PAPER • **OPEN ACCESS**

Effect of Dimensionality Reduction on Prediction Accuracy of Effort of Agile Projects Using Principal Component Analysis

To cite this article: Ms. Manju Vyas and Dr. Naveen Hemrajani 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1099** 012008

Effect of Dimensionality Reduction on Prediction Accuracy of Effort of Agile Projects Using Principal Component Analysis

Ms. Manju Vyas[1] and Dr. Naveen Hemrajani[2]

[1]Research Scholar, JECRC University, Jaipur, India
[2]Professor, JECRC University, Jaipur, India[2]

E-mail: vyas.manju@gmail.com

Abstract. Agile framework for software development has received a lot of recognition in software industry in previous years as it focuses on rapid incremental delivery, lower risk and customer satisfaction. At early stages of development, the effort must be predicted so that the project is completed successfully within the time and cost deadlines. In recent years, various researchers have done study in this area and it is observed that the prediction of effort faces a problem of large dimension of features. Hence the prediction accuracy may be increased by reducing the dimensions of the features. In this paper, PCA has been used for reduction of feature dimensions for effort estimation. PCA identifies the key attributes by reducing the dimensions of the attribute which are those having highest correlation with the effort. The methodology shows the effect of PCA on the original dataset and the results are observed by applying various machine learning techniques pre and post PCA. The comparison metrics used are Mean Magnitude relative Error (MMRE), Root Mean Square Error (RMSE), and Prediction Accuracy (PRED (25)). The decreased values of errors and increased value of accuracy shows the better model accuracy when PCA is applied on the dataset. All the computations and implementations in this paper are done using Python on Scikit-learn library.

1. Introduction

1.1 Estimation

The estimation of effort plays a significant role in software development management and planning. The estimations are expected to be reasonable in terms of resources, cost and schedule. A limited time frame is planned for the estimations and they must be updated regularly as well with the progress of project development. The effort basically predicts the resources in man-hours [1]. Agile is used as a software development framework in recent years in software industry as it is found to have many strengths over other traditional development framework like waterfall etc. Agile is an iterative approach which focuses on customer satisfaction and incremental delivery of software so that at any moment requirements may be accommodated during the development cycle thus making it dynamic and low risk prone [2]. The effort is calculated in agile mainly depending on the software size which is measured by no. of story points which is a measure of user stories and team velocity. We have considered a public dataset of projects having various features required for calculation of effort specifically for software projects developed using agile framework [3]. The prediction normally suffers from an issue of large dimension of features [4][5][6]. In this paper we suggest the application of PCA for feature extraction for improving the accuracy of prediction and reducing the error. The methodology discusses the detailed steps required for PCA like finding the covariance matrix and

calculating the Eigen values. The experimental evaluation is done using various machine learning techniques and observing the accuracy before and after applying PCA. The results show that the error of the prediction reduces after the application of PCA as the correlated data is now removed which further maximises the non-linearity. The paper is arranged into five parts. First Section discusses the context and introduction of the area. Second Section describes the review of existing literature available. The third Section discusses the various background techniques used including Regression, SVR and Principal Component Analysis. This section throws a light on the techniques, the dataset and all other relevant details regarding the algorithm proposed. Section Four describes the practical implementation and results. The results are implemented using Python. At the end, Section Five reports the conclusion and scope of future work.

1.2 Motivation and Objective

The motivation and objective behind this work is to provide an insight on the usage of dimensionality reduction and its impact on the effort and cost estimation of the software projects. Since now-a-days most of the industries use agile as development framework and dimensionality reduction has proven better accuracy in prediction in other frameworks so this paper is written to test the effect of the PCA technique for reducing the dimensions on the accuracy of estimation.

2. Related Work

In his research Zia et al. [3] proposed a statistical technique based on SWOT analysis which takes into consideration various internal and external factors and their influence on effort and cost estimation. Various Factors like team structure, systematic requirements etc. are considered and are quantified. The paper also published a dataset of several agile projects containing various features and concludes that the accuracy of the proposed model in terms of Pred (25) comes out to be PRED Time (7.19) = 57.14 %, PRED Cost (5.76) = 61.90 % while the error is measured in metrics Mean Magnitude of Relative error and it comes out to be MMRE (Time) = 7.19 % & MMRE (Cost) = 5.76 %

Garg et al. in [4] proposed a model for cost estimation of agile projects. They applied PCA in the model for dimensionality reduction of the attributes and the maximum correlation attributes were identified. Then constraint solving approach was applied for suitability in agile manifesto. The model was compared with other published approaches and was concluded to have better accuracy in terms of low values of MMRE.

