



ISM 6251 – Final Group Project

A CLASSIFICATION MODEL FOR OPERATIONAL EFFICIENCY

Mude Jayaprakash Naik
Nairi Keeney
Arushi Panwar




Table of Contents

INTRODUCTION	2
PROJECT TITLE	2
BUSINESS CONTEXT	2
BUSINESS PROBLEM	2
PREVIOUS WORKS	3
OBJECTIVE	4
DATA DETAILS	5
DATASET SELECTION	5
EXTRACTION DETAILS	5
THIS DATA WAS COLLECTED MANUALLY AND INCLUDES THE MAIN ATTRIBUTES OF THE GARMENT MANUFACTURING PROCESS AND THE PRODUCTIVITY OF THE EMPLOYEES, WHICH IS ALSO VALIDATED BY INDUSTRY EXPERTS.	5
DATASET OVERVIEW	5
ANALYSIS PLAN	6
DATA LOADING	6
EXPLORATORY DATA ANALYSIS (EDA)	6
DATA PREPROCESSING	6
MODEL BUILDING AND EVALUATION	8
MODEL SELECTION	8
MODEL TUNING AND OPTIMIZATION	8
MODEL EVALUATION	9
ERROR ANALYSIS	9
BUSINESS VALUE	10

Tables

Table I: Previous Works	3
Table II: Dataset Overview	5
Table III: Type of Errors (FP and FN) and their business impact	9

Introduction

Project Title

Predicting Team-Level Productivity in Garment Manufacturing: A Classification Model for Operational Efficiency

Business Context

The garment industry is one of the most labor-intensive sectors globally. In countries like Bangladesh, where this dataset originates, the garment sector is a key economic driver. Ensuring that production teams consistently meet their daily productivity targets is critical for profitability and on-time delivery. However, these decisions are often reactive rather than proactive, leading to inefficiencies, increased labor costs, and missed deadlines.

This project seeks to develop a predictive classification model that can identify whether a garment production team is likely to meet its productivity target for a given workday, based on operational variables such as team size, task type, overtime hours, idle time, and more.

This prediction would allow factory managers to identify underperforming teams early in the day and intervene proactively by adjusting workloads, redistributing tasks, or providing support before productivity suffers. This transforms workforce management from reactive to data-driven and strategic.

Business Problem

At the beginning of the workday, identify if a team will **meet their daily productivity targets** based on available operational features.

Previous Works

Past research on garment employee productivity has primarily focused on **regression-based approaches** or **multi-class classification**, aiming to predict exact productivity scores or categorize performance levels. Key studies include

Table I: Previous Works

Study	Approach	Key Contributions
Ali et al. (2019)	Deep Neural Network (Regression)	Demonstrated deep learning's ability to capture complex labor productivity patterns, achieving lower MAE than linear regression. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8820486
Hossain et al. (2021)	Random Forest (Classification)	Compared algorithms like SVM and decision trees; found Random Forest to offer the best accuracy for productivity classification. https://www.researchgate.net/publication/351901793
Hossain et al. (2021)	Explainable AI (SHAP + ML)	Used SHAP with Random Forest and XGBoost to explain key drivers of productivity (e.g., idle time, overtime), advancing model interpretability. https://www.researchgate.net/publication/368642549
Zendy (2021)	Comparative Regression Study	Showed that Random Forest consistently produced the lowest error rates (MAE: 0.0787), confirming its reliability in industrial datasets. https://zendy.io/title/10.33480/techno.v18i1.2210
Balla et al. (2021)	Ensemble Learning	Introduced hybrid models using AdaBoost and Bagging; Random Forest again achieved the highest accuracy (98.3%) and best RMSE (0.1423). https://www.inderscience.com/offers.php?id=118183

Objective

While previous studies have focused on predicting exact productivity values or segmenting them into multiple classes, our project takes a more actionable approach by reframing the problem as a **binary classification task**:

“Will a team meet its productivity target today?”

