DOI: 10.15439/2020F214
ISSN 2300-5963 ACSIS, Vol. 21

Machine Learning models to predict Agile Methodology adoption

Ridewaan Hanslo
Council for Scientific and
Industrial Research, South Africa.
Department of Information
Systems, University of Cape Town,
South Africa.
Email: rhanslo@csir.co.za

Maureen Tanner
Department of Information
Systems, University of Cape Town,
South Africa.
Email: mc.tanner@uct.ac.za

***Abstract*—Agile software development methodologies are used in many industries of the global economy. The Scrum framework is the predominant Agile methodology used to develop, deliver, and maintain complex software products. While the success of software projects has significantly improved while using Agile methodologies in comparison to the Waterfall methodology, a large proportion of projects continue to be challenged or fails. The primary objective of this paper is to use machine learning to develop predictive models for Scrum adoption, identifying a preliminary model with the highest prediction accuracy. The machine learning models were implemented using multiple linear regression statistical techniques. In particular, a full feature set adoption model, a transformed logarithmic adoption model, and a transformed logarithmic with omitted features adoption model were evaluated for prediction accuracy. Future research could improve upon these findings by incorporating additional model evaluation and validation techniques.**

Index Terms—Adoption, Agile methodologies, Machine learning, Scrum.

I Introduction

THE Scrum framework is one of many Agile software development methodologies [1]-[3]. The purpose of the Scrum framework is to develop, deliver, and maintain complex software products. The Scrum Guide [4] defines Scrum as *"a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value."*

Scrum remains the predominant Agile software development methodology used for project management according to the 13th annual State of Agile survey. According to the survey [5], Scrum and Scrum variants (such as Scrumban and the Scrum/XP hybrid) account for 72% of the Agile methodologies used.

The rise in popularity of Agile approaches has grown to other industries within the global economy. Some of these industries we are referring to are Transportation, Education, Energy, Healthcare and Pharmaceuticals, and Financial Services [5]. With this growth in Agile popularity, this study posits that incorporating predictive and prescriptive analytics with Agile methodologies' context of use is a start at unpacking the complex relationships between factors related to Agile project outcome.

Machine Learning (ML) can be defined as the study of a *"real world"* phenomenon implementing the scientific principle to iteratively validate and refine a model or hypothesis [6]. From literature as recent as 2015, there was a mention for the need to incorporate Agile and data science methodologies to see frequently realized gains to software development and applications [7].

The purpose of this paper was to use ML techniques to predict adoption of the Scrum Agile methodology. This research takes the reader through the data science lifecycle of the defined problem, data collection, data preparation and exploration, feature extraction, prediction model development, testing and evaluation, followed by the discussion of the findings.

The remainder of this paper comprises of the following sections: Sect. 2 discusses the research problem; Sect. 3 provides literature on incorporating machine learning techniques with Agile software development; Sect. 4 presents the research methodology including the statistical analysis techniques. The results of the machine learning predictive modelling are presented in Sect. 5 and a discussion of the research findings are provided in Sect. 6. Section 7 concludes the paper and provides recommendations for future research.

II Research Problem

There is plenty of literature on the benefits and success of Agile software development methodologies over traditional methodologies such as the Waterfall method [5], [8], [9]. However, literature also note that even when organizations use Agile methodologies and practices for software development projects, less than half of these projects were deemed successful.

The Standish Group's modern criteria for determining project outcomes are known as the triple constraint [9]. These constraints are:

Ridewaan Hanslo wishes to thank the Council for Scientific and Industrial Research (CSIR) for their support allowing the author to continue his scientific inquiry in the Agile software development and Artificial Intelligence research areas.

1. OnBudget – The project remained within the planned budget.
2. OnTime – The project was resolved within a reasonable time estimation.
3. A satisfactory result – The project delivered user and customer satisfaction even though changes were made to the initial scope.

The project outcome definitions taking the triple constraint into account can, therefore, be summarized as follows;

1. Successful – A project that has met all three constraints, OnBudget, OnTime, and with a satisfactory result.
2. Challenged – A project that has accomplished two of the three constraints upon project completion, for example, the project was delivered on budget with a satisfactory result but did not keep to the planned time-of-delivery.
3. Failed – A project that was cancelled before it could be completed, or completed but was not used.

