Statistical Learning
Data Analysis Project 2

STITCHING SUCCESS

Advanced Analysis of Garment Workers Productivity Dataset

Prepared By

Group 11

Ruwinda Rowel - s15654

Darshi Yashodha - s15584

Sithmi Pehara - s15494

Anjana Jayasinghe - s15627

Abstract

This report presents an advanced analysis conducted by utilizing machine learning algorithms. It was built upon insights gleaned from a prior descriptive analysis with the aim of constructing a model to predict the actual productivity of garment employees. For this task, we used a comprehensive garment workforce productivity dataset sourced from Kaggle. This innovative approach provides valuable insights for management, HR professionals, investors, and researchers in the garment industry. By employing predictive modelling, our study not only enhances decision-making processes but also facilitates efficient resource allocation. Moreover, it contributes to a deeper understanding of the dynamic factors influencing productivity in the realm of garment workforce management.

Table of Contents

List of Figures

List of Tables

Introduction

The garment industry, a cornerstone of global manufacturing, plays a pivotal role in economies worldwide, contributing significantly to employment and trade. As this industry thrives on efficiency, the importance of workforce productivity cannot be overstated. This report embarks on a journey to unravel the intricate dynamics of garment workforce productivity, with a focus on constructing a predictive model. Designed to offer a comprehensive insight into the factors shaping productivity within this critical sector, this model serves as a valuable resource for management, HR professionals, researchers, and investors alike.

Description of the Question

Many nations, especially those in developing and emerging economies like Bangladesh, Pakistan, India, Sri Lanka, Vietnam, Cambodia, Ethiopia, and Turkey, have woven substantial segments of their economies around garment and textile manufacturing. Despite technological advancements, including robotics, that have undoubtedly enhanced various industries, the garment industry faces unique challenges. Unlike solid materials, fabric is tricky for robots as it bends and stretches, requiring constant minute adjustments – something humans do naturally but machines find challenging. Thus, the garment industry still heavily relies on human participation. Consequently, the garment industry often falls short of its targeted productivity owing to the diversity of employee aptitudes and work pace. This discrepancy leads to substantial losses. To address this issue, predicting productivity becomes imperative. Enabling decision-makers to anticipate productivity based on available resources and the human workforce fosters effective planning, streamlined decision-making processes, and optimal resource allocation. This, which will be a valuable tool for achieving business goals and

meeting targets within the correct timeframe, not only benefits owners, HR, and management but also extends advantages to researchers and investors. Therefore, our primary objective is,

- To construct a reliable model to predict the actual productivity of garment employees by identifying the factors with high potential to influence actual productivity.

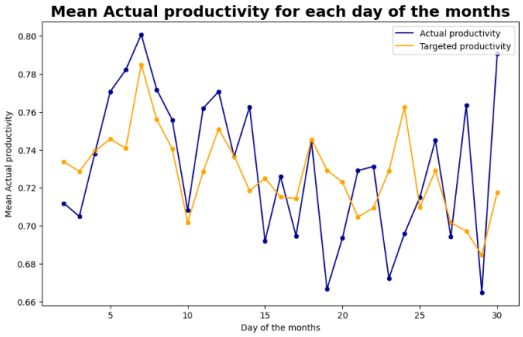
Description of the Data Set

The "Productivity Prediction of Garment Employees" dataset, sourced from the UCI Machine Learning repository and made available on Kaggle, presents a comprehensive collection of attributes relevant to the garment manufacturing process and the productivity of the employees. The dataset spans from 1st January 2015 to 4th March 2015. The dataset comprises of 1197 observations and 15 variables.