By using team-level operational features, we aim to predict whether a production team will achieve its daily target. The target variable is defined as:

- **1** → Team did not meet the target
- **0** → Team met or exceeded the target

This binary framing transforms the model into a practical decision-support tool. It empowers factory supervisors to act early, adjusting resources, workloads, or incentives, before productivity slips, enabling proactive rather than reactive management.

Data Details

Dataset Selection

The publicly available dataset chosen for this assignment is the ‘Productivity Prediction of Garment Employees’ dataset from the UCI Machine Learning Repository. The dataset contains information on 1197 Instances with missing values. This dataset has 15 variables, including the Target variable.

Here is the link to the dataset:

<https://archive.ics.uci.edu/dataset/597/productivity+prediction+of+garment+employees>

Extraction Details

This data was collected manually and includes the main attributes of the garment manufacturing process and the productivity of the employees, which is also validated by industry experts.

Dataset Overview

Table II: Dataset Overview

Feature	Role	Type	Description	Missing Values
date	Feature	Date	Date in MM-DD-YYYY	no
quarter	Feature	Categorical	A portion of the month. A month was divided into four quarters	no
department	Feature	Categorical	Type of department (sewing, finishing, etc.)	no
day	Feature	Categorical	Day of the week	no
team	Feature	Integer	Identifier for each production team	no
targeted_productivity	Feature	Continuous	Productivity target set for each team for each day.	no
smv	Feature	Continuous	Standard Minute Value - time allocated for a task	no
wip	Feature	Integer	Work-in-progress; Includes number of unfinished items for products	yes
over_time	Feature	Integer	Minutes of overtime worked by each team.	no
incentive	Feature	Integer	Bonus paid to motivate performance (in Bangladeshi Taka - BDT)	no
idle_time	Feature	Integer	Minutes the team spent inactive during working hours.	no
idle_men	Feature	Integer	Number of workers who were idle during downtime.	no
no_of_style_change	Feature	Integer	Number of changes in the style of a particular product.	no
no_of_workers	Feature	Integer	Number of workers in the team	no
actual_productivity	Target	Continuous	Team's achieved productivity (0 to 1 scale).	no

Analysis Plan

Data Loading

- We will start by loading the dataset from the UCI Machine Learning Repository.
- Checking for data shape and columns.
- Creating the target variable **productivity_met** as:
 - If `actual_productivity < targeted_productivity`, then 1 (Team did not meet the target)
 - If `actual_productivity >= targeted_productivity` then 0 (Team met or exceeded the target)
- Dropping the `actual_productivity`, `targeted_productivity`, and other non-relevant columns if any to make sure that the features represent operational realities like team size, idle men, and incentives.

Exploratory Data Analysis (EDA)

- Checking for data types, **null values**, and basic statistics, ensuring no anomalies.
- Identifying different types of variables (categorical vs numerical) and understanding their data distributions via visualizations.
- **Target Distribution:** Visualizing the target variable to understand class imbalance. If the target is imbalanced, we will be using appropriate rebalancing techniques and will adjust evaluation metrics accordingly.
- **Feature Correlation:** Using correlation matrices, we will explore how features relate to one another and the target variable. Based on this, we will finalize what features need to be dropped/kept and what new features need to be derived.

Data Preprocessing

- **Handling Missing Data:** We will be using the following imputing methods
 - For numerical variables: mean or median (based on the distribution).
 - For categorical variables: the most frequent category (mode).
- **Feature Engineering:** We will be creating features based on the EDA. Apart from these, for Deeper Insight & Better Prediction, we are planning to add the following derived variables (if not created already) that improve the model's ability to reflect real-world conditions more accurately:
 - **SMV per worker**
 - `smv / no_of_workers`

- To normalize task complexity against workforce size.
- **Idle ratio**
 - $\text{idle_time} / (\text{total_time_available})$
 - A normalized view of downtime. Raw idle time doesn't account for team size or work hours—this ratio gives better comparability.
- **Work in Progress per worker**
 - $\text{wip} / \text{no_of_workers}$
 - Measures the average workload each worker is managing. High WIP per worker can predict burnout or bottlenecks.
- **Incentive per worker**
 - $\text{incentive} / \text{no_of_workers}$
 - Helps assess motivation effectiveness. A high total incentive might not mean much if the team size is large, normalizing gives better insight.