The Standish Group's 2018 CHAOS report stated that 42% of the surveyed Agile projects succeeded, while 50% were challenged, and 8% were reported as failures [10]. While 42% success is not an ideal rate, it is nonetheless an improvement from previous Standish Group CHAOS reports. For example, the 2015 report for 2011 to 2015 had Agile project success as 39%, challenged projects at 52% and failed projects at 9%. When combining Agile and Waterfall projects, the successful project outcomes drops to a low 29% with projects that experienced challenges at 52% and failed projects at 19% [9]. The recent CollabNet VersionOne [5] annual global survey also stated that *"95% of respondents reported at least some of their agile projects have been successful with 48% reporting that most or all of their agile projects were successful"*.

The authors are, therefore, aware of the low success rate of software development project outcomes regardless of the industry, methodology, and project size. We are optimistic that the future project outcome success will have an upward trajectory, however, we are also aware that the acceleration of autonomous and converged technologies can deepen the problem.

We, therefore, posit that ML algorithms can be used to contribute towards improving the success of project outcomes. As a start to solving this complex problem, this research paper focused on developing ML models to predict Agile methodology adoption. Before the outcomes of the project are predicted, we think that predicting the adoption of an Agile methodology during the early stages of the methodologies inclusion in software development projects could contribute significantly to the future understanding and outcomes of Agile projects. In other words, we believe that by understanding the problem earlier at the adoption phase could allow the project team to implement strategies that could pivot the trajectory of future project outcomes.

III. Fusing machine learning with Agile methodologies

From an engineering perspective, ML involves developing software that implements scientific principles. This complex process can be simplified into three steps. The first step is to formulate a hypothesis about a phenomenon, which also includes the model selection. Secondly, collect data to test the hypothesis and validate the model. Lastly, iteratively refine the hypothesis for continuous model increments [6].

Both the Agile software development methodology and ML incorporates an iterative approach to providing solutions to complex challenges. Indeed, past studies have successfully utilized ML within the context of Agile software development. Kahles and others [11] applied ML to automate the root cause analysis in Agile software testing environments. The study was able to produce an ML model that could achieve a prediction accuracy of 88.9% by using artificial neural networks to either classify or pre-process the data for clustering, using manually labelled data.

Another research area within Agile software development where ML models are often used is in effort estimation. Software development effort estimation is the process of estimating the effort required by the software development team within the Agile environment to develop and maintain software [12]. The studies by [12], and [13] used ML algorithms for effort prediction. Satapathy and Rath [13] used ML algorithms such as Random Forest (RF), decision tree (DT), and stochastic gradient boosting (SGB) to improve upon the manual and tedious story pointing approach of effort estimation. The results indicate that RF, DT and SGB improved upon the story point approach, however, SGB outperforms the other two ML algorithms.

Moharreri and others [12] also developed an automated estimation methodology called *"Auto-Estimate"*. The study's ML model construction used supervised learning algorithms. The model was used to improve upon the manual Agile Planning Poker (PP) for effort estimation. PP involves all key stakeholders of the Agile planning team to estimate the effort required to complete a task, which usually makes use of playing cards with estimates using the Fibonacci series of numbers [12]. The results of the study indicate that the J48 Decision Tree (J48) and the Logistic Model Tree (LMT) ML algorithms outperformed PP. The results also suggest combining PP with J48 or LMT yields lower aggregate costs which could in future augment human effort estimation.

Other studies that combined Agile methodologies and ML include a study by [14], which successfully incorporated Agile practices in big data analytics, and the study by [15] which built a model using the J48 ML technique to predict software code defects during automated testing with 85% accuracy, drastically lowering the time needed to detect these potential problems. In addition, Schleier-Smith [7] incorporated Agile practices into data science real-time recommendation system development for benefits such as faster development cycles, quick feedback mechanisms and improved teamwork.

In summary, there is sufficient literature to be found on ML being used with Agile methodologies. The authors found more than 100 search results of ML being used with Agile methodologies on the Scopus and Web of Science citation databases alone.

IV. Methodology

Before the authors could develop and evaluate the ML predictive models a few preprocessing steps had to be undertaken. The first preprocessing step was to extract and synthesize Scrum and Agile adoption challenges within the literature. This was published in a paper entitled *"Scrum adoption challenges detection model: SACDM"* [16] in which a conceptual model was developed to test and evaluate challenges to Scrum adoption. A narrative review was conducted on the existing Agile and Scrum adoption challenges experienced globally and by practitioners in South Africa (SA). The synthesized challenges were used as the independent variables of the model. The first iteration of the Conceptual Framework (CF) known as SACDM generated 19 independent factors that are used to evaluate Scrum adoption as the dependent factor. Some of the independent factors included organizational structure, organizational culture, teamwork, experience, communication, collaboration, complexity, compatibility, and the relative advantage of the Scrum framework. This CF is a custom model adapted from the Diffusion of Innovation (DOI) theory and a study of the adoption of new technology by [17]. The descriptions of each of the independent and dependent variables can be obtained from the *"Scrum adoption challenges detection model: SACDM"* open-access paper [16].