No.	Attribute	Type of variable	Comments
1	date	Categorical-Ordinal	Date in MM-DD-YYYY
2	quarter	Categorical-Ordinal	A portion of the month. A month was divided into four quarters
3	department	Categorical-Nominal	Associated department with the instance
4	day	Categorical-Ordinal	Day of the Week
5	team	Categorical-Nominal	Associated team number with the instance
6	targeted_productivity	Numerical-Continuous	Targeted productivity set by the Authority for each team for each day.
7	smv	Numerical-Continuous	Standard Minute Value, it is the allocated time for a task
8	wip	Numerical-Discrete	Work in progress. Includes the number of unfinished items for products
9	over_time	Numerical-Discrete	Represents the amount of overtime by each team in minutes
10	incentive	Numerical-Continuous	Represents the amount of financial incentive (in BDT) that enables or motivates a particular course of action
11	idle_time	Numerical-Continuous	The amount of time when the production was interrupted due to several reasons
12	idle_men	Numerical-Discrete	The number of workers who were idle due to production interruption
13	no_of_style_change	Numerical-Discrete	Number of changes in the style of a particular product
14	no_of_workers	Numerical-Discrete	Number of workers in each team
15	actual_productivity	Numerical-Continuous	The actual % of productivity that was delivered by the workers. It ranges from 0-1

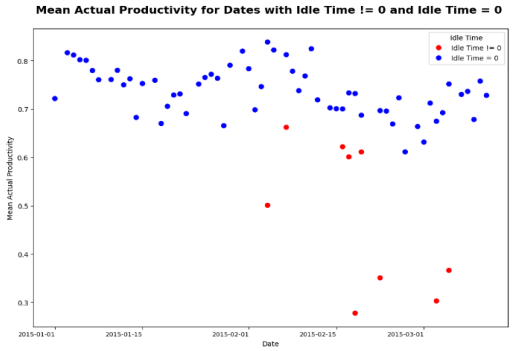
Table 1 : Description of the variables

Important Results of the Descriptive Analysis



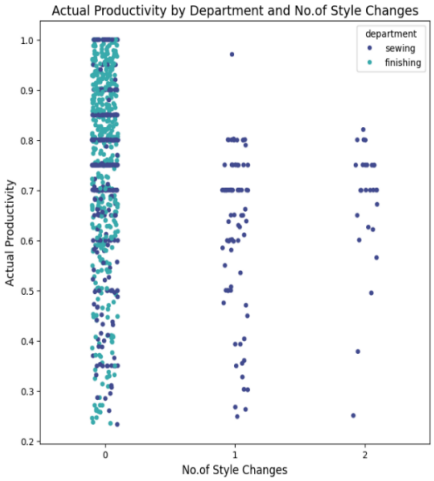
Towards the end of the month, a noticeable decrease in productivity is evident, potentially attributed to worker fatigue. However, after the 25th, a sudden surge is observed, possibly linked to compensation on payday.

Figure 1: Mean Actual Productivity for Each Day of the Month



Days with idle time correspond to lower productivity, indicating that idle time can have a detrimental effect on overall productivity

Figure 2: Mean Actual Productivity for Dates with Idle Time !=0 & Idle Time = 0



When examining the overarching trend of actual productivity and No. of style changes, a discernible downward relationship emerges. This suggests that an increase in the No.of style changes disrupts the workflow and impacts productivity. Furthermore, a closer look at individual departments reveals that sewing tasks are more susceptible to the influence of style changes, while finishing tasks appear to remain unaffected.