These features offer richer, interpretable signals to both the model and business decision-makers, something previous works did not explore.

- We will be dropping unwanted variables based on EDA.
- We will be encoding the categorical variables depending on whether they are nominal, ordinal, or binary.
- **Train-Test Split:** We will split the data into training (70-80%) and testing (20-30%) sets, ensuring stratified sampling to preserve the target distribution.
- **Feature Scaling:** If needed, will normalize or standardize the numerical features (after splitting the data into train and test sets), depending on the model chosen, to ensure consistent measurement and model performance.

Model Building and Evaluation

Model Selection

- **Baseline Model:** We will start with a simple model - **Logistic Regression** to establish baseline performance.
- **Model Comparison:**
 - **Logistic Regression:** A good starting point for binary classification as it offers interpretability and works well for linear relationships.
 - **Random Forest:** A more powerful model that can capture non-linear relationships and can better handle interactions between features.
 - **Gradient Boost Models (XGBoost/LightGBM):** Generally, these models lead to performance improvements. Particularly effective on imbalanced datasets.

Model Tuning and Optimization

- **Hyperparameter Tuning:** We will use **GridSearchCV** or **RandomizedSearchCV** to optimize key hyperparameters like C, penalty etc. for logistic regression and max_depth, n_estimators, and learning_rate for tree-based models, to get the best model configuration for improved performance.
- **Cross-Validation:** If computationally possible, will perform **10-fold cross-validation** if not, **5-fold cross-validation**. This ensures that the model generalizes well and doesn't overfit to the training data.
- **Precision-Recall and ROC Curve:** Will generate whatever curve makes sense based on data imbalance and will calculate the area under the curve (AUC):
 - **AUC-ROC:** Tells us how well the model discriminates between teams that met and did not meet the target across different decision thresholds.
 - **Precision-Recall Curve:** Tells us how well the model identifies the minority class (teams meeting the target) and is more informative than ROC if the data is imbalanced
- **Optimal Point:** Based on the evaluation metric chosen, the threshold for the optimal point is calculated, and the model will be evaluated at this threshold (best).

Model Evaluation

Evaluation Metrics

- **Accuracy:** Measures the overall proportion of correctly predicted productive teams. If the data is imbalanced, other metrics provide better insights.
- **Precision:** Of all predicted productive teams, how many were truly productive? This is crucial to avoid false positives.
- **Recall:** Of all truly productive teams, how many did we correctly identify? High recall helps to ensure that the most productive teams are recognized.
- **F1-Score:** A single metric that balances precision and recall, especially useful when the data is imbalanced.

We will compare these metrics across different models to understand how each model is performing.

Error Analysis

- **Misclassified Instances:** We will identify where the model made errors (False Positives and False Negatives) with the help of the **Confusion Matrix** at the best threshold calculated.
- **Business Interpretation of Errors**

Table III: Type of Errors (FP and FN) and their business impact

Error	Meaning	Business Impact
False Positive (FP)	Predicted - team will meet target, but they don't	Wasted incentives/resources Missed opportunity to intervene Reduced trust in decision
False Negative (FN)	Predicted - team won't meet target, but they do	Missed recognition Underutilization Possible drop in motivation

Based on the error analysis and metric comparison, we will determine the best-performing model.

Business Value

- Supports **data-driven supervision** over intuition-based decisions
- Empowers supervisors to make **proactive decisions**
- Boost team motivation through **fair recognition**
- **Reduce costs** from overtime and production delays

Additionally, when combined with business analytics, resource allocation and planning can be optimized via root cause analysis.