To be able to identify the factors that contribute significantly to the adoption of the Scrum framework, there was a need for testing and evaluation of the CF. This was presented in another paper entitled *"Factors that contribute significantly to Scrum adoption"* [18] which described the process behind the three iterations of the CF that lead to the factors of significance. During the second iteration of the CF, SACDM was renamed as Scrum Adoption Challenges Conceptual Framework (SACCF). The online survey questionnaire serving as a Likert-type scale gathered response data from 78 questionnaire items. The Likert-type scale was used to record the perceived outcomes of Scrum adoption within the organisation, team and individually. The questionnaire design used in the previous paper is accessible online (https://bit.ly/scrumchallengessurvey). The sample consisted of Scrum practitioners working within South African organizations. The research design took the form of a narrative review and survey questionnaire. For the research analysis a set of 207 valid responses to this survey was used to perform Exploratory Factor Analysis (EFA) and Cronbach's alpha analysis, which confirmed the validity and reliability of the questionnaire as the measuring instrument. EFA further revealed that the factors can be reduced from 19 to 14 independent variables. Fig. 1 depicts the 14 factors with Scrum adoption as the dependent variable. The results from the correlational and multiple linear regression (MLR) statistics were used to identify factors that have a significant linear relationship with Scrum adoption. Factors revealed as significant were the management of the Scrum sprint, and the complexity and relative advantage of the Scrum framework. The details of the analysis results and findings can be obtained from the *"Factors that contribute significantly to Scrum adoption"* open-access paper [18].

Using quantitative analysis on Scrum adoption the authors were able to test nineteen research hypotheses in a chapter entitled *"Quantitative Analysis of the Scrum Framework"* [19]. Four hypotheses were shown to be statistically significant to Scrum. These hypotheses are the following;

1. Sprint Management: There is a significant linear (positive correlation) relationship between Sprint Management and Scrum adoption.
2. Change Resistance: There is a significant linear (negative correlation) relationship between Change Resistance and Scrum adoption.
3. Relative Advantage: There is a significant linear (positive correlation) relationship between Relative Advantage and Scrum adoption.
4. Complexity: There is a significant linear (negative correlation) relationship between Complexity and Scrum adoption.

The three papers just described allowing us to firstly, build a conceptual model and test the reliability and validity of the model as a CF. Secondly, the authors further found significant factors that contribute to Scrum adoption. These factors were quantitatively analyzed using correlation coefficients and MLR. Thirdly, thereafter, we could test the research hypotheses. To contribute further to the research field the authors want to incorporate predictive analytics on projects using Agile methodologies. This research paper, therefore, looks at developing the capability for teams and organizations to predict Scrum adoption using predictive analytics.

The factors discussed above form part of the feature engineering process, which is a pre-requisite to the ML model building. To build and test the ML models the sample data had to be split between the training set and test set. For both the training set and test set, Scrum adoption (dependent variable) and the features (independent variables) are added as arguments to the model. The following code sample adds the features and Scrum adoption to the *train_test_split* function of the scikit-learn machine learning library for Python (1).

X_train, X_test, y_train, y_test = train_test_split(features, adoption, test_size=0.3, random_state=4) (1)

In the code sample, the random state was set for testing and replicability (*random_state=4*). The dataset was split into a 69.57% training set and 30.43% test set. Before training the models it was important that the data was normalized and that all assumptions had been met. These assumptions are the assumption of normality of residuals, the assumption of no autocorrelation of residuals, the assumptions of linearity and homoscedasticity, and the assumption of no multicollinearity.

The Bayesian Information Criterion (BIC) is a model selection criterion for a finite list of models [20]. Weakliem [21] critiques BIC for excessively favouring simple models in practice, however, we used it because BIC is a widely used and popular criterion for model selection in linear regression. The lower the BIC value the better the model. The BIC equation can be defined as (2);

$$BIC = \ln(n)k - 2\ln(\hat{L}). \tag{2}$$

1. k=the data points.
2. n=the number of parameters estimated by the model.
3. $\hat{L}$=the maximum value of the likelihood function of the model.

For this paper, the authors used three models to test the prediction accuracy; the transformed logarithmic (log) adoption model, full feature set adoption model and the transformed log with omitted features adoption model. Each of these three models are using the MLR ML statistical analysis technique using the 14 explanatory variables to predict Scrum adoption as mentioned earlier.