Tosun et al. in [5] mentioned that the analogy based estimation models treats all project features in equal scale which is a shortcoming as various factors may have different impacts on the estimation which depends on their relevance with the metric which is being estimated. They proposed two techniques for assigning weights to the features for estimating the cost. One technique used Principal Component Analysis for extracting the features and tests the proposed work using public datasets. The paper concludes that feature extraction increases the accuracy of prediction. Also it concludes that as the dataset grows vertically the results of the proposed technique increases.

In [6] Weng et al. also observed the analogy based models used for software effort estimation and found that the similarity measure used in analogy based projects is Euclidean distance which assigns uniform weights to all the features, resulting in inaccurate estimation. They proposed a PCA based approach for feature extraction and then used Pearson correlation coefficient to establish correlation between effort and extracted features. The experimental evaluation was carried out on three benchmark datasets and it was shown that the estimation accuracy of the suggested model is higher as compared to the previously published researches.

Another research paper from Asnawi et al. in [7] identified 15 factors using factor analysis technique on a data of 27 variables in context of agile which was collected by survey from software practitioners who adopted agile methodology. They identified the clusters and their correlation using eigenvalue rules and concluded the factors which impacts the agile methodology like organization culture, developer's involvement etc. Although the observation is that these factors does not have significant impact on cost, time and effort estimation.

In [8], Keung et al. proposed a method and named it Analogy-X in which the candidate features were examined for their influence & level of significance on a correlation coefficient between the Euclidean distance & distance vectors. The result was an optimal subset of features.

Drawing motivation from the above mentioned researches we investigated the effect of PCA on application of various machine learning techniques applied pre and post PCA for effort estimation of agile projects [9].

3. Background Techniques

3.1 Machine Learning Techniques

The proposed work has checked the impact of dimensionality reduction on three major techniques as discussed below i.e. Regression, Naive Bayes and Support Vector Regression.

3.1.1 Regression

Regression models are based on mathematical concepts and are built by gathering historical data of completed projects and then establishing relationship between various variables by the use of the regression equations. The prediction for a fresh project is made by substituting the parameters to the mathematical model. The effort is considered as independent / predictor variable. For application of regression models, missing values and outliers need to be handled along with the elimination of co linearity. The regression models applied here are linear regression, ridge regression & logistic regression.

3.1.2 Naive Bayes

The naive bayes approach basically uses the advantages of both expert based methods and regression based models. It uses probabilistic concepts to predict the dependent variable. Gaussian NB deals with continuous data and discretizes the values to get a new set of features. Likewise other NB techniques work on the various concepts of probability.

Support Vector Machine: SVR as described in [10] is a version of SVM which implements the inductive principle of minimization of structural risk. It uses a regularization parameter to resolve the problem of over-fitting and local minima. It also uses a kernel function which can be optimized thus having improved capability of generalization. [11] proposed an SVM called ε-SVR which uses ε loss function for prediction. ε-SVR basically defines a band which surrounds the output by defining the loss function. The concept is that those errors which falls inside the band i.e., the errors which are less than certain threshold ε>0 are not considered while those which lies out of the band are measured by the use of variables ζ and ζ*.

In case of linear regression using SVR, function g(x) is defined by g(x) = (w, x) + d, where w Ɛ χ and x Ɛ belongs to R. In case of non-linear regression g(x) = (w, ϕ(x)) + d, ϕ denotes a non linear function. The values of w and d are selected for optimisation of below problem [11].

$$minimize_{w,b,\xi,\xi^*} \quad \frac{1}{2}\langle w,w\rangle + C\sum_{i=1}^{l}(\xi_i + \xi_i^*),$$

$$subject\ to \quad \begin{cases} \left(\langle w, \Phi(x^i)\rangle + b\right) - y_i \le \varepsilon + \xi_i, \\ y_i - \left(\langle w, \Phi(x^i)\rangle + b\right) \le \varepsilon + \xi_i, \\ \xi_i, \xi_i^* \ge 0. \end{cases} \qquad (1)$$

The role of SVR is used to optimize the function which is defined using the slack variables ζ and ζ* and constant C.

