
Apache	RQGPR*	1.76	0.63	3.098	1.175	1.999	0.55	3.999	1.38	1.814	0.61	3.292	1.226	1.867	0.59	3.487	1.247
JBoss	RQGPR	1.696	0.66	2.877	0.993	2.432	0.63	5.919	1.142	2.379	0.65	5.661	1.219	2.007	0.71	4.029	1.129
JIRA	RQGPR	0.596	0.82	0.355	0.284	0.635	0.83	0.403	0.297	0.730	0.80	0.533	0.327	0.778	0.80	0.606	0.344
MongoDB	RQGPR	1.557	0.9	2.424	0.542	1.283	0.94	1.646	0.458	1.295	0.93	1.679	0.539	1.071	0.96	1.148	0.488
Spring	RQGPR	1.346	0.85	1.813	0.863	1.214	0.91	1.474	0.822	1.33	0.9	1.77	0.871	1.453	0.89	2.113	0.883

6. Conclusion

After training of dataset using Matlab Regression Learner app, all the regression models namely, Apache, JBoss, JIRA, MongoDB and Spring were evaluated. The comparison of all the regression methods on the basis of RMSE and R-squared scores have shown that Rational Quadratic Gaussian Process Regression is the most significant model for prediction of number of team members for a Scrum project. Table 3 has shown the Performance Analysis of the best models with all four performance measures, RMSE score, R-squared, MSE and MAE values iteration wise. Though we have considered two factors RMSE and R-squared for comparing all the models. The lowest value for RMSE and R-squared value closer to 1 is tabulated for all the regression models in the given tables. We have evaluated our models on the basis of number of team members only, though there are many other factors that influence the selection and prediction of team members such as characteristics of team members, their work habits, team performance etc. In future all such factors for team analytics can be explored.

References

[1] Beck, Kent, *et al.* (2001): 2009 "The agile manifesto." Website. http://www.agilemanifesto.org.

[2] Choetkiertikul, Morakot, Hoa Khanh Dam, Truyen Tran, Aditya Ghose, and John Grundy. (2017) "Predicting delivery capability in iterative software development." *IEEE Transactions on Software Engineering* **44**, (**6**): 551-573.

[3] Cockburn, Alistair, and Jim Highsmith. (2001) "Agile software development, the people factor." *Computer* **34** (**11**): 131-133.

[4] Drury, Meghann, Kieran Conboy, and Ken Power. (2012) "Obstacles to decision making in Agile software development teams." *Journal of Systems and Software* **85**, (**6**): 1239-1254.

[5] Gren, Lucas, Goldman, Alfredo and Jacobsson, Christian (2019) “Agile Ways of Working: A Team Maturity Perspective”.

[6] Hamed, Amani Mahdi Mohammed, and Hisham Abushama. (2013) "Popular agile approaches in software development: Review and analysis." *In 2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEEE)*, pp. 160-166. IEEE

[7] L. Fried, “When Bigger Is Not Better: Productivity and Team Size in Software Development,” *Software Engineering Tools, Techniques, Practice* **2, (1):** 15-25.

[8] Lalsing, Vikash, Somveer Kishnah, and Sameerchand Pudaruth. (2012) "People factors in agile software development and project management." *International Journal of Software Engineering & Applications* **3**, (**1**): 117.

[9] Lei, Howard, Farnaz Ganjeizadeh, Pradeep Kumar Jayachandran, and Pinar Ozcan. (2017) "A statistical analysis of the effects of Scrum and Kanban on software development projects." *Robotics and Computer-Integrated Manufacturing* **43**: 59-67.

[10] M. Cohn. (2009) "Team Structure," *in Succeeding with Agile: Software Development Using Scrum, Addison-Wesley Professional* 177-199.

[11] Mundra, Ashish, Sanjay Misra, and Chitra A. Dhawale. (2013) "Practical scrum-scrum team: Way to produce successful and quality software."*13th International Conference on Computational Science and Its Applications, IEEE* 119-123.

[12] Overhage, Sven, Sebastian Schlauderer, Dominik Birkmeier, and Jonas Miller. (2011) "What makes IT personnel adopt scrum? A framework of drivers and inhibitors to developer acceptance." *44th Hawaii International Conference on System Sciences, IEEE.*1-10.

[13] Radu L. (2019) “Effort Prediction in Agile Software Development with Bayesian Networks,”*14th International Conference on Software.*

[14] Rising, Linda, and Norman S. Janoff. (2000) "The Scrum software development process for small teams." IEEE software **17, (4)**: 26-32.

[15] Rubin, Kenneth S. (2012) “Essential Scrum: A practical guide to the most popular Agile process”. USA: Addison-Wesley.

[16] Sharp, Jason H., Sherry D. Ryan, and Victor R. Prybutok. (2014) "Global agile team design: an informing science perspective." *Informing Science: The International Journal of an Emerging Transdiscipline* **17**: 175-187.

[17] Sutherland, J., Viktorov, A., Blount, J. and Puntikov, N. (2007) “Distributed scrum: Agile project management with outsourced development teams”. *40th Annual Hawaii International Conference on System Sciences (HICSS'07) IEEE.* 274a-274a.

[18] Whitworth, Elizabeth. (2006) “Agile experience: communication and collaboration in agile software development teams*”.*

[19] Y. Lindsjørn, D. I. Sjøberg, T. Dingsøyr, G. R. Bergersen and T. Dybå. (2016) "Teamwork quality and project success in software development: A survey of agile development teams." *Journal of Systems and Software* **122, (12)**: 274-286.

[20] Zare, F., Zare, H.K. and Fallahnezhad, M.S. (2016). "Software effort estimation based on the optimal Bayesian belief network." *Applied Soft Computing* **49**: 968-980.

[21] Zia, Ahmed, Waleed Arshad, and Waqas Mahmood. (2018) "Preference in using agile development with larger team size." *International Journal of Advanced Computer Science and Applications* **9**, (**7**): 116-123.

[22] https://github.com/SEAnalytics/datasets/tree/master/agile%20sprints/IEEE%20TSE2017/dataset.

[23] www.leadingagile.com › 2018/08 › Transformation-Whitepaper-final agile transformation – Leading Agile.

[24] Apache projects, https://issues.apache.org/jira/

[25] JBoss projects, https://issues.jboss.org/

[26] Jira projects, https://jira.atlassian.com/

[27] MongoDB projects, https://jira.mongodb.org/

[28] Spring projects, https://jira.spring.io/