The full feature set model includes the fourteen features (over-engineering, relative advantage, recognition, experience, teamwork, specialization, escalation of commitment, compatibility, resource management, customer collaboration, complexity, training, sprint management, organizational behaviour) to predict Scrum adoption. The transformed log model normalized the skewed data and includes the full feature set to predict Scrum adoption. The transformed log with omitted features model also normalized the skewed data, however, three of the fourteen features (experience, recognition, and compatibility) have been excluded from the feature set to predict Scrum adoption. The BIC value is -0.88 for the log adoption model, and -15.72 for the log with omitted features adoption model.

V. Results

To remind the reader, the prediction of Scrum adoption referred to as adoption going forward, is the focus of this paper. Fig. 1 displays the correlations of the feature set. The stronger the correlation the darker the displayed colour. The negatively phrased questions of features sprint management, teamwork, and over-engineering were recoded (identified by the r prefix).

Some of the relationships between the features and their significance are discussed below.

1. A positive and significant relationship between Relative Advantage and Adoption (*r=0.66, p<0.001*). The correlation was moderate to strong in strength.
2. A positive and significant relationship between Recognition and Organizational Behaviour (*r=0.66, p<0.001*). The correlation was moderate to strong in strength.
3. A positive and significant relationship between Relative Advantage and Compatibility (*r=0.64, p<0.001*). The correlation was moderate in strength.
4. A positive and significant relationship between Customer Collaboration and Training (*r=0.51, p<0.001*). The correlation was moderate in strength.
5. A positive and significant relationship between Resource Management and Organizational Behaviour (*r=0.64, p<0.001*). The correlation was moderate in strength.
6. A positive and significant relationship between Teamwork and Sprint Management (*r=0.71, p<0.001*). The correlation was strong in strength.
7. A positive and significant relationship between Complexity and Relative Advantage (*r=0.51, p<0.001*). The correlation was moderate in strength.
8. A positive and significant relationship between Resource Management and Training (*r=0.39, p<0.001*). The correlation was weak to moderate in strength.
9. A positive and significant relationship between Resource Management and Recognition (*r=0.48, p<0.001*). The correlation was moderate in strength.
10. A positive and significant relationship between Compatibility and Adoption (*r=0.50, p<0.001*). The correlation was moderate in strength.