3.2 Principal Component Analysis

PCA is used for dimensionality reduction. It is a linear technique and is unsupervised. The technique uses orthogonal transformation for mapping the original data space into reduced space in terms of dimensions. The process used for reduction is given below:

3.2.1 Feature Extraction: PCA is a statistical technique which extracts some major features from actual components with the use of linear transformation. The dominant features are then deducted by filtering these features.

Step 1: Feature set data – The data consists of the X-axis values on the basis of which the patterns are identified. Since the data values are of high variance, it has to be scaled to a limited range first.

Step 2: Standard Scaling values - Standard Scaling remove the mean and scaling to unit variance. Since the data values constituted of large variance, the estimator fails in learning correctly from the values. Scaling brings the values in a similar range which thus reduces the variance.

Step 3: PCA transformed values- After the application of PCA, the size of the data comes out reduces in terms of factors. This reduction in the number of columns occurs due to the fact that PCA remove the correlated data thus maximizing the non-linearity in the data. Since some of the data had linear relations, so it was removed by PCA, thus leaving only the un-correlated data. The flowchart of the PCA is shown in Figure 1.

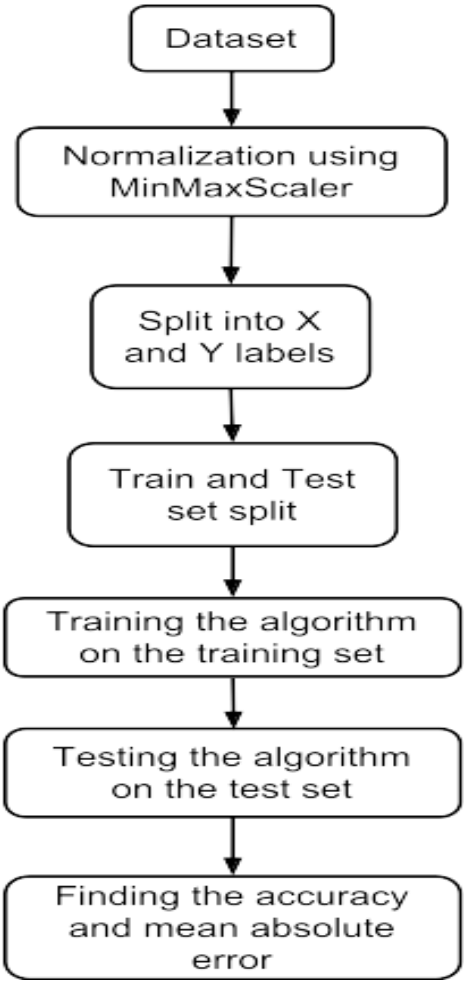

Figure 1. Generalized Flow of Proposed Technique

3.2.2 Correlation Calculation of the feature set: For the application of PCA on effort estimation, we have to check the relationship between various features to find out the most important feature. A covariance matrix is constructed. After the covariance matrix the Eigen vectors and Eigen values are calculated which indicates how the principal components are identified, which are basically linear combinations of current features. The Eigen vectors are the weights, which represent the weight age of each feature. [12][13][14][15] shows that following steps are used for covariance calculation

i) The mean of all values is calculated.
ii) The covariance matrix is then created by subtracting the elements from mean value.
iii) The covariance matrix is used to determine the Eigen-values
iv) The eigenvectors are sorted in decreasing order the required number of vectors based on the required number of dimensions are extracted.
v) Finding the dimensions based on first k Eigen vectors, and transforming the n dimensional points to k dimensions.

Table 1 shows the correlation matrix having covariance values of various features and Table 2 shows the eigen vectors and Figure 2 shows the covariance plotting.

