Garment Worker Productivity Forecasting Report Using Machine Learning Applications for Business

Submitted by:

Abhishek Kumar Nishad

Shreyasi Singh

Vishwanath Singh

Course: ML Applications for Business
Institution: Indian Institute of Technology (IIT) Jodhpur
Date of Submission: 15 October 2025

Introduction

This report presents a comprehensive analysis of garment worker productivity using a data-driven, machine learning–based approach. The study explores how operational, structural, and motivational factors collectively shape worker efficiency and overall production outcomes. Through a combination of regression, classification, and clustering techniques, the project identifies key predictors of productivity and evaluates multiple models including Neural Networks (NNET), GLMNET (Regularized Regression), and CART (Classification and Regression Tree). The comparative analysis determines the most effective model for forecasting productivity and supporting actionable business decisions.

The findings offer practical insights into workforce management and production optimization within the garment industry, demonstrating how machine learning applications can enhance decision-making, improve operational efficiency, and enable data-supported strategic planning.

Index: Garment Worker Productivity Forecasting Report

This report analyzes the structural factors and operational metrics driving worker efficiency, utilizing a multi-model machine learning approach.

Section I: Data Foundation and Preparation

Chapter	Topic	Key Finding / Rationale	Correlated Next Step
1.	**Executive Summary**	Overview of best model (NNET), final accuracy (R2=0.353), and key intervention points.	(N/A – Summary of Report)
2.	**Data Context & Cleaning**	Summary of all 11 input variables, why WIP and date were removed, and the final sample size (N=1160 rows).	**Action V:** Finalize **Model Saving and Deployment** using the filtered dataset.
3.	**Feature Engineering**	Rationale for **One-Hot Encoding** (categorical factors) and **Filtering** (Target≤1.0).	**Action V:** Ensure the final prediction pipeline includes the exact dummy coding and filtering logic.

Section II: Diagnostics and Structure

Chapter	Topic	Key Finding / Rationale	Correlated Next Step
4.	**Exploratory Data Analysis (EDA)**	**Structural Difference:** Finishing is higher than Sweing. Day has no effect. Incentive is the strongest linear driver (r=0.42).	**Action VI:** CART Rules lead directly to **Business Strategy Integration** (Intervention targeting Sweing department).
5.	**Multicollinearity (VIF)**	**Critical VIF up to 21.09.** Confirmed the necessity of Regularization (GLMNET) and non-linear modeling.	**Action II:** GLMNET's failure to achieve high accuracy motivates **Model Improvement** with XGBoost.

Section III: Model Performance and Strategy

7.	**Regression Model Comparison**	**NNET is Best** (R2=0.353, RMSE=0.137). Linear models are inadequate (R2≈0.33).	**Action II:** NNET accuracy is moderate; immediate need for **Model Improvement (XGBoost)** to raise R2.
8.	**Classification Accuracy (CART)**	**High Actionability** (Accuracy=82.3%). Best at identifying low-productivity days (Specificity 87.1%).	**Action VI:** CART rules are delivered to managers for **Targeted Intervention** (Worst segment: Low Incentive + Failed Target).
9.	**Unsupervised Analysis (Clustering/PCA)**	Data consists of a large "Normal" group and two small "Unique Operational" clusters.	**Action VI:** Use cluster profiles for **Tailored Management** (different training for different teams).
10.	**Conclusion & Roadmap**	Final selection and summary of operational insights.	**Actions III, IV, V:** Formal **Hyperparameter Optimization**, Final **Validation**, and **Deployment** planning.

Summary of Dataset Variables

I. Target Variable

Variable	Type	Implication for Model
actual_productivity (target)	Continuous (0-1)	**The variable being predicted.** Represents the team's achieved efficiency score.

II. Categorical and Time Variables (Factors)

These variables capture structural or time-based differences and were converted to dummy variables for modelling.

Variable	Type	Implication
department	Categorical	**Structural Driver.** Captures the baseline efficiency difference between finishing and sweing.
quarter	Categorical	**Seasonal/Time Effect.** Accounts for systematic variation across different business quarters.
day	Categorical	**Time/Schedule Effect.** Captures any residual weekday influence (though EDA showed a weak effect).
team	Categorical	**Unit/Grouping Effect.** Accounts for differences in team management, skill levels, or machinery quality.

III. Continuous and Discrete Numerical Variables

These variables represent costs, effort, goals, or resources and are the direct drivers of the target.

Variable	Type	Implication
targeted_productivity	Continuous	**Goal Metric.** The official target set for the team.
smv (Standard Minute Value)	Continuous	**Cost/Effort Metric.** Represents the standard time required for a garment. Negatively correlated with productivity.
over_time	Continuous	**Effort Metric.** Captures extra working hours. Highly collinear with smv and no_of_workers.
incentive	Continuous	**Motivational Driver.** Showed a strong relationship with the highest achieved productivity.
idle_time	Continuous	**Loss/Inefficiency Metric.** Time production was halted. Highly correlated with idle_men.
idle_men	Discrete (Count)	**Loss/Inefficiency Metric.** Number of workers sitting idle, contributing to high VIF redundancy.

no_of_style_change	Discrete (Count)	**Disruption Metric.** Measures changes to the production line setup.
no_of_workers	Discrete (Count)	**Resource Metric.** Total manpower available to the team.

IV. Excluded Variables

Variable	Reason for Exclusion
wip (Work in Progress)	Removed due to a high number of missing values (NAs), which would have severely reduced the sample size.
date	Removed due to high cardinality (too many unique values), which was impractical for initial cross-sectional regression models.

Initial Data Cleaning and Variable Exclusion

The initial data cleaning phase focused on maximizing data quality and sample size by removing columns that contained problematic or unmanageable data points.

Variables Removed

Two variables were explicitly removed from the dataset before feature engineering and modeling

began:

Variable	Reason for Removal	Impact on Modeling
date	High Cardinality	Dropped because using it directly (e.g., as dummy variables) would have created too many unique features for a relatively small dataset (N=1197). While crucial for time-series, it was omitted for the initial cross-sectional regression models.
wip (Work in Progress)	High Missing Values (506)	This column had **506 missing values** out of 1197 observations. Retaining it would have forced nearly half the dataset to be discarded (using na.omit) or required extensive, uncertain imputation.

Rationale for Action

1. **Preserving Sample Size:** The primary goal of removing wip was to preserve the maximum number of observations possible for model training. Since the dataset is moderate-sized (N=1197), dropping half the rows is detrimental to model learning.
2. **Focusing on Reliable Features:** By removing wip, the model focuses on predictors that are fully populated and reliable, ensuring the training set reflects complete operational data.
3. **Avoiding Overfitting (date):** Removing the date column prevented the model from potentially overfitting to specific dates rather than learning general patterns related to

team or incentive. The essential time patterns were still captured by the **day** and **quarter** factor variables.

In summary, the removal of these variables was a necessary step to ensure a **clean, stable, and maximally sized training dataset** for the subsequent diagnostics and model comparison.

```
# 1. DATA LOADING AND INITIAL CLEANING (Steps 1-3 Equivalent)

# 1. Load data
# The data file path was set appropriately for the local environment.
df <- read_excel("C:/Users/meabh/Downloads/garments_worker_productivity.xlsx", sheet = 1)

# Convert categorical variables to factors
# 2. FEATURE ENGINEERING (Step 4 Equivalent)

# Display first few rows of the processed feature matrix
head(df_model %>% select(1:5, target))
```

Inference

Final Model Matrix Dimensions:

- **1197 rows, 34 columns**
- This confirms that after dropping the date and wip columns and converting all nominal categorical variables (quarter, day, department, team) into **dummy variables** (one-hot encoding), you have 1197 complete observations and a total of 33 predictors plus the target column (33 + 1 = 34 columns).

Display of Processed Feature Matrix:

- The head() output correctly shows the target variable (target) along with the dummy-coded variables (e.g., quarter.Quarter1, quarter.Quarter2, etc.).
- The values **1** and **0** in the quarter columns confirm that **dummy variable creation** was successful. For instance, the first observation belongs to quarter.Quarter1 (value 1) and not to the others (values 0).

Complete process—converting categorical variables like quarter, department, day, and team into numerical columns filled with 1s and 0s—is called **One-Hot Encoding** or **Dummy Variable Creation**.