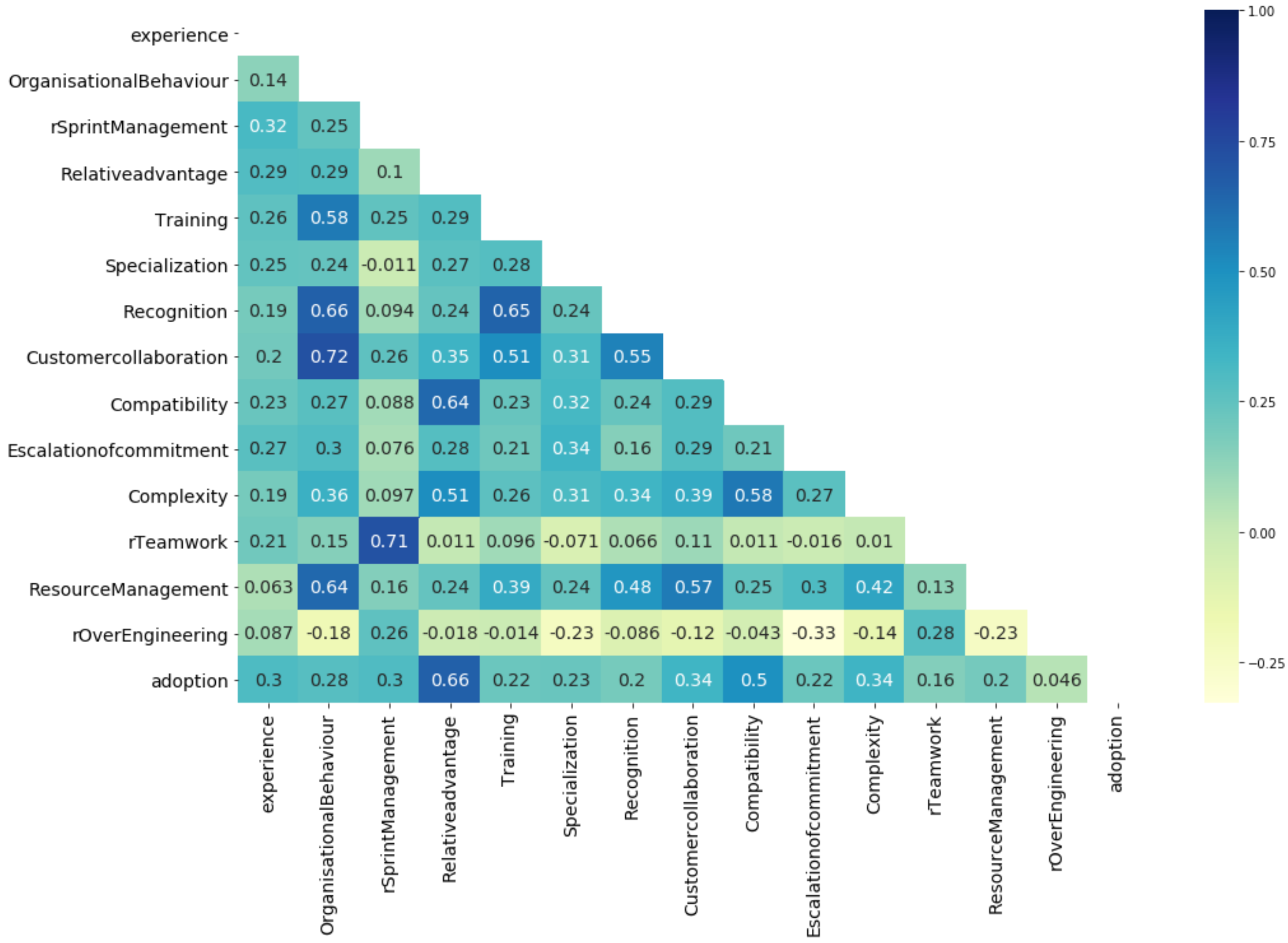

Fig. 1 The Feature correlation heat map. Displays the relationship between the features and Scrum adoption.

The first model is the full feature set model which has an actual and predicted adoption correlation of 0.75. Fig. 2 displays the actual and predicted adoption and Fig. 3 depicts the residuals. The 95% prediction interval is 4.83 and 1.98 for the upper and lower bound in the full feature set adoption model, respectively.

For the second model, Fig. 4 displays the actual and predicted adoption correlation for the log transformation, while Fig. 5 displays the residual and predicted values. The actual and predicted log adoption correlation is 0.73. The 95% prediction interval is 3.80 and 3.01 for the upper and lower bound in the log adoption model, respectively.

The third model is transformed using log adoption and simplified by dropping three features, namely, experience, recognition, and compatibility, as mentioned earlier. Fig. 6 depicts the transformed and simplified model while the residuals of this model are displayed in Fig. 7. The actual and predicted log adoption with omitted features correlation is 0.73. The 95% prediction interval is 3.79 and 3.01 for the upper and lower bound, respectively, in the log adoption with omitted features model.

Table I displays the R-squared (R^2), and the Mean Squared Error (MSE) for each of the three ML predictive models. The R^2 is a statistical measure of the variance of the predicted values divided by the variance of the data [22]. A 0% R^2 value indicates that the models explain none of the variability of the data, in other words, it is worse than predicting the mean. A 100% R^2 value indicates that the model explains all the variability of the data. The R^2 equation divides the sum of the squares due to regression by the total sum of squares (3).

$$R^2 = 1 - \frac{SS_{regression}}{SS_{total}} \tag{3}$$

The MSE criterion calculates how close the regression line is to the data points by taking into account the predicted value of the observation and eliminates the arbitrariness associated with the residual sum of the squares [23]. Put in another way, the MSE equation measures the average squared error of our predictions where y_i is the actual output and $\hat{y}_i$ is the model's prediction (4). The lower the MSE value the lower the variance of error.

$$MSE = \frac{1}{n}\sum_{i-1}^{N}(y_i - \hat{y}_i)^2 \tag{4}$$

TABLE I.
ML PREDICTIVE MODELS

Adoption Model	R-squared	R-squared %	MSE
Transformed log model	0.527	52.7	0.039
Full feature set model	0.564	56.4	0.507
Transformed log with omitted features model	0.527	52.7	0.038

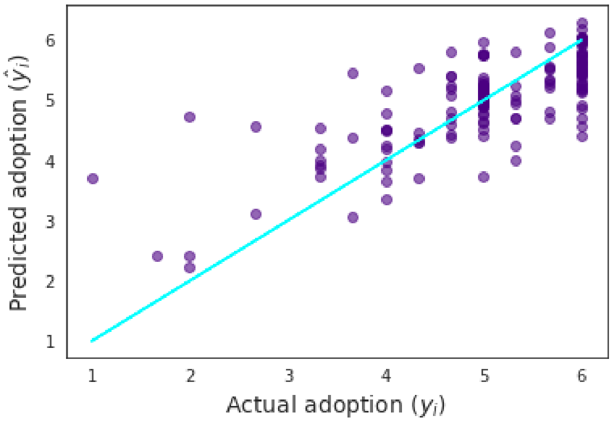

Fig. 2 The normal probability plot for the untransformed full feature set model. The assumption of normality of residuals was met because the actual and predicted adoption residuals were approximately linear.