Table 1. Covariance Matrix

Effort	Vi	D=no. of ideal days	V=velocity of the team	Sprint Size	Work days	Team Salary	Act Time	Covariance
Effort	6846.414	4.876429	2.176514	12.077143	0	-1.**914286**	2813357	1979.028571
Vi	4.876429	0.106143	0.009436	0.109714	0	0.018571	6385.714	-0.452143
D	2.176514	0.009436	0.005045	0.02762	0	0.003969	2088.024	0.153729
V	12.07714	0.109714	0.02762	0.191905	0	0.02619	13345.24	0.169286
Sprint Size	0	0	0	0	0	0	0	0
Work days	-1.91428 6	0.018571	0.003969	0.02619	0	0.047619	-690.4762	-1.778571
Team Salary	2813357	6385.714286	2088.02381	13345.2381	0	-690.47619	2154762000	641714.2857
Act Time	1979.029	-0.452143	0.153729	0.169286	0	-1.778571	641714.3	685.257143

Table 2. Eigen Vectors

Effort	**Vi**	**D**	**V**	**Sprint Size**	**Work days**	**Team Salary**	**Act Time**
Effort	0.001305647	-0.938402	-0.345427	-0.007258	-0.003866	0.003409	-0.00007409916
Vi	0.000002963531	0.00113	-0.013718	0.750099	0.107065	0.636479	-0.1435032
D	0.000000969026 4	0.000189	-0.003373	0.043125	-0.037789	-0.261757	-0.9634231
V	0.000006193361	0.001762	-0.023316	0.657515	-0.193601	-0.692069	0.2251387

Sprint Size	0	0	0	0	0	0	0
Work days	-0.00000032 04425	0.000416	-0.015228	0.049574	0.974419	-0.217448	0.02313212
Team Salary	0.9999991	0.001328	0.000172	-0.000004	0.000003	0.000001	-0.00000004515743
Act Time	0.00029781 26	-0.345535	0.937926	0.025603	0.011014	-0.011112	0.0003821539

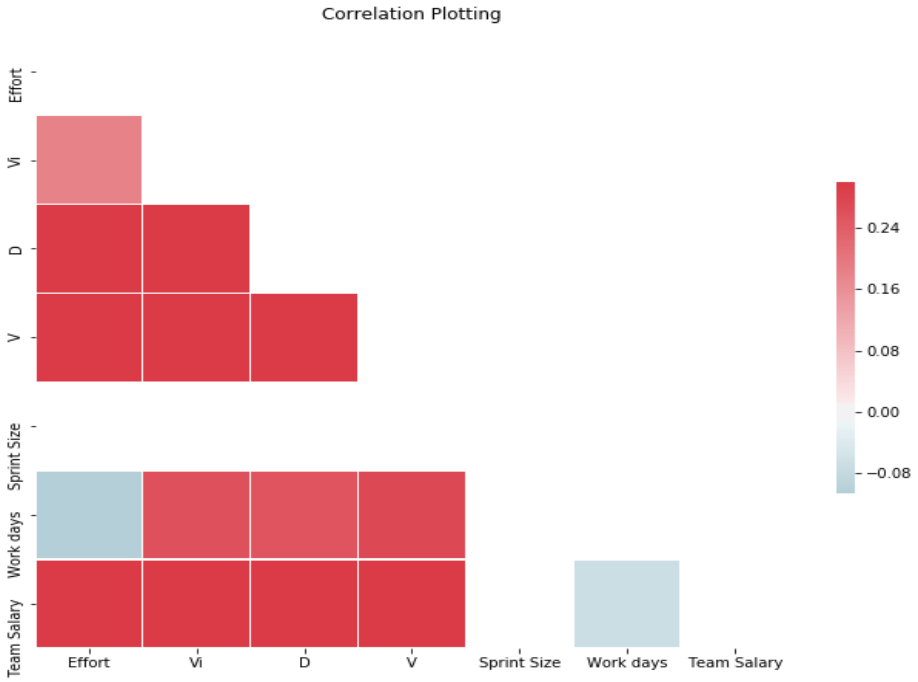

Figure 2. Correlation Plotting

4. Practical Implementation & Results

We used 12 predictive algorithms selected from the existing work on effort estimation. The selection is made on the basis of representative work from reputed publications in the area of software effort estimation of projects developed in agile methodology. All the algorithms were implemented using Python on Scikit-learn library. The database is split into two training and testing sets (80 % and 20 %) and is input to various classifiers. Then the accuracy, is tested using various evaluation metrics like MMRE, RMSE, & Pred (25) shown in eq. 2-5. The table 3 shows the various errors like MMRE,RMSE and prediction accuracy and Figures 3, 4 & 5 shows the plotting of MMRE, RMSE and Pred() pre and post PCA.