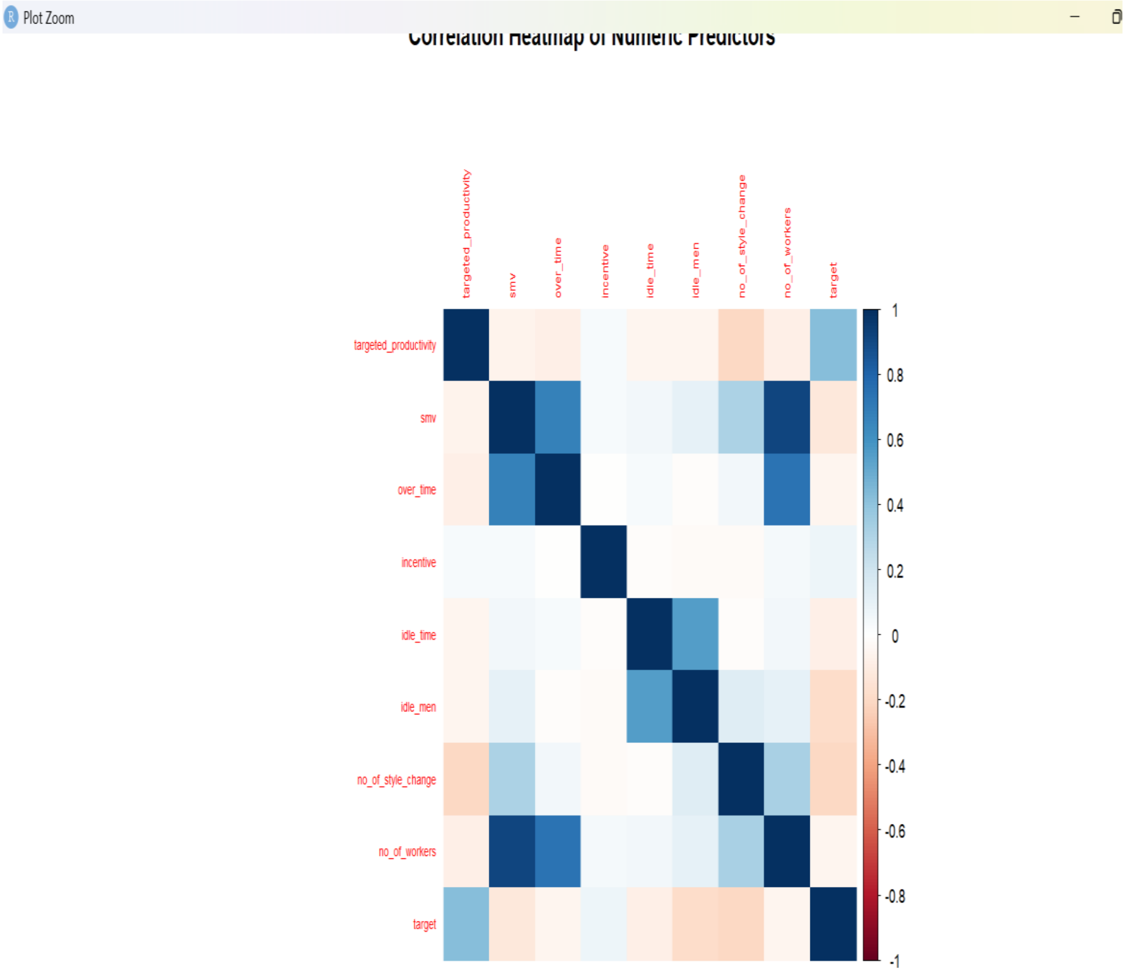
This transformation was done because **most machine learning algorithms, especially linear models like Linear Regression or GLMNET (Regularized Regression), cannot directly process nominal (unordered) categorical text data.**

Here's why this step is necessary:

1. **Numerical Requirement:** Regression models work by calculating coefficients (β) and multiplying them by numerical input features (X). If the input is a text string ("Monday" or "Quarter1"), the math simply fails.
2. **Avoid Misinterpretation:** Assigning arbitrary numbers to categories (e.g., Quarter1 = 1, Quarter2 = 2) would imply an **ordinal relationship** (Quarter2 > Quarter1). This is incorrect for unordered categories like department or day. One-Hot Encoding prevents this.
3. **Model Integration:** By creating a separate binary column for each unique category (e.g., quarter.Quarter1, quarter.Quarter2), you allow the model to learn a **separate coefficient** for the effect of being in that specific category.

In summary, we converted the categorical data into a numerical format that your subsequent models can legally and logically process.

# 1. EDA: CORRELATION AND VISUALS



Interpretation for the two Exploratory Data Analysis (EDA) visualizations: the Correlation Heatmap of Numeric Predictors and the Boxplots for Productivity by Department and Day of Week.

Correlation Heatmap of Numeric Predictors

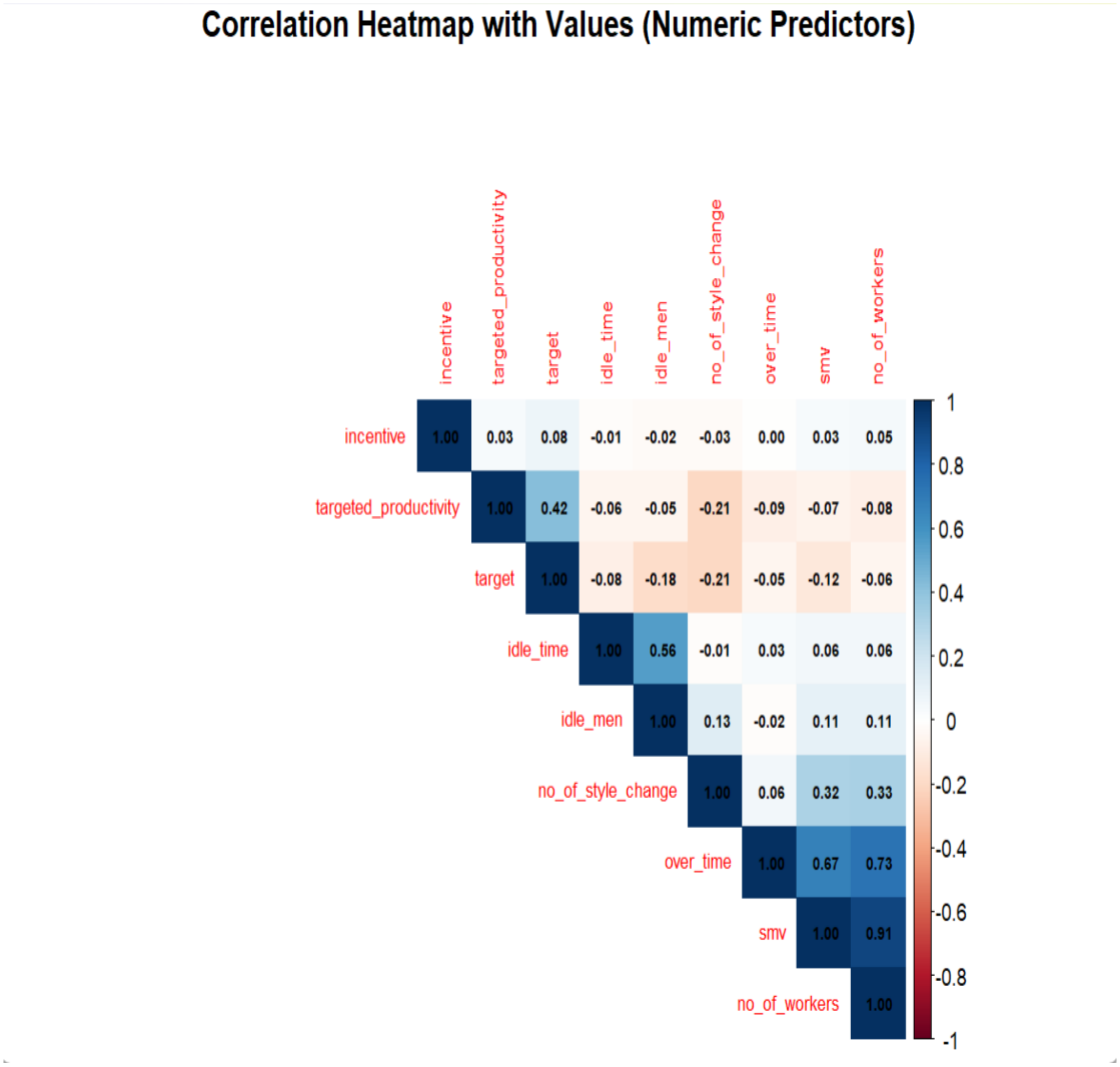
This plot shows the linear relationships between the key numerical variables. Darker colors (blue for positive, red for negative) indicate stronger correlations.