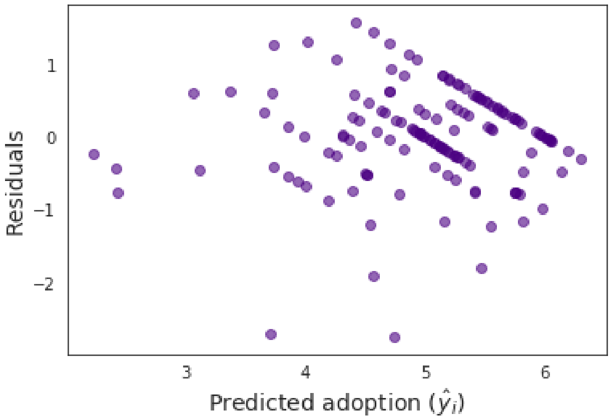

Fig. 3 The Scatterplot for the untransformed full feature set model. The assumptions of linearity and homoscedasticity were met because the residual and predicted values did not curve or funnel out.

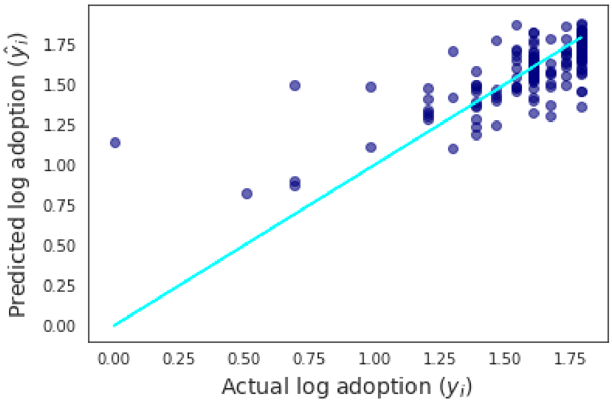

Fig. 4 The normal probability plot for the log adoption model. The assumption of normality of residuals was met because the actual and predicted adoption residuals were approximately linear.

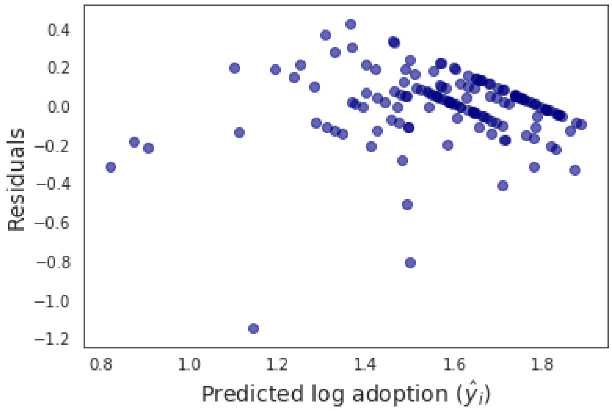

Fig. 5 The Scatterplot for the log adoption model. The assumptions of linearity and homoscedasticity were met because the residual and predicted values did not curve or funnel out.

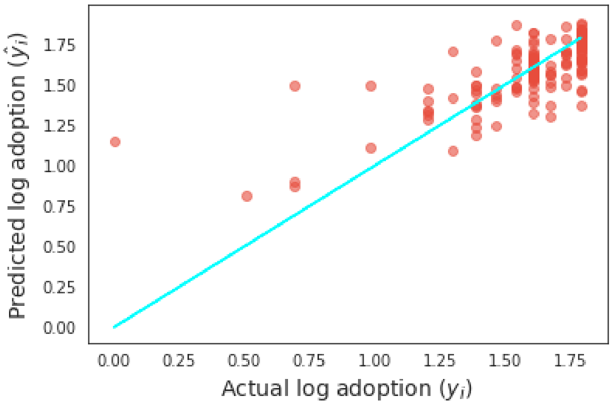

Fig. 6 The normal probability plot for the log adoption with omitted features model. The assumption of normality of residuals was met because the actual and predicted adoption residuals were approximately linear.