$$\mathrm{MRE} = \frac{|\mathrm{AE} - \mathrm{PE}|}{\mathrm{AE}} \qquad (2)$$

$$\mathrm{MMRE} = \frac{1}{\mathrm{m}}\sum_{\mathrm{k}=1}^{\mathrm{m}} \mathrm{MRE_k} \qquad (3)$$

Where MRE = Mean Relative Error, MMRE = Mean Magnitude of Relative Error, AE = Actual Effort and PE = Predicted Effort

$$Pred(l) = \frac{p}{m} \qquad (4)$$

Table 3. Comparison of Errors & Accuracy Pre & Post PCA

Algorithm	Result	RMSE	MMRE	PRED(25)
Linear_Regression	Before PCA	8.54	0.1	100
	After PCA	8.21	0.09	100
Gaussian_NB	Before PCA	17.57	0.21	71.42
	After PCA	13.06	0.18	85.71
Bernoulli_NB	Before PCA	10.41	0.2	71.42
	After PCA	9.73	0.16	85.71
Multinomial_NB	Before PCA	12.43	0.12	100
	After PCA	10.52	0.11	100
Logistic_Regression	Before PCA	14.67	0.18	71.42
	After PCA	11.31	0.15	85.71
Ridge_Regression	Before PCA	9.79	0.14	85.71
	After PCA	8.57	0.13	100
SVM	Before PCA	16.78	0.25	57.14
	After PCA	15.32	0.18	71.42
Ada_Boosting	Before PCA	15.06	0.29	57.14
	After PCA	11.73	0.19	71.42
Random_forest	Before PCA	15.22	0.23	71.42
	After PCA	14.28	0.2	71.42
Extra_Trees	Before PCA	10.13	0.21	57.14
	After PCA	8.07	0.1	100
Bagging_Regressor	Before PCA	17.25	0.29	57.14
	After PCA	14.82	0.23	71.42
XGB_Regression	Before PCA	11.91	0.13	100
	After PCA	9.95	0.094	100