Figure 3: Actual Productivity by Department & No. of Style Changes

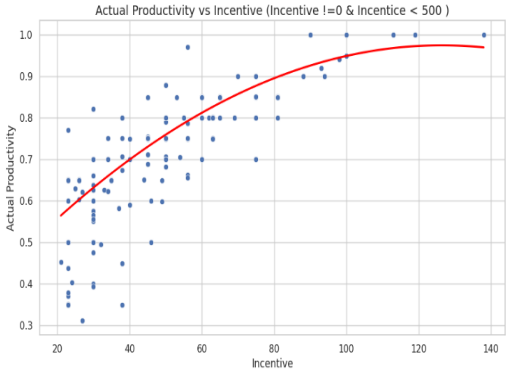
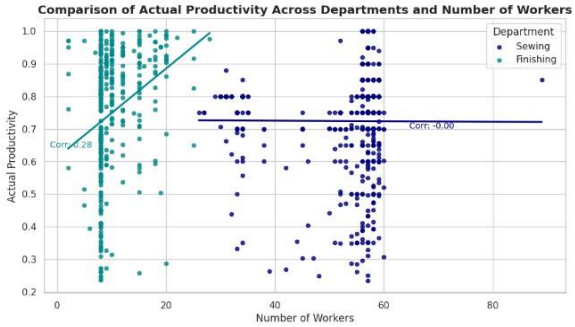


Figure 4: Actual Productivity Vs 0 < Incentive < 500

When the incentive falls within the range of 0 < incentive < 500, actual productivity appears to increase. However, beyond a certain point, the rate of this increase seems to decelerate. Nevertheless, when the incentive surpasses 100 BDT, all data points reflect the highest productivity level, marked as 1. This pattern suggests a relationship between incentive levels and actual productivity, with a notable shift after reaching a specific threshold.



When examining the No. of workers with departments, it is evident that the finishing department, despite having a smaller workforce, demonstrates an increase in productivity as the No. of workers grows. In contrast, the sewing department, with a larger workforce, does not show any discernible relationship with the No. of workers.

Figure 5: Comparison of Actual Productivity across Departments & No. of Workers

There seem to be three main groups of SMV values, and for high SMV, lower productivity is evident. This can be explained by the widely accepted industry fact that higher SMV tasks take longer, are more complex, and result in lower productivity.

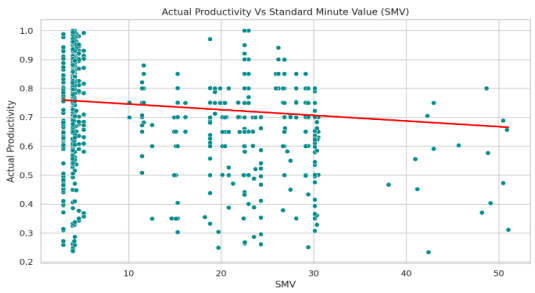


Figure 6: Actual Productivity Vs SMV

Important Results of the Advanced Analysis

Before we start to do the advanced analysis, we looked for outliers using Mahalanobis distance criteria and found that there are 69 multivariate outliers out of 1197 data points which belongs to the 'idle_time', 'idle_men', and 'no_of_style_changes' variables. However, after a closer inspection, we found that removing these outliers would result in a similar effect to removing those variables. Hence, we did not remove any outliers when conducting our advanced analysis.

Next, we tried to identify clusters by plotting the principal component score plot and using the DBSCAN algorithm but found out that there is only one cluster. Therefore, we decided to fit one regression model for the whole dataset.

Since we are aiming to generalize our model to predict actual productivity for any garment company, we dropped the 'team' variable as it is specific to the garment company to which this dataset belongs.

Lastly, we encoded all the qualitative variables to numerical values using the one-hot encoder function in Python.

Multiple Linear Regression (MLR)

The decision to employ MLR as our first option emerged because it is the simplest and fundamental model which align with the characteristics of our data set. As mentioned before we did not use ‘team’ variable to build the model because of generalization issues.

We utilized both forward selection method and backward elimination method to select the best variables for our model. Forward selection procedure gave us the model with all the variables while backward elimination selected the model with only ‘targeted_productivity’ variable. Below are the results we obtained.