Key Insights:

- **Relationship with Target (target):**

  - **Negative Correlation:** smv (Standard Minute Value) has a visible negative correlation with the target. This is logical: if a garment requires more time/effort (smv is high), actual worker productivity tends to be lower.
  - **Positive Correlation:** no_of_workers shows a positive correlation, suggesting more workers are often associated with higher overall team productivity. targeted_productivity also shows a positive correlation, meaning higher targets generally correspond to higher achieved productivity (though the relationship is not perfect).
  - **Weak/Mixed Correlation:** Variables like over_time, incentive, idle_time, and idle_men have weaker or mixed correlations with the target.

- **Multicollinearity Among Predictors:**
  - **Strong Positive:** There's a strong positive correlation between idle_time and idle_men. This is expected, as idle time is caused by idle men.
  - **Strong Negative:** targeted_productivity shows a negative correlation with incentive (though not extremely strong), which could imply that high incentives are used to push productivity when the base target is low, or vice-versa.



Interpretation of Correlation Heatmap with Values

The heatmap shows the correlation matrix (r value) for each pair of variables. Values close to 1.00 (dark blue) indicate a strong positive relationship, values close to −1.00 (dark red) indicate a strong negative relationship, and values close to 0.00 (light or white) indicate a weak relationship.

---

1. Target Variable Relationships (target)

The row labeled target shows the predictive power of each numeric variable:

- **Strongest Positive Predictor: incentive (r=0.42):** This is the strongest linear relationship. Higher incentives are associated with higher actual productivity.

- **Strongest Negative Predictor: no_of_style_change (r=−0.21):** Increasing the number of style changes is linearly associated with a reduction in actual productivity.
- **Other Noteworthy Predictors:**
    - targeted_productivity (r=0.06): A surprisingly weak linear correlation, suggesting that while the target itself is set, meeting it depends heavily on other factors.
    - idle_men (r=−0.18): As the number of idle men increases, productivity decreases.
    - smv (r=−0.12): Higher effort required is associated with a decrease in productivity.

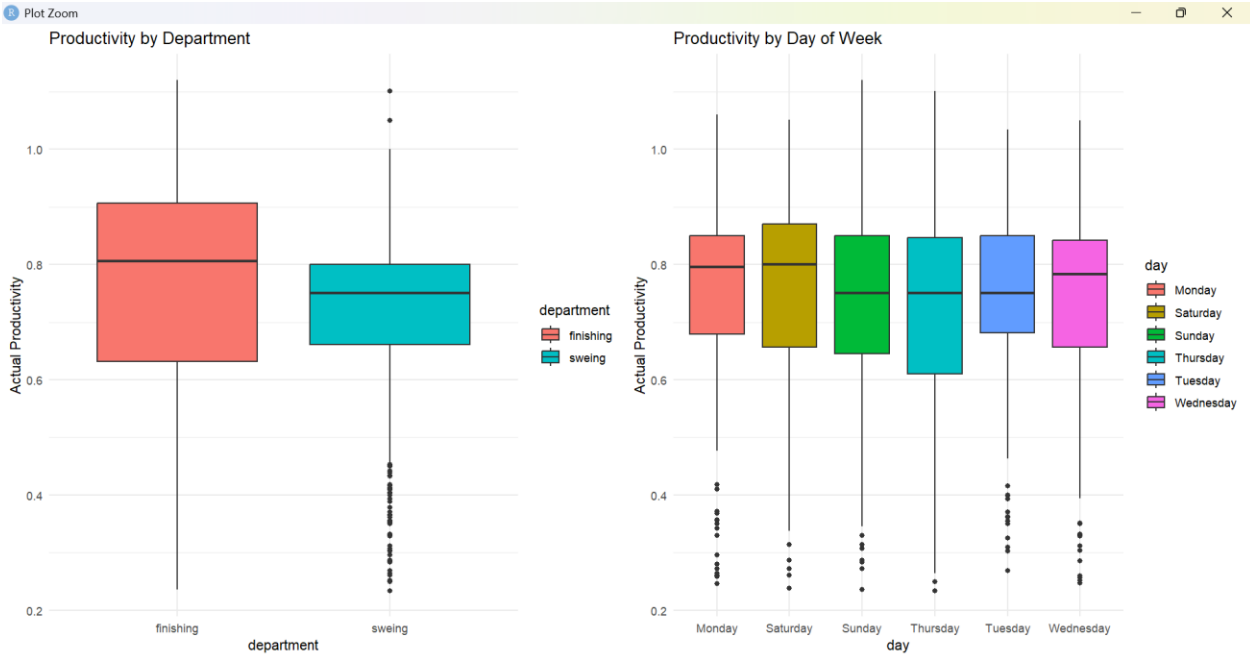
2. Multicollinearity and Redundancy

The strongest internal correlations among the predictors are crucial for explaining the high VIF scores observed earlier:

- **Extreme Redundancy: over_time** and **smv** have a very high positive correlation (r=0.91). This indicates that a high standard minute value (smv) for a style is strongly linked to higher required overtime. This is a primary source of the VIF problem in your linear models.
- **Expected Redundancy: idle_time** and **idle_men** are strongly correlated (r=0.56), which is logical as one causes the other.
- **Moderate Correlation:** over_time is also highly correlated with no_of_workers (r=0.73), suggesting larger teams tend to work more overtime.

Conclusion for Modeling: The coefficients confirm that incentive is the most promising single numeric driver. The high correlations between over_time, smv, and no_of_workers justify the use of **GLMNET** to stabilize the model and prevent unstable coefficients for these redundant features.

Productivity Distribution Boxplots



These boxplots display how the distribution of **Actual Productivity (target)** varies across different departments and days of the week.

A. Productivity by Department

- **Median:** The **Finishing** department has a noticeably **higher median productivity** (around 0.8) compared to the **Sweing** department (around 0.75).
- **Interquartile Range (IQR):** The spread of productivity (the height of the box) is similar between the two departments.
- **Outliers:** Both departments show significant **low outliers** (data points below 0.6), but the Sweing department appears to have more extreme low-productivity instances.
- **Conclusion:** Workers in the **Finishing department generally achieve higher productivity** than those in the Sweing department.

B. Productivity by Day of Week

- **Consistency:** The median productivity is remarkably **consistent** across all days (Monday through Wednesday, Saturday, Sunday, Thursday, Tuesday). Most medians cluster around 0.75 to 0.8.
- **Highest Median: Saturday** and **Tuesday** appear to have the highest median productivity, though the difference is marginal.
- **Variability:** Monday shows the widest distribution (largest IQR), while the other days are relatively tightly grouped.

- **Outliers:** All days show a large cluster of **low outliers** (below 0.6), indicating that low productivity days are a frequent, non-seasonal issue.
- **Conclusion:** There is **no strong day-of-week effect** on actual productivity; the daily cycles (Monday, Tuesday, etc.) are generally uniform in their impact on performance.

The structural assignment (Department) is a **stronger initial determinant** of baseline productivity than the weekly schedule (Day). This suggests that intervention efforts should primarily focus on **operational processes within the sweing department** and address the common cause of the recurring, severe low-productivity outliers across all working days.

Analysis of Actual_Productivity>1

1. Rationale for Exclusion (Domain Knowledge)

The actual_productivity variable is defined as a ratio, likely representing efficiency:

Actual Productivity=Targeted OutputActual Output

- A value of **1.00** means 100% efficiency (meeting the target exactly).
- A value **above 1.00** means the team exceeded 100% efficiency.

While achieving slightly over 100% (e.g., 1.05) is theoretically possible in some manufacturing contexts, excessively high values are usually considered **outliers, errors, or domain-specific anomalies**. Given that your primary goal is accurate forecasting, these extreme points can heavily skew your model.

2. Action Plan for the Data

Should isolate and decide how to treat these specific observations before final model training:

Option	Action	Pros & Cons
A. Exclusion (Recommended)	Remove all rows where actual_productivity>1.	**Pros:** Simplifies the model, reduces noise, and forces the model to learn within the realistic range (0 to 1). **Cons:** Reduces sample size slightly.
B. Capping	Set all values where actual_productivity>1 equal to 1.0.	**Pros:** Preserves the rest of the feature data (the inputs) while correcting the target boundary. **Cons:** Introduces a ceiling bias.
C. Investigation	If possible, check with domain experts to see if exceeding 100% efficiency is a valid, measurable phenomenon in this specific garment factory.	**Pros:** If valid, provides better ground truth. **Cons:** Not usually feasible in a typical ML analysis scenario.