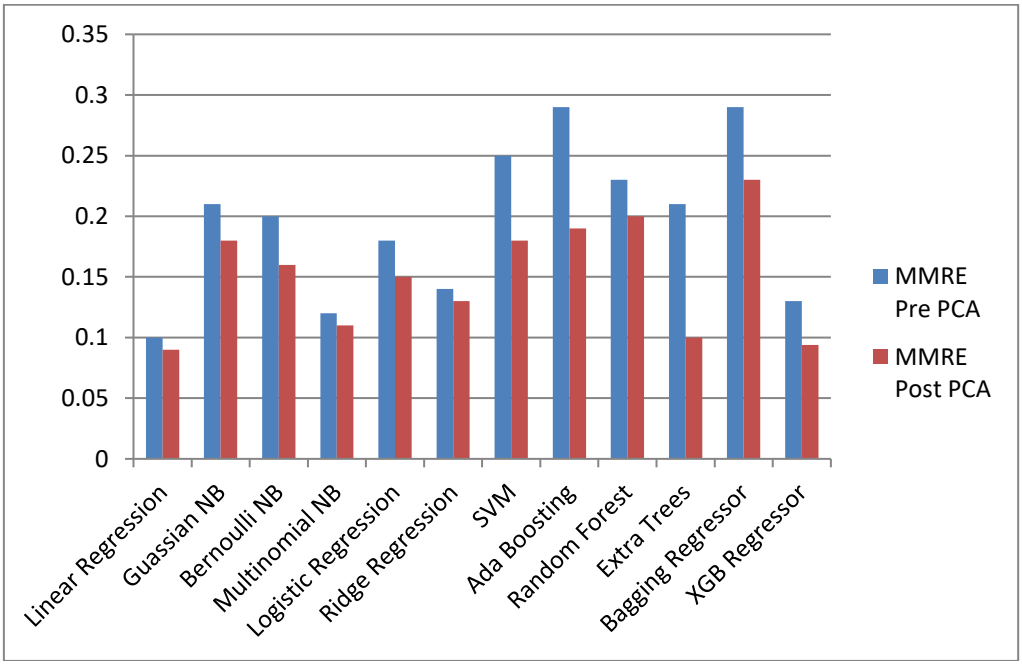

Figure 3. MMRE Pre and Post PCA

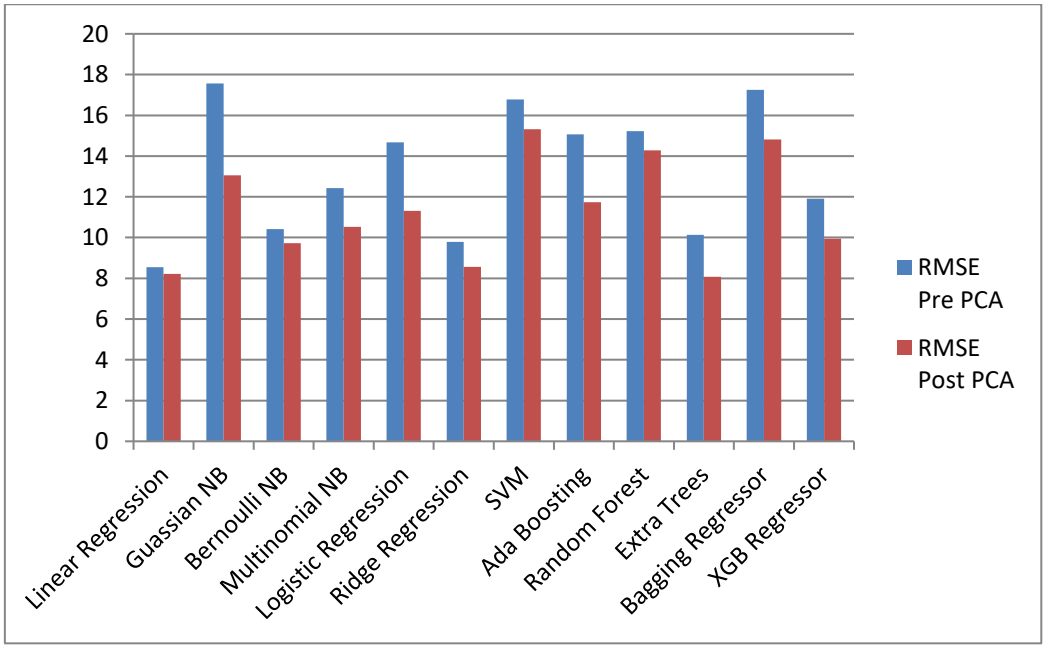

Figure 4. RMSE Pre and Post PCA

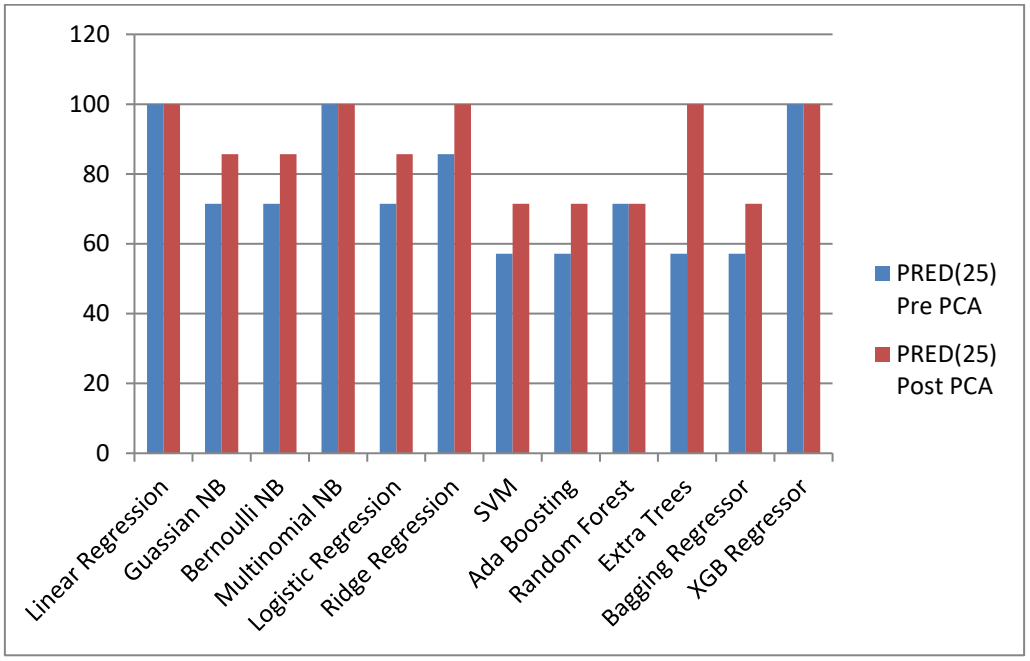

Figure 5. Prediction Accuracy Pre and Post PCA

5. Conclusion & Future Work

This paper has proposed the use of PCA for the feature extraction so that the maximum weightage features impacting the dependent variable may be extracted. The experimental results depicted by tables and plots, show that feature selection & appropriate weights improved the performance of the algorithms in terms of MMRE, RMSE and Pred (25), no matter which algorithm is used for the implementation. Although the size of the available dataset is very small, still the reduction gave significant results. The algorithms used are selected on the basis of existing literature.

Future work will take into consideration the ensemble techniques into consideration for further improvement of prediction accuracy and more tuning of parameters may be attempted through the use of various optimization techniques like grid-search and back propogation. Also giving the same number of parameters using PCA without dimensionality reduction will be implemented.

6. References

[1] Satapathy S. M., Rath, S. K., "*Empirical assessment of machine learning models for agile software development effort estimation using story points*" Innovations in Systems and Software Engineering, Springer, 1-10 (2017)

[2] Satapathy S. M., Rath, S. K., "*Empirical assessment of machine learning models for agile software development effort estimation using story points*" Innovations in Systems and Software Engineering, Springer, 1-10 (2017)

[3] Malgonde, O. and Chari, K., "*An ensemble-based model for predicting agile software development effort", Empirical Software Engineering*, pp.1-39 (2018)

[4] A. Trendowicz and R. Jeffery, "*Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success*", Springer Publishing Company, Incorporated, 2014, p. 469.