Backward Elimination Method

	RMSE	**R^2**
Training set	0.158	0.167
Test set	0.152	0.222

Table 2: Evaluation metrics for Backward Elimination method

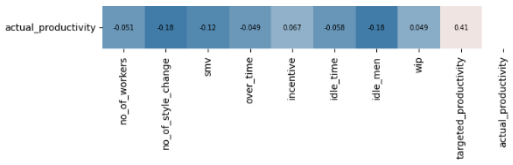
Forward Selection Method

	RMSE	**R^2**
Training set	0.148	0.256
Test set	0.145	0.283

Table 3: Evaluation metrics for Forward Selection method

Considering these values it is evident that forward selection method gives better model with lower RMSE values. Furthermore to check the validity of the given model we conducted a residual analysis,

1. **Linear relationship:**



The correlation plot underscores small associations between predictors and the response variable, indicating a limited strength of linearity in the model.

Figure 7: Correlation values for predictor variables

2. **Multicollinearity:**

The variables 'smv' and 'no_of_workers' exhibit Variance Inflation Factor (VIF) values exceeding 10, indicating the presence of multicollinearity

3. **Independence & Homoscedasticity**

The residual vs. predicted plot reveals a lack of random scattering around the zero-centre line, accompanied by a trend line. This implies a violation of both homoscedasticity and independence of residuals.

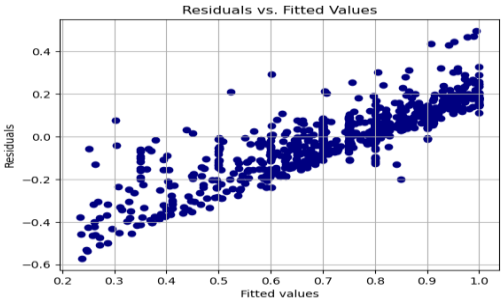


Figure 8: Residual Vs Fitted Values for MLR

4. **Multivariate Normality:**

- Shapiro-Wilk Test - Statistic: 0.9497618079185486
- P-value: 1.5363998103639508e-17

The normality assessment, conducted through the Shapiro-Wilk test together with Q-Q plot indicates the departure from normal distribution assumptions in the model's residuals.

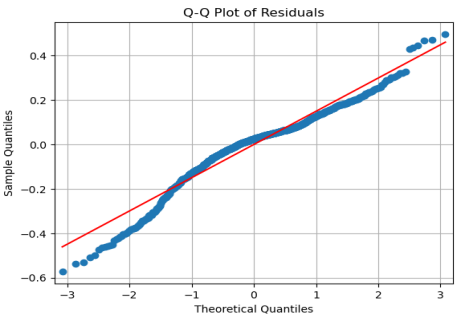
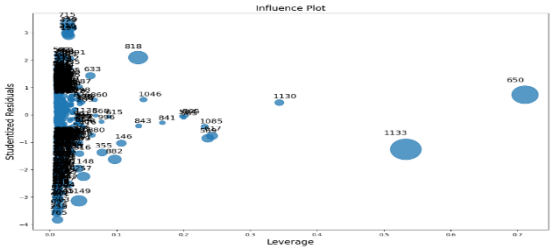


Figure 9: Q-Q plot of Residuals for MLR

Influential Points:



The Influential plot reveals the presence of outliers and influential observations

Figure 10: Influential plot

Violation of the assumptions of MLR model and existence of outliers and influential points indicate the need of more robust regression techniques.

Regularization Methods:

Following the MLR analysis, the incorporation of regularization methods is recommended. These technique addresses potential pitfalls observed in the MLR model, providing a valuable means to improve overall robustness and reliability of the model.

Results obtained for Ridge regression, Lasso regression and Elastic Net Regression:

Model	Best λ	Train RMSE	Test RMSE	Train R^2	Test R^2
Ridge	10.0	0.14990	0.14425	0.2546	0.2958
Lasso	0.0004	0.14985	0.14425	0.2551	0.2958
Elastic Net	0.0007	0.14985	0.14425	0.2551	0.2958

Table 4: Evaluation metrics for Ridge, Lasso, & Elastic Net

Even though test RMSE values are similar for all three methods Elastic Net and Lasso will be a good fit as they can handle the effect of outliers somewhat better than Ridge regression. And they give a rather simple model with removing 'quarter' variable from the model.