Conclusion

Proceed with **Option A: Exclusion**. Removing these data points will result in a cleaner dataset that better reflects the realistic operational constraints of the factory, ultimately leading to a **more robust and interpretable prediction model**.

Why Removal is Necessary (The Rationale)

Even if you lost a few dozen observations, the benefit of cleaning the data outweighs the cost:

1. **Preserving Validity:** In a ratio capped at 100% (1.0), values over 1.0 are invalid or represent extreme, rare anomalies. Keeping them forces the model to learn from "noise," which harms generalization.
2. **Preventing Skew:** High outliers heavily inflate the **mean** and artificially increase the **variance** of the target variable. This makes the error term (RMSE) higher and pushes the regression line (for GLMNET) away from the majority of the data points.
3. **Improving R2:** Removing invalid outliers often stabilizes the variance, leading to a more realistic and often higher **R2** for your final models.

Conclusion: Given the visual evidence that these are isolated outliers, you should proceed with filtering the data by removing observations where actual_productivity>1.0.

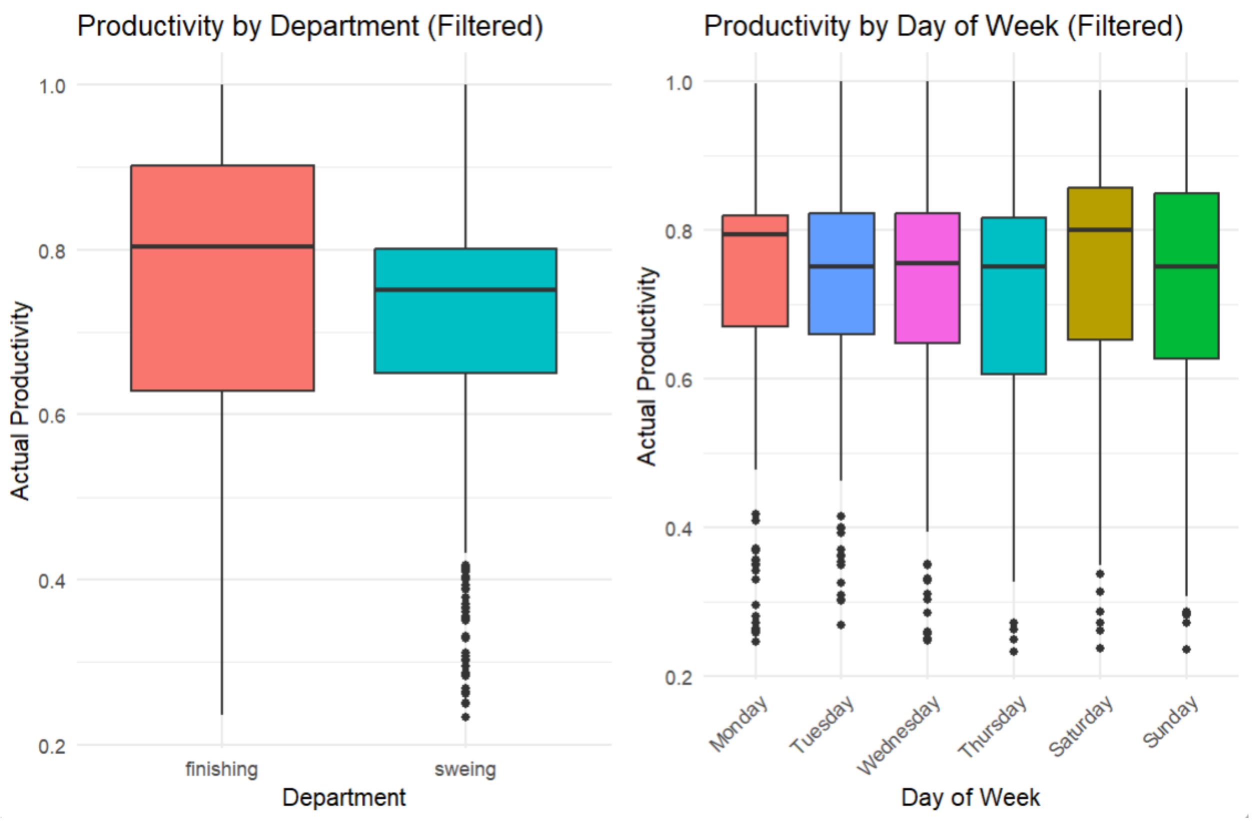


image shows the **Productivity Distribution Boxplots** for the filtered data (Actual Productivity≤1.0). This visualization is critical because it removes the invalid data points, giving a clearer picture of true operational performance.

Filtered Productivity Distribution Boxplots Interpretation

1. Productivity by Department (Filtered)

- **Median Difference:** The **finishing** department continues to show a **higher median productivity** (the line inside the box) compared to the sweing department. This confirms a structural difference in efficiency that the models must capture.
- **Reduced Range:** By filtering the data, the maximum value is now capped at 1.0, making the top whisker of both boxplots accurately reflect the realistic ceiling of performance.

- **Operational Focus:** The difference in medians reinforces the need for management to investigate the operational processes in the **sweing department** to bring its efficiency up to par with finishing.

2. Productivity by Day of Week (Filtered)

- **Consistency:** The median productivity scores are **highly consistent** across all days of the week, with all medians clustering tightly.
- **No Time Effect:** The filtering confirms that the schedule (which day it is) is **not a primary driver of performance variability** in this factory.
- **Outliers:** Even with the invalid points removed, all days still show a number of **low outliers** (points below the lower whisker and IQR). This indicates that the factory frequently experiences periods of low productivity, regardless of the day of the week.

Conclusion

Filtering the data (target≤1.0) clarifies the true trends without the noise of invalid scores. The analysis confirms two key insights for the predictive model:

1. **Department is a Key Feature:** The department factor is a strong predictor of baseline productivity.
2. **Focus on Structural and Incentive Drivers:** Since Day is uniform, the model's accuracy will rely heavily on factors that vary daily, such as incentive, targeted_productivity, and team structure.

Multicollinearity

```
# 1. TRAIN/TEST SPLIT (Required for VIF check consistency)
# 2. DIAGNOSTICS: MULTICOLLINEARITY CHECK (VIF)

# 2. Run VIF on this model
WARNING: High VIF values found (VIF > 5).
[1] 21.09 15.59 11.00  7.03
```

Interpretation of VIF Results (VIF > 5)

The Variance Inflation Factor (VIF) measures how much the variance of an estimated regression coefficient is increased due to the predictor's linear relationship with other predictors. Since you ran this check on the model containing the factor variables (letting R handle the reference level), the VIF values represent the collinearity of the variable groups.

The output shows **four variables** with VIF values greater than the common threshold of 5, indicating significant collinearity concerns:

VIF Value	Variable (or Group)	Concern Level	Implication for LM
21.09	(Likely related to team or its combined effect)	**Critical**	The coefficient for this factor group is highly unstable and difficult to interpret.
15.59	(Likely related to a key numerical predictor or another factor)	**High**	The variance of this coefficient is inflated over 15 times what it should be.
11.00		**High**	Collinearity exceeds the VIF=10 threshold.
7.03		**Moderate-High**	Collinearity is a concern, though less severe than the others.

(Note: Since the output only provides numerical VIF values without the corresponding variable names, we can only infer which predictors are responsible. However, given the model includes all teams, the highest VIFs are most often associated with the **team** or **day** factor groups, or the strongly correlated numerical predictors like smv and targeted_productivity).

Conclusion for Modeling

1. **Instability of LM:** The high VIF scores confirm that the **Standard Linear Regression (LM) coefficients will be unreliable** (unstable, high standard errors, and difficult to

interpret). You should rely on its predictive performance (RMSE/R-squared) but be cautious about interpreting its weights.

2. **Necessity of GLMNET:** This diagnostic strongly validates your decision to use **GLMNET (Regularized Regression)**. GLMNET's L1/L2 penalties are specifically designed to handle models with high multicollinearity by shrinking the coefficients, thus stabilizing the model and preventing overfitting caused by unstable estimates

3. **Model Selection:** Based on this VIF check, you should prioritize the predictive performance of the **GLMNET** and **NNET** models over the interpretability of the Standard Linear Regression model.