[5] *M. Jørgensen and M. Shepperd, "A systematic review of software development cost estimation studies," IEEE Trans.Software Eng., vol. 33, no. 1, pp. 33–53, 2007.*

[6] *Ziauddin, et. al, (2012) 'An effort Estimation Model for agile Software Development',* Advances in Computer Science and its Applications (ACSA), Worlds Science Publisher, 2(1), ISSN 2166-2924.

[8] Arora, Amandeep Singh, Linesh Raja, and Barkha Bahl. "Data centric security approach: A way to achieve security & privacy in cloud computing." *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies* (ICIoTCT). 2018.

[9] Poonia, Ramesh C., et al., eds. Smart Farming Technologies for Sustainable Agricultural Development. *IGI Global*, 2018.

[9] Kaushik, A., Verma, S., Singh, H.J. and Chhabra, G.," *Software cost optimization integrating fuzzy system and COA-Cuckoo optimization algorithm*" International Journal of System Assurance Engineering and Management, 8(2), pp.1461-1471. 2017

[10] V. Vapnik, S. E. Golowich, and A. Smola, Support vector method for function approximation, regression estimation and signal processing, 1996.

[11] Advanced Computing and Intelligent Engineering", Springer Science and Business Media LLC, 2020

[12] Saroha, M. and Sahu, S., 2015, May. Tools & methods for software effort estimation using use case points model—A review. In *International Conference on Computing, Communication & Automation* (pp. 874-879). IEEE

[13] Trendowicz, A. and Jeffery, R., 2014. Principles of effort and cost estimation. In *Software project effort estimation* (pp. 11-45). Springer, Cham

[14] Mahapatra, S., Kumar, A., Sharma, A. and Sahu, S.S., 2020. Effect of Dimensionality Reduction on Classification Accuracy for Protein–Protein Interaction Prediction. In *Advanced Computing and Intelligent Engineering* (pp. 3-12). Springer, Singapore.

[15] Kaushik, A., Tayal, D.K. and Yadav, K., 2020. A Fuzzy Approach for Cost and Time Optimization in Agile Software Development. In *Advanced Computing and Intelligent Engineering* (pp. 629-639). Springer, Singapore.