Principal Components Regression (PCR)

Principal Components Regression utilizes the Principal Components of a predictor matrix and uses an optimal number of Principal Components to predict a given response variable. Since PCR applicable only for quantitative data we only considered the 9 quantitative variables in our data set to predict the actual productivity. Using cross validation, we found the optimal number of components to be 9 indicating no dimensionality reduction. As the given table shows the model has over fit the data with the sudden increase in the test value.

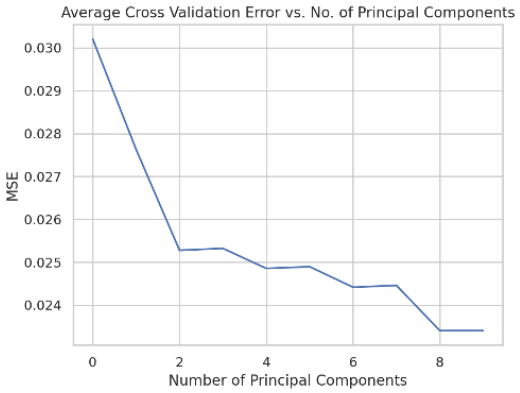


Figure 11: Average CV error Vs. No. of principal components

	Training	Testing
RMSE	0.1511	0.2128
R^2	0.2419	-0.5324

Table 5: Evaluation metrics for PCR

Partial Least Squares Regression (PLSR)

Overcoming the limitations of PCR we fitted a PLSR model to our dataset. PLSR considers both the predictors and the response in its tabulation thus we hoped to overcome any limitations we may have faced.

Once we conducted a grid search for the optimal number of components utilizing cross validation we found that the optimal number of components to be 10 where we saw a slight reduction in the RMSE for both training and testing data.

	Training	Testing
RMSE	0.1445	0.1427
R^2	0.3073	0.3104

Table 6: Evaluation metrics for PLSR

Considering the results we obtained for regularization methods it is evident that there is room for improvement for our model. So we decided to further improve our model by employing tree based methods.

Tree based methods:

Tree-based algorithms are robust to outliers and multicollinearity issues, and they can capture non-linear relationships between predictors and the response variable.

When examining the Variance Inflation Factor (VIF) values among predictors, we observed multicollinearity issues. Furthermore, when assessing the correlation between the response variable (actual_productivity) and predictor variables, most predictors showed weak correlations with the response variable. This suggests a potential non-linear and joint relationship between predictors and the response variable.

The specialty of tree-based modelling lies in the fact that there are not many assumptions to satisfy, and there is no need to scale the data before modelling.

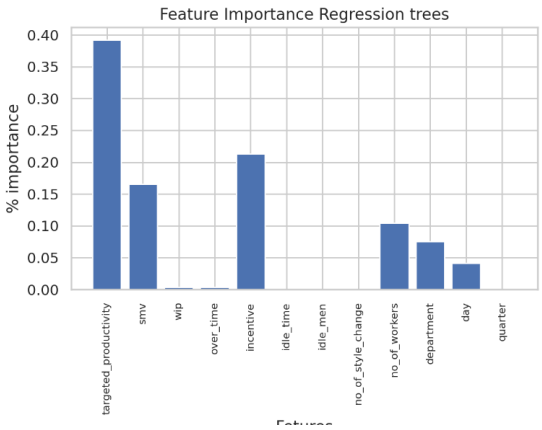
Regression Trees

Regression trees partition a data set into smaller groups and then fit a simple model (constant) for each subgroup. Upon training the model and tuning parameters the following results were obtained.

We can observe a somewhat higher training R-squared value than testing R^2 which is indication of slight over-fitting.

	Training	Testing
RMSE	0.1212	0.1378
R^2	0.5128	0.3577

Table 7: Evaluation metrics for Regression Trees



Next, we computed the percentage importance of each variable in the model and found varibles, 'ilde_time', 'idel_men', 'no_of_style_changes' and 'quarter' is not important to the model. Variables, 'targeted_productivity, 'smv' and 'incentives' are more important to the model.