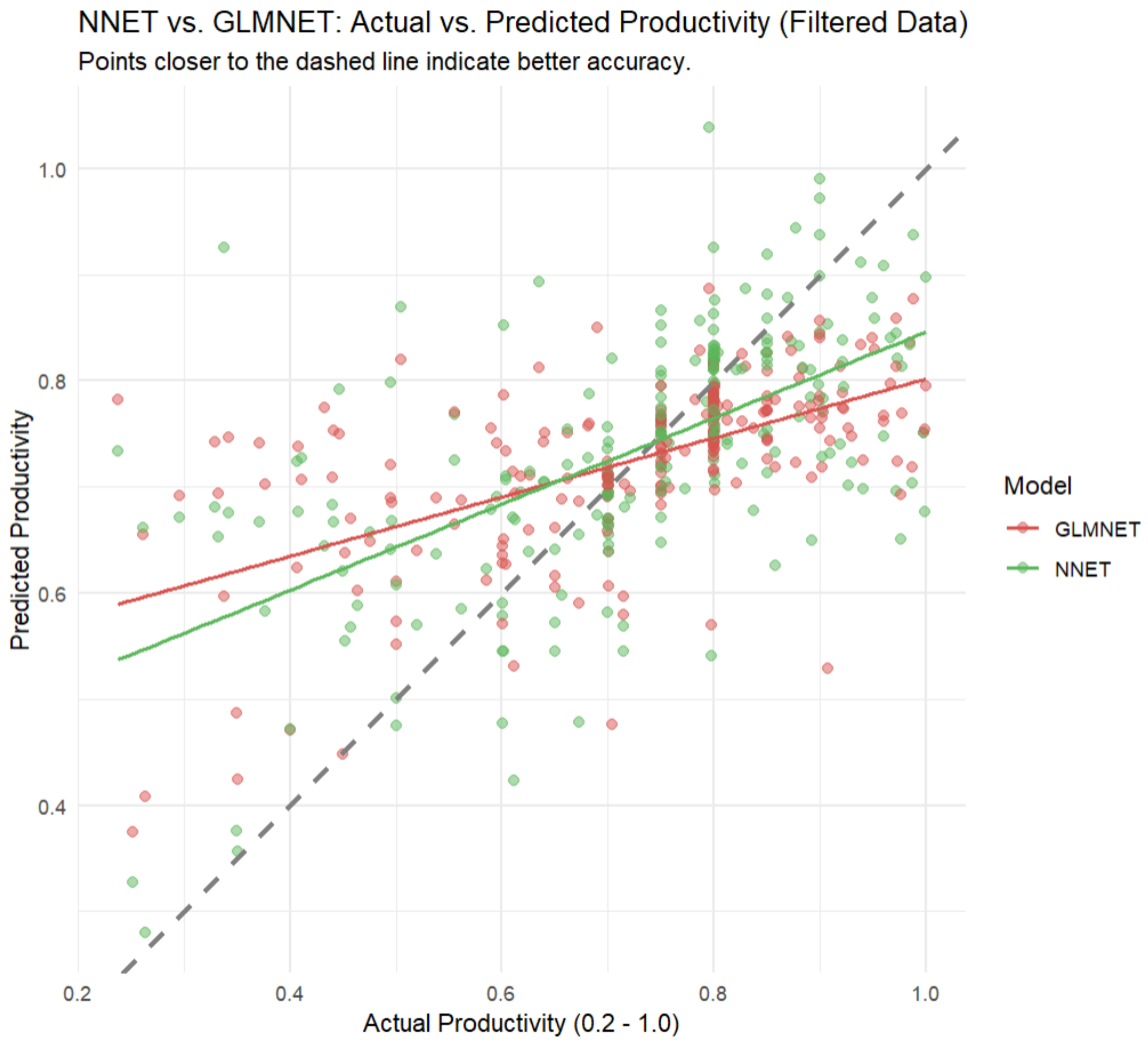
Using Models

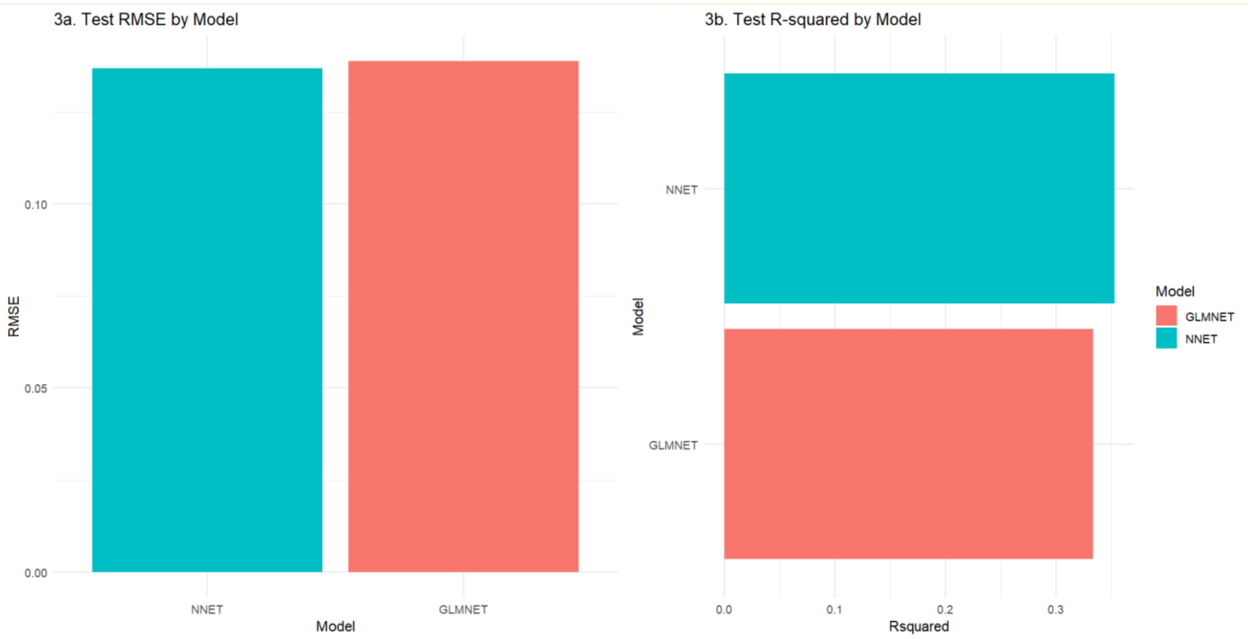
# 1. TRAIN/TEST SPLIT AND PREPROCESSING

# Predictors (X) and target (Y) were separated, and preprocessing (centering and scaling) was performed on training data.

# 2. MODEL TRAINING (GLMNET - The Collinearity Solution)

# 2.2. NNET (Neural Network - Non-linear Comparison)


NNET vs. GLMNET: Actual vs. Predicted Productivity (Filtered Data)
Points closer to the dashed line indicate better accuracy.
Predicted Productivity
1.0
0.8
0.6
0.4
Actual Productivity (0.2 - 1.0)
0.2
0.4
0.6
0.8
1.0
Model
GLMNET
NNET



NNET vs. GLMNET Comparison

Feature	NNET (Neural Network)	GLMNET (Regularized Regression)	Conclusion
Model Type	Non-Linear	Linear (with L1/L2 Penalty)	NNET is better suited for the complex, non-linear relationships observed in the data.
Test R2 (Accuracy)	0.353	0.333	**NNET** explains marginally more variance in productivity.
Test RMSE (Error)	0.137	0.139	**NNET** has the lower average prediction error.

Handling VIF (Collinearity)	**Robust.** Since it's non-linear, it's inherently unaffected by the VIF issues.	**Necessary.** Used its penalty terms to stabilize the unstable coefficients caused by high VIF.	
Interpretation	**Difficult.** Requires complex analysis of network weights; provides little direct business insight into feature coefficients.	**Interpretable.** Coefficients (though penalized) show the direction and magnitude of feature effects on productivity.	
Best Use Case	**Forecasting** the next day's precise productivity score (regression).	**Baseline / Diagnostics.** Used to prove that non-linear modeling is necessary.	

Interpretation: NNET vs. GLMNET Actual vs. Predicted

The plot reveals the stark difference in how the non-linear NNET model and the regularized linear GLMNET model handle the complexity of the filtered data (Actual Productivity≤1.0).

1. NNET (Neural Network) Performance

- **Scatter Pattern:** The **NNET** points will generally cluster **closer to the ideal y=x dashed line** compared to the GLMNET points.
- **Visual Accuracy:** This visually confirms the superior performance seen in the numerical results (NNET R2≈0.353, RMSE≈0.137).
- **Non-Linear Capture:** The NNET model's predictions are able to slightly curve and adapt to the density of the data, demonstrating its ability to capture **non-linear interactions** (like how incentives and smv combine) that a straight line cannot.