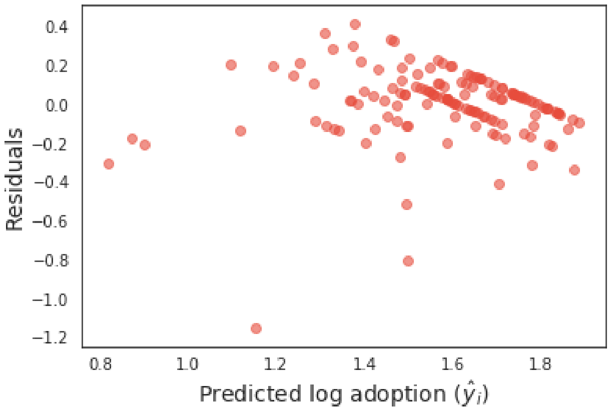

Fig. 7 The Scatterplot for the log adoption with omitted features model. The assumptions of linearity and homoscedasticity were met because the residual and predicted values did not curve or funnel out.

VI. Discussion of Findings

The preprocessing and feature engineering of the response data as described in the research methodology section allowed us to build and evaluate three machine learning (ML) models. The three models were not an exhaustive collection of predictive models as this approach was beyond the scope of this research paper. We wanted to investigate whether different models which include transformations and simplified feature sets can predict Scrum adoption with less variance and error, in other words, at what prediction accuracy.

The three models evaluated in this study was the log adoption model, full feature set adoption model, and the log with the omitted features adoption model. As mentioned in the results, the R-squared (R^2) value measures how close the data are to the regression line, and the Mean Square Error (MSE) measures the average of the square of the errors.

The full feature set adoption model has a moderate variance value of 0.564, explaining more than half of the model instances. The closer the R^2 is to 1 usually the greater the prediction accuracy. This model also has an MSE value of 0.507, indicating a high error rate as the error value is closer to one.

The log adoption model is a transformation of the full feature set model. This model has an R^2 value of 0.527 with a 0.039 MSE value. It is immediately evident that the log model is a better model for adoption prediction accuracy because of the MSE being closer to zero while the R^2 value is greater than 0.5 and less than 0.6, similar to the full feature set model.

The third model is the log with the omitted features adoption model. This model simplified the feature set by removing three of the fourteen features. The three features are experience (*p-value=0.929*), recognition (*p-value=0.969*), and compatibility (*p-value=0.820*) due to their high p-values. The higher the p-values, the less significant of a factor it is to adoption. With the three features removed the R^2 value is 0.527 and the MSE is 0.038.

The transformed log with omitted features model is, therefore, the best-fit prediction model even though it gives a marginally lower error level than the log model. We are fairly confident that we can improve upon the prediction accuracy with a greater randomized sample. Further, we can improve upon the best-fit model by developing a model with lower variance and MSE.

VII. Conclusion

This research paper reports on the development of ML models to predict the accuracy of Scrum adoption based on a feature set derived from a survey questionnaire's response data. The sample size of 207 response data was used to train and test the prediction models. Data cleaning and preprocessing was required before the models could be trained and tested. We trained each of the three models with approximately 70% of the dataset while 30% was used to test the models. The three prediction models was a full feature set adoption model, transformed logarithmic (log) adoption model, and a transformed log with omitted features adoption model. The adoption model with the highest prediction accuracy was the transformed log with omitted features model with an *R^2=0.527* and *MSE=0.038*. The full feature set model was the least accurate when looking at the combination of *R^2=0.564* and *MSE=0.507*. Implications of these findings, while still preliminary, allows researchers and practitioners to gain a better understanding into which features are potentially significant to predicting Scrum adoption. Researchers could compare our findings against their own and modify their modelling techniques.

Limitations of this research are threefold. Firstly, the training and test split used in this research paper for the dataset has been reported previously as displaying biases [24]. Secondly, the model evaluation measure of R^2 being used for goodness-of-fit of the models are one of many metrics used for prediction model evaluation. Thirdly, additional model validation techniques such as bootstrap sampling has not been used in this preliminary research.

Additional research, therefore, could implement the bootstrap aggregating technique to improve the stability and accuracy of the ML algorithms. Metrics such as max error, mean absolute error, the median squared error could be used to further evaluate prediction accuracy. Using a larger randomized sample would improve the predictive accuracy of the models used within this research paper. Further research could develop a logistic regression model with a larger dataset to predict Agile project outcomes, modifying the conceptual framework and methodology as required.

References

[1] A. Przybylek, D. Kotecka, “Making agile retrospectives more awesome,” *In Federated Conference on Computer Science and*

Information Systems (FedCSIS), Prague, Czech Republic, 2017. https://doi.org/10.15439/2017F423

[2] A. Przybylek, M. Zakrzewski, "Adopting Collaborative Games into Agile Requirements Engineering," *In 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE'18)*, Funchal, Madeira, Portugal, 2018. https://doi.org/10.5220/0006681900540064

[3] A. Przybyłek, M. Olszewski, "Adopting collaborative games into Open Kanban," *In Federated Conference on Computer Science and Information Systems (FedCSIS)*, Gdansk, Poland, 2016. https://doi.org/10.15439/2016F509

[4] K. Schwaber and J. Sutherland, "The Scrum Guide," *scrum.org*, https://www.scrum.org/index.php/resources/scrum-guide. 2020.