Figure 12: Feature Importance plot of Regression Trees

After fitting the model by removing all unimportant variables, model controlled over-fitting and further increases the testing R^2 by reducing testing RMSE.

	Training	Testing
RMSE	0.1217	0.1350
R^2	0.5082	0.3829

Table 8: Evaluation metrics for Regression Trees after removing unimportant variables

Random Forest

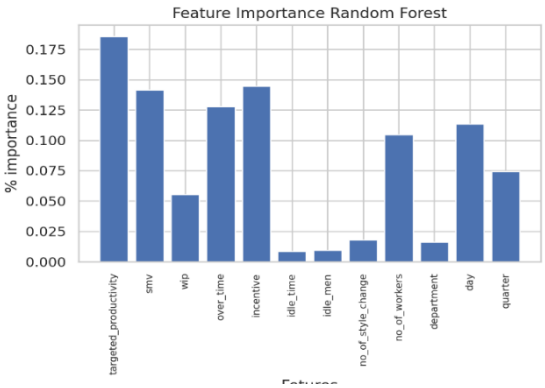
Random Forest is a machine learning algorithm that uses regression trees as its base learning model. The underlying assumption of Random Forest is that each tree will make different mistakes, so combining the results of multiple trees should be more accurate than any single tree. This way, the model fits several decision tree regressors on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Upon training the model and tuning parameters the following results were obtained.

	Training	Testing
RMSE	0.0526	0.1213
R^2	0.9079	0.5015

Table 9: Evaluation metrics for Random Forest

We can observe a considerable higher training R^2 value than testing R^2 which is indication of over-fitting.



According to the variable importance, the predictor variables like 'targeted_productivity', 'smv' and 'incentives' are more important to the model.

Figure 13: Feature Importance plot of Random Forest

XGBoost

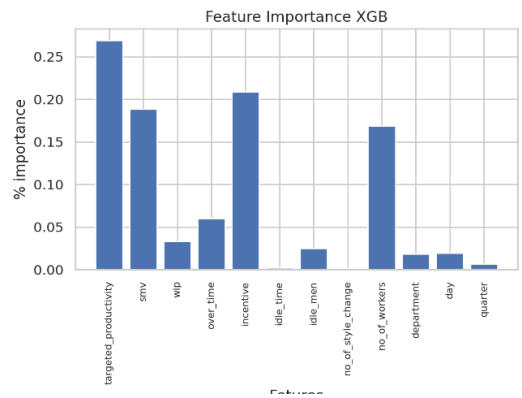
XGBoost is a powerful open-source tool designed to help build better models and works by combining decision trees and gradient boosting. It is a boosting algorithm that uses bagging, which trains multiple decision trees and then combines the results. It allows XGBoost to learn more quickly than other algorithms but also gives it an advantage in situations with many features to consider. Due to the popularity of XGT boost outperforming Random Forest Classifier in various aspects, it was also decided to run XG Boost on the Garment workers' productivity Dataset.

Upon training the model and tuning parameters the following results were obtained.

	Training	Testing
RMSE	0.1047	0.1203
R^2	0.6357	0.5095

Table 10: Evaluation metrics for XGBoost

When comparing testing R^2 value of XGB with random forest, it does not literally increase, but training R^2 value has decreased in XGB when comparing to random forest. This implies XGM controls the over-fitting issue.



According to the variable importance, varibles 'no_of_style_changes' and 'idle_time' are not important to the model. And note that 'targeted_productivity', 'smv', 'incentives' and 'no_of_workers' are more important to the model.

Figure 14: Feature Importance plot of XGBoost

Then we refited the model by removing not important variables and tune the parameters of the model. The model performance are as follows.