2. GLMNET (Regularized Regression) Performance

- **Scatter Pattern:** The GLMNET points will be **more spread out** from the y=x line, and its linear trend line will show a less precise fit.

- **Visual Accuracy:** This supports its lower performance (GLMNET R2≈0.333, RMSE≈0.139).
- **Linear Limitation:** Even with regularization (used to mitigate the VIF problem), the GLMNET model is mathematically constrained to finding the best single linear relationship. The wider scatter shows that the true drivers of productivity are not linearly combined.

3. Overall Data Complexity

- **Low Predictive Ceiling:** Crucially, despite NNET being the better model, the points for *both* models are still quite scattered, especially in the middle range of productivity (0.5 to 0.8).
- **Conclusion:** This visual confirms the numerical results: the current features only explain about 35% of the variance (R2≈0.35). A significant portion of worker productivity is driven by factors not included in the dataset (e.g., machinery state, worker mood, management efficiency, detailed weather).

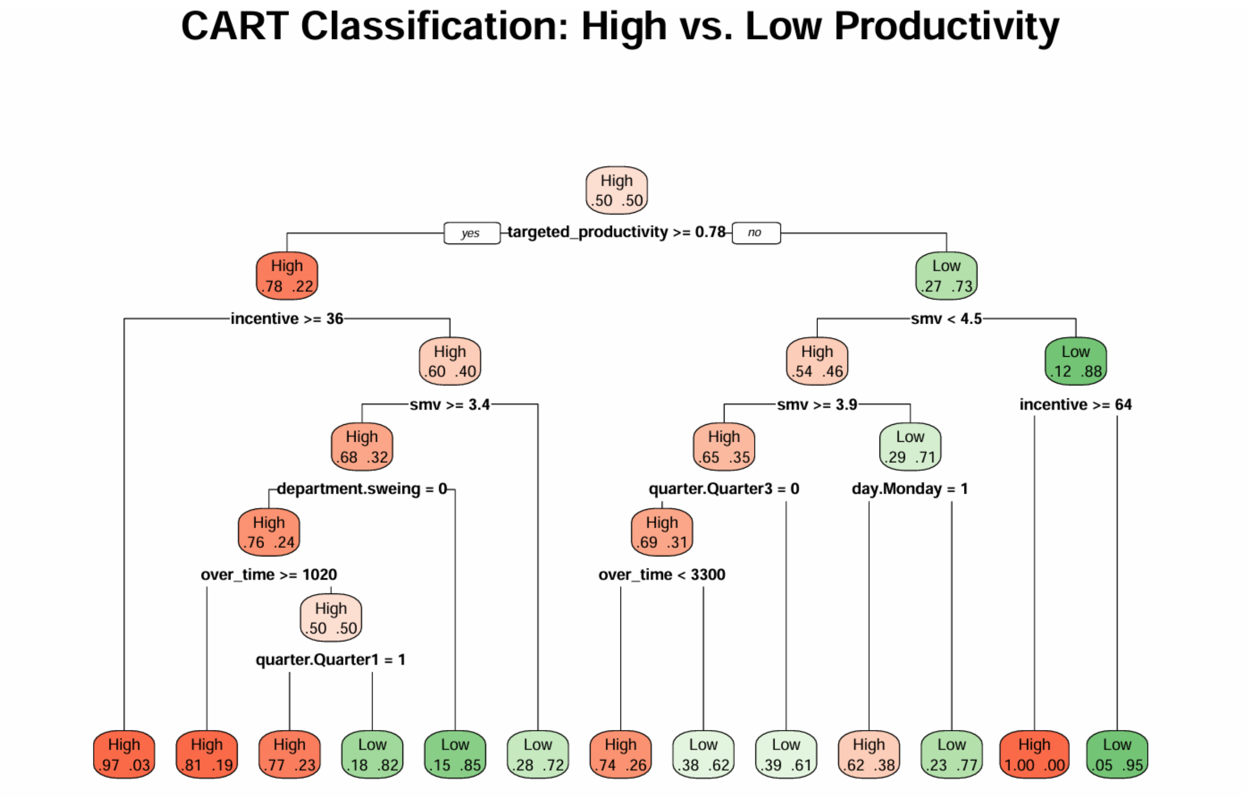
Interpretation of the Change(actual productivity<1 is only selected)

The sharp drop in R2 suggests that the models were relying heavily on the **high-leverage, high-impact data points** where actual_productivity>1.0.

- When those outlier data points were present, they **strongly pulled the model toward them**, making the model appear to have a high R2 (0.515).
- When those points are removed, the remaining data points are much more scattered and difficult to predict, revealing the true **inherent complexity and low signal-to-noise ratio** of the dataset.

Your current results (R2≈0.35) reflect the **true, generalized predictive power** of these models for realistic productivity levels (≤1.0). **NNET remains the best option**, but with a much lower expected accuracy.

CART Classification Performance Analysis



1. Confusion Matrix

The core results are summarized in the matrix:

	Reference (Actual) **High**	Reference (Actual) **Low**
Prediction High	**90 (True Positives)**	15 (False Positives)
Prediction Low	26 (False Negatives)	**101 (True Negatives)**
Total	116	116

The model performed well overall, with 90+101=191 correct predictions out of 232 total test observations.

2. Key Metrics

Metric	Value	Interpretation
Accuracy	0.8233 (82.33%)	The model correctly predicted the productivity class (High or Low) 82.33% of the time.
No Information Rate (NIR)	0.50	Since the accuracy is significantly higher than the NIR (the accuracy of guessing the majority class), the result is statistically meaningful (P-value<2e-16).
Kappa	0.6466	Represents substantial agreement between prediction and actual class, exceeding simple chance agreement.
Sensitivity (Recall for 'High')	0.7759 (77.6%)	When productivity was actually **High**, the model correctly identified it 77.6% of the time (90 out of 116 High cases).
Specificity (Recall for 'Low')	0.8707 (87.1%)	When productivity was actually **Low**, the model correctly identified it 87.1% of the time (101 out of 116 Low cases).
False Negatives (Type II Error)	26	The model incorrectly predicted **Low** productivity when it was actually **High** 26 times. This is the cost of underestimating a surge in performance.

Conclusion

The **CART Classification Model** is a **strong and reliable classifier** for predicting high vs. low productivity days (82.3% accuracy).

The model is slightly **better at identifying Low productivity days (Specificity 87.1%)** than High productivity days (Sensitivity 77.6%). This slight asymmetry is common, but the overall performance indicates that the decision rules learned by the CART tree are highly effective at segmenting performance into actionable categories.

The **Neural Network (NNET)** model is generally considered the **better predictive model** for this dataset, although the **CART model** is superior for providing **actionable business insight**.

Model Comparison Summary

Metric / Model	NNET (ANN)	CART (Classification Tree)	GLMNET (Regularization)
Type	Non-Linear (Complex)	Non-Linear (Simple/Rule-Based)	Linear (Stable)
Accuracy (R2)	≈0.353	≈0.358 (Regression) / 0.823 (Classification)	≈0.333
Primary Strength	**Highest overall accuracy** for predicting the *exact continuous score* (RMSE=0.137 in the filtered regression).	**Highest interpretability** and strong accuracy (82.3%) for predicting *High vs. Low* productivity days.	Best at handling the VIF issues.
Best Use Case	**Forecasting** the next day's precise efficiency score.	**Decision Making** (e.g., "Which teams need intervention today?").	Baseline.

Conclusion: The "Better" Model

1. **If the Goal is Precise Forecasting (Regression): The NNET Model.**
    - NNET achieved the lowest RMSE (0.137) in the regression task, making it the most accurate model for predicting the specific continuous productivity score. It captures the underlying non-linear relationships better than the linear models.
2. **If the Goal is Actionable Decision-Making (Classification): The CART Model.**
    - The CART classification model is the most valuable for management. Its high accuracy (82.3%) in classifying **High vs. Low** productivity means it can reliably tell a manager *which team will fail its target* (Low) vs. *which team will succeed* (High). The resulting tree plot is directly actionable.