[5] CollabNet VersionOne, "13th Annual State of Agile Report," *collab.net*, https://www.stateofagile.com. 2020.

[6] A. Jung, "Machine Learning: Basic Principles," *arXiv: 1805.05052v11[cs.LG]*, https://arxiv.org/pdf/1805.05052.pdf. 2019.

[7] J. Schleier-Smith, "An architecture for agile machine learning in real-time applications," *In Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2059-2068, ACM, 2015.

[8] R. Hoda and L.K. Murugesan, "Multi-level agile project management challenges: A self-organizing team perspective," *The Journal of Systems & Software*, 117, 245-257. 2016. https://doi.org/10.1016/j.jss.2016.02.049

[9] The Standish Group, "CHAOS Report 2015," *The Standish Group*, https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf. 2020.

[10] Vitality Chicago, "Agile Project Success Rates are 2X Higher than Traditional Projects," *VitalityChicago* https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/. 2020.

[11] J. Kahles, J. Torronen, T. Huuhtanen and A. Jung, "Automating Root Cause Analysis via Machine Learning in Agile Software Testing Environments," *In Proceedings - 2019 IEEE 12th International Conference on Software Testing, Verification and Validation*, ICST 2019, pp. 379-390. [8730163] Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICST.2019.00047

[12] K. Moharreri, A.V. Sapre, J. Ramanathan and R. Ramnath, "Cost-effective supervised learning models for software effort estimation in agile environments," *In 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC),* vol. 2, pp. 135-140. IEEE. 2016.

[13] S.M. Satapathy and S.K. Rath, "Empirical assessment of machine learning models for agile software development effort estimation using story points," *Innovations in Systems and Software Engineering*, 13(2-3), pp.191-200. 2017.

[14] H.M. Chen, R. Kazman and S. Haziyev, "Agile big data analytics for web-based systems: An architecture-centric approach," *IEEE Transactions on Big Data*, 2(3), pp.234-248. 2016.

[15] L. Butgereit, 2019, "Using Machine Learning to Prioritize Automated Testing in an Agile Environment," *In 2019 Conference on Information Communications Technology and Society (ICTAS)*, pp. 1-6. IEEE. 2019.

[16] R. Hanslo and E. Mnkandla, "Scrum Adoption Challenges Detection Model: SACDM," *In Federated Conference on Computer Science and Information Systems (FedCSIS)*, Poznan, Poland: IEEE: 949–957. 2018.

[17] F. Sultan and L. Chan, "The adoption of new technology: the case of object-oriented computing in software companies," *IEEE transactions on Engineering Management*, 47(1): 106–126. 2000.

[18] R. Hanslo, E. Mnkandla and A. Vahed, "Factors that contribute significantly to Scrum adoption," *In Federated Conference on Computer Science and Information Systems (FedCSIS)*, Leipzig, Germany: IEEE: 821–829. 2019.

[19] R. Hanslo, A. Vahed and E. Mnkandla, "Quantitative Analysis of the Scrum Framework," *In: Przybyłek A., Morales-Trujillo M. (eds) Advances in Agile and User-Centred Software Engineering. LASD 2019, MIDI 2019*, Lecture Notes in Business Information Processing, vol 376. Springer, Cham. 2020. https://doi.org/10.1007/978-3-030-37534-8_5

[20] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," *In Proc. DARPA broadcast news transcription and understanding workshop*, vol. 8, pp. 127-132. Feb. 1998.

[21] D.L. Weakliem, "A critique of the Bayesian information criterion for model selection," *Sociological Methods & Research*, 27(3), pp.359-397. 1999.

[22] A. Gelman, B. Goodrich, J. Gabry and A. Vehtari, "R-squared for Bayesian regression models," *The American Statistician*, 73(3), pp.307-309. 2019.

[23] D.M. Allen, "Mean square error of prediction as a criterion for selecting variables," *Technometrics*, 13(3), pp.469-475. 1971.

[24] C. Tantithamthavorn, S. McIntosh, A. E. Hassan and K. Matsumoto, "An Empirical Comparison of Model Validation Techniques for Defect Prediction Models," *In IEEE Transactions on Software Engineering*, vol. 43, no. 1, pp. 1-18, Jan. 2017. https://doi.org/10.1109/TSE.2016.2584050