	Training	Testing
RMSE	0.1044	0.1208
R^2	0.6382	0.5195

Table 11: Evaluation metrics for XGBoost after removing unimportant variables

Since the RMSE is high for random forest compared to XGB, it is conclusive that XG Boost to outperform Random Forest Classifier in our analysis. Also XGB controls the over-fitting issue in our study.

Issues Encountered and Proposed Solutions

In the course of our advanced analysis, several challenges occurred. To begin, we encountered data quality issues, including inconsistent entries and missing values. Our solution involved implementing data cleaning and imputation techniques. Moving into the analysis, we identified violations of key assumptions in the MLR model. To mitigate this, we implemented diagnostic checks. In regularization techniques, due to unexpected equality in results, we used cross-validation to optimize regularization parameters. In PCR, challenges arose as dimensionality failed to reduce as anticipated. Additionally, difficulties in generating a scree plot prevented us from identifying components in PLSR. We explored innovative approaches to address these issues. Finally, in tree-based methods, some variables exhibited zero importance. The solution involved the removal of these variables, followed by model refitting, optimizing the predictive power of the model. This troubleshooting approach, addressing challenges at each stage of the analysis, demonstrates our commitment to producing reliable and robust insights in the face of analytical hurdles.

Discussion and Conclusions

The R^2 and RMSE values of all types of optimal models (on Test data and Training data) are summarized as follows:

	Training		Testing	
	RMSE	R^2	RMSE	R^2
MLR (Forward Selection)	0.148	0.256	0.145	0.283
MLR(Backward Elimination)	0.158	0.167	0.152	0.222
Ridge	0.14990	0.2546	0.14425	0.2958
Lasso	0.14985	0.2546	0.14425	0.2958
Elastic net	0.14985	0.2551	0.14425	0.2958
PCR	0.1511	0.2419	0.2128	-0.5324
PLSR	0.1445	0.3073	0.1427	0.3104
Regression Tree	0.1217	0.5082	0.1350	0.3829

Random Forest	0.0526	0.9079	0.1213	0.5015
XGBoost	0.1047	0.6357	0.1203	0.5095

Table 12: Summary of all R^2 & RMSE values

As uncovered during the stepwise Advanced Analysis, XGBoost Regressor is the best model for predicting the productivity of garment workers with a relatively high-Test R^2 and lowest testing RMSE among all algorithms, along-side a minimal difference between train and test R^2 implying that control of over-fitting.

Hence, the data-product will be developed containing a back-end feature where the input variables of the factors affecting to the garment workers' productivity will be analysed using a Hyper-parameter tuned XGBoost Regressor to give an output of the productivity score of garment workers.

References

1. https://online.stat.psu.edu/stat508/lesson/11
2. https://www.analyticsvidhya.com/blog/2021/04/distinguish-between-tree-based-machine-learning-algorithms/#:~:text=Tree%2Dbased%20is%20a%20family,value%20according%20to%20the%20features.
3. https://www.statology.org/partial-least-squares-in-python/
4. https://www.kaggle.com/code/phamvanvung/partial-least-squares-regression-in-python
5. https://www.statology.org/principal-components-regression-in-python/
6. Kern C, Klausch T, Kreuter F. Tree-based Machine Learning Methods for Survey Research. Surv Res Methods. 2019 Apr 11;13(1):73-93. PMID: 32802211; PMCID: PMC7425836.
7. Hasan, Hasibul & Nuha, Nigar & Gomes, Paul & Lameesa, Aiman & Alam, Md. Ashraful. (2023). Interpretable Garment Workers' Productivity Prediction in Bangladesh Using Machine Learning Algorithms and Explainable AI.

Appendix

1. Link for the dataset: Productivity Prediction of Garment Employees
2. The python code used in our project is conveniently accessible through our GitHub repository. You can find the Colab notebook containing all relevant code by following this GitHub link: https://github.com/ruwindarowel/Data-Analytics-and-Machine-Learning-to-Predict-Workers-Productivity/tree/branch1