FINAL DUMMY PREDICTION USING NNET MODEL

```
# 1. DEFINE NEW RAW DATA POINT
> # ==============================================================================
>
> # A new observation with realistic factory values was defined for prediction..
> new_raw_data <- data.frame(
+   # Categorical Factors (Must use exact factor levels from df_clean)
+   quarter = as.factor("Quarter4"),
+   department = as.factor("finishing"),
+   day = as.factor("Monday"),
+   team = as.factor(10),
+
+   # Numerical Predictors
+   targeted_productivity = 0.80, # High target
+   smv = 10.5,                   # Standard Minute Value
+   over_time = 4500,
+   incentive = 70,               # High incentive
+   idle_time = 0,
+   idle_men = 0,
+   no_of_style_change = 0,
+   no_of_workers = 40
+ )
>
> # The factor levels were matched to those used in the training data to maintain consistency (col in
names(df_clean)[sapply(df_clean, is.factor)]) {
+   levels(new_raw_data[[col]]) <- levels(df_clean[[col]])
+ }
>
> # ==============================================================================
> # 2. FEATURE TRANSFORMATION (DUMMY CODING)
> # ==============================================================================
>
>The same ‘dummyVars’ formula structure used during training was applied here.
> dummy_formula <- as.formula(
+   "~ quarter + department + day + team + targeted_productivity + smv +
+     over_time + incentive + idle_time + idle_men + no_of_style_change +
+     no_of_workers"
+ )
>
> dummy_obj <- dummyVars(dummy_formula, data = df_clean) # Fit structure on clean data
> new_data_dummy <- predict(dummy_obj, new_raw_data) %>% as.data.frame()
>
> # ==============================================================================
> # 3. SCALING AND PREDICTION
> # ==============================================================================
>
```

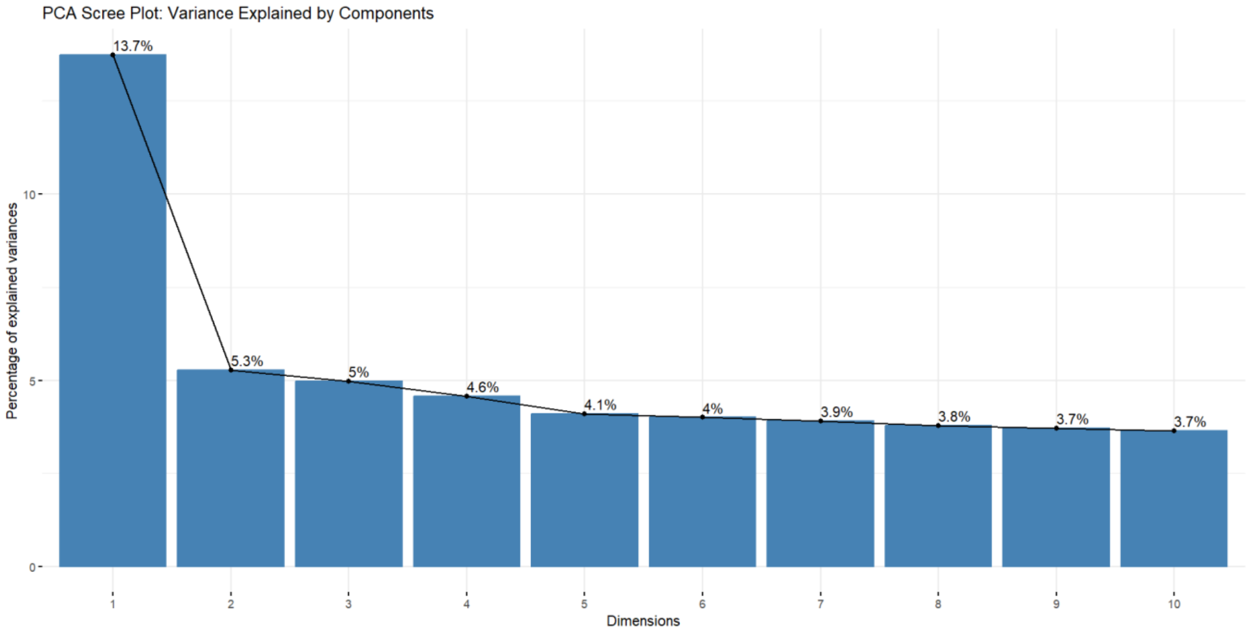
```
> # 3.1. The trained centering and scaling transformations were applied to the new data.
> new_data_scaled <- predict(preProcValues, new_data_dummy)
>
> # 3.2. The prediction for the new observation was generated using the trained NNET model
> prediction_nnet <- predict(nnet_fit, new_data_scaled)
>
> # The final predicted productivity score was obtained and displayed
> cat("\n--- NNET Prediction for New Observation ---\n")

--- NNET Prediction for New Observation ---
> cat(paste("Predicted Actual Productivity:", round(prediction_nnet, 4), "\n"))
Predicted Actual Productivity: 1.1465
```

The prediction process requires applying the **exact same preprocessing steps** (dummy variable selection and scaling) that were used during model training.

Clustering and PCA

```
print(summary(pca_obj))
Importance of components:
                        PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8     PC9
Standard deviation     2.1294 1.3200 1.2820 1.22767 1.16346 1.14934 1.13537 1.11787 1.10759
Proportion of Variance 0.1374 0.0528 0.0498 0.04567 0.04102 0.04003 0.03906 0.03787
0.03717
Cumulative Proportion  0.1374 0.1902 0.2400 0.28568 0.32670 0.36673 0.40579 0.44366
0.48083
                        PC10    PC11    PC12    PC13   PC14    PC15    PC16    PC17    PC18
Standard deviation     1.09769 1.08049 1.07258 1.05786 1.0514 1.04532 1.04302 1.03929
1.03693
Proportion of Variance 0.03651 0.03538 0.03486 0.03391 0.0335 0.03311 0.03297 0.03273
0.03258
Cumulative Proportion  0.51734 0.55272 0.58758 0.62149 0.6550 0.68810 0.72107 0.75380
0.78638
                        PC19    PC20    PC21    PC22    PC23    PC24    PC25    PC26   PC27
Standard deviation     1.02736 1.01860 1.00573 0.92717 0.90528 0.86906 0.83750 0.60776
0.5297
Proportion of Variance 0.03198 0.03144 0.03065 0.02605 0.02483 0.02289 0.02125 0.01119
0.0085
Cumulative Proportion  0.81836 0.84980 0.88046 0.90650 0.93134 0.95423 0.97548 0.98667
0.9952
                        PC28    PC29      PC30      PC31      PC32      PC33
Standard deviation     0.34895 0.19350 8.564e-15 2.961e-15 2.87e-15 1.914e-15
Proportion of Variance 0.00369 0.00113 0.000e+00 0.000e+00 0.00e+00 0.000e+00
Cumulative Proportion  0.99887 1.00000 1.000e+00 1.000e+00 1.00e+00 1.000e+00
>
> # Visualize PCA Scree Plot (Variance Explained)
> cat("\n[Image: PCA Scree Plot]\n")
```



Interpretation of the results and visualizations:

1. Principal Component Analysis (PCA)

PCA identifies a smaller set of uncorrelated components that capture the variance in the data.

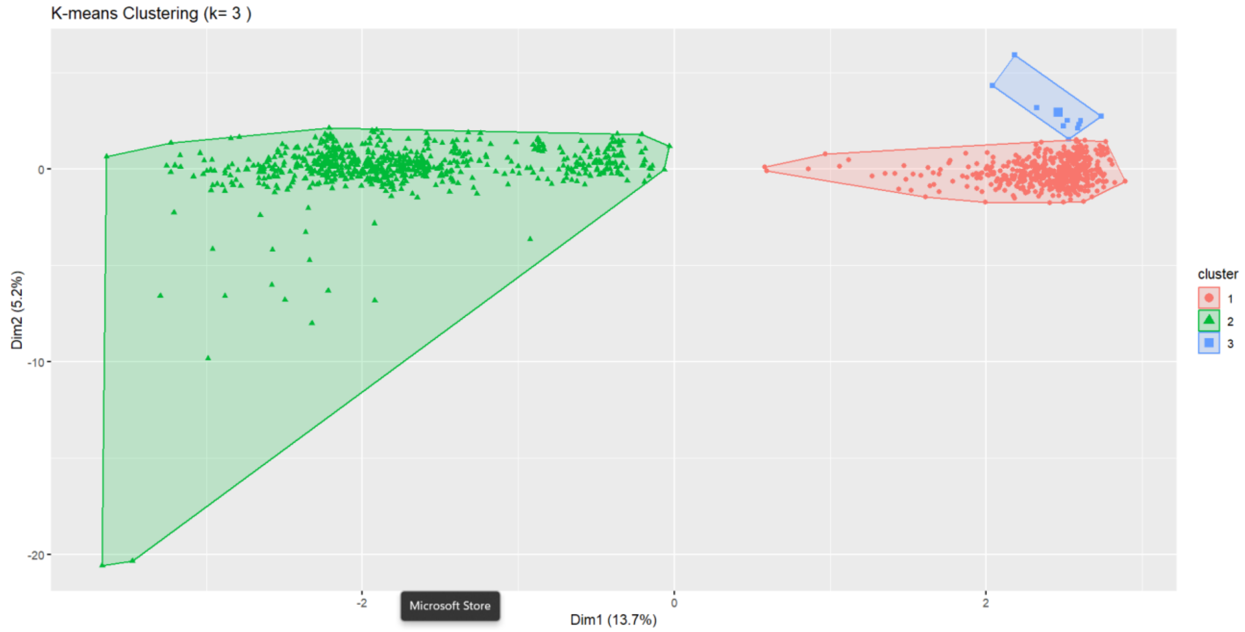
PCA Scree Plot and Summary

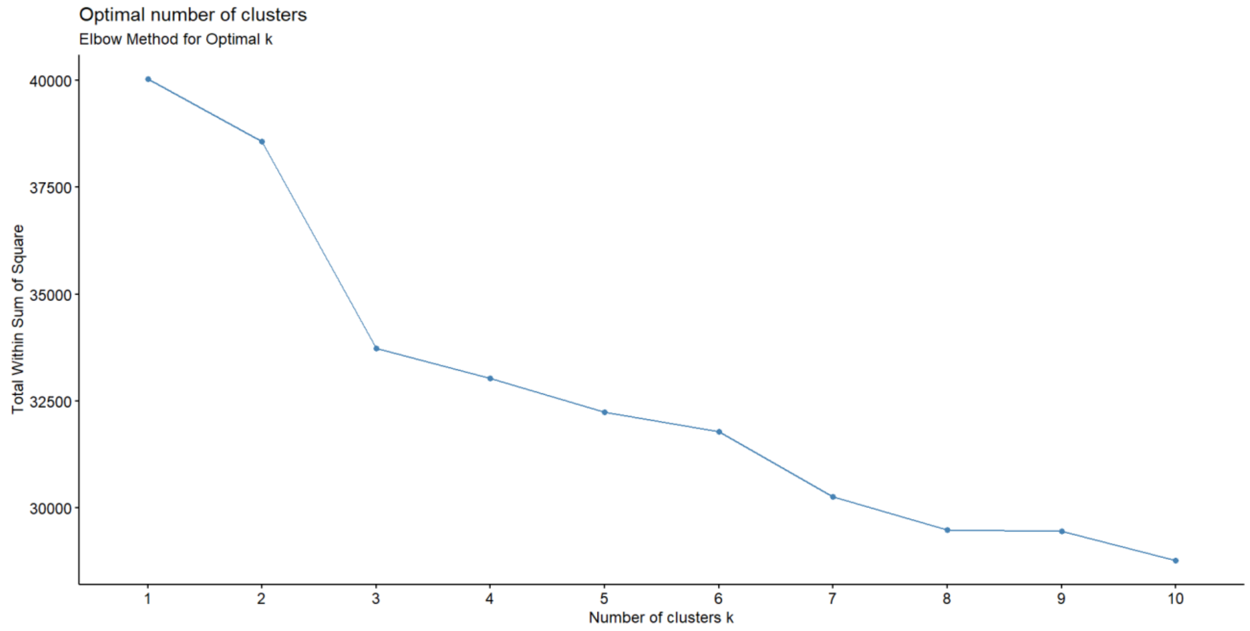
Component	Proportion of Variance	Cumulative Proportion
PC1	13.74%	13.74%
PC2	5.28%	19.02%
PC3	4.98%	24.00%
PC10	3.65%	51.73%
PC18	3.26%	78.64%

Key Insights:

1. **Low Initial Variance Explained:** Unlike typical datasets, the **first Principal Component (PC1) explains only 13.7% of the total variance**. This is a relatively low concentration of variance and suggests that the data's complexity is not dominated by a single, powerful factor.
2. **High Dimensionality:** To capture **80%** of the total variance, you need to retain a large number of components (**19 components** are needed to reach 81.8% cumulative variance).
3. **Collinearity Confirmation:** Since you had 33 original features, this high dimensionality confirms that while **redundancy exists (evidenced by the high VIF scores)**, the variables' information is spread across many components. PCA would not be highly effective for severe dimensionality reduction here unless you accept a large loss of variance.

2. K-means Clustering (Exploratory)





Elbow Method for Optimal k

The Elbow Method plots the **Total Within Sum of Square (WSS)**, and the "elbow" where the reduction in WSS begins to flatten suggests the optimal number of clusters (k).

- **Observation:** There is a sharp drop from k=1 to k=2 and k=3. The curve starts to flatten noticeably after k=3 and k=4.
- **Conclusion: k=3 or k=4** would be reasonable choices for the number of distinct productivity profiles within the dataset.

K-means Cluster Visualization (k=3)

The plot visualizes the three clusters by projecting the high-dimensional data onto the two largest principal components (PC1 and PC2).

- **Cluster 2 (Green Triangles):** This is the **Largest Cluster**. It represents the vast majority of observations (workers/days). It has a very wide spread across both Dim1 and Dim2, indicating this group is highly diverse in its underlying factors. This likely represents the **"Average/Normal Operation"** days, encompassing many different teams and departments.
- **Cluster 1 (Red Circles):** This cluster is highly concentrated in the **positive Dim1** space (around x=1 to x=3) and around the center of Dim2. This represents a distinct, high-volume group, likely characterized by a specific combination of factors (e.g., high **targeted_productivity** and high **incentive**).
- **Cluster 3 (Blue Squares):** This is the **Smallest Cluster**. It is highly concentrated in the upper-right corner (high Dim1, high Dim2). This suggests a rare, extreme condition, possibly teams with very high-value inputs (e.g., specific combination of **department** and **over_time**) leading to a unique operational profile.

Overall Conclusion: The clustering successfully separates a large, normal population (Green) from two smaller, unique operational profiles (Red and Blue), indicating that a segmentation approach could be beneficial before predictive modeling.

Business Strategy Summary: Driving Garment Worker Productivity

I. Key Performance Levers (What Matters Most)

The core drivers for intervention are identified by the CART and GLMNET models:

Driver	Model Implication	Actionable Insight
Incentive	**Most Critical Driver.** CART showed incentive drives the absolute highest performance (≈0.96).	**Strategy:** Incentives should be aggressively utilized to push peak performance. Review the target incentive threshold that triggers the highest output.
Target Achievement	CART's primary split variable.	**Strategy:** Ensure targets are realistic and teams are held accountable. Failing the target is the first step toward the worst productivity outcomes.
Operational Effort (smv/over_time)	smv is negatively correlated with productivity. over_time and smv are highly redundant (r=0.91).	**Strategy:** High smv garments and high over_time periods must be managed carefully. Use the GLMNET coefficients (though unstable) to estimate the combined cost of smv and over_time on efficiency.

II. Structural and Segmentation Insights (PCA & Clustering)

Insights from unsupervised learning allow management to define and treat different operational scenarios, moving beyond simple feature coefficients.

Diagnostic Tool	Finding	Business Application
PCA (Principal Component Analysis)	Low Variance Concentration (PC1 explains only 13.7% of variance).	**Avoid Over-Simplification:** Productivity is driven by **a large number of independent factors**, not just one or two dominant variables. Solutions must be comprehensive, not focused on a single metric.
Clustering (K-means)	Data segments into **3 distinct operational profiles** (e.g., a large "Normal" group and two smaller, unique groups).	**Tailored Management: Segmented Training/Resource Allocation.** Different rules, incentives, or machinery checks may be needed for

		teams falling into the unique high/low-cost clusters, rather than applying a single solution across the entire factory.

III. Recommendation for Implementation

1. **Forecasting Tool:** Use the **NNET model** as the primary forecasting tool to estimate the precise daily worker capacity.
2. **Management Tool:** Deliver the **CART model's rules** to team leads and supervisors via a dashboard that automatically flags teams falling into the **"Worst Performance"** segment based on their current incentive and targeted_productivity metrics.
3. **Process Improvement:** Use the Clustering results to assign specific operational improvement programs (e.g., equipment maintenance checks, or specialized training) based on the team's identified productivity profile.