# Garments Worker Productivity Analysis


Group 4

Aditya Chowdhuri

KPV Abhishek

Ankit

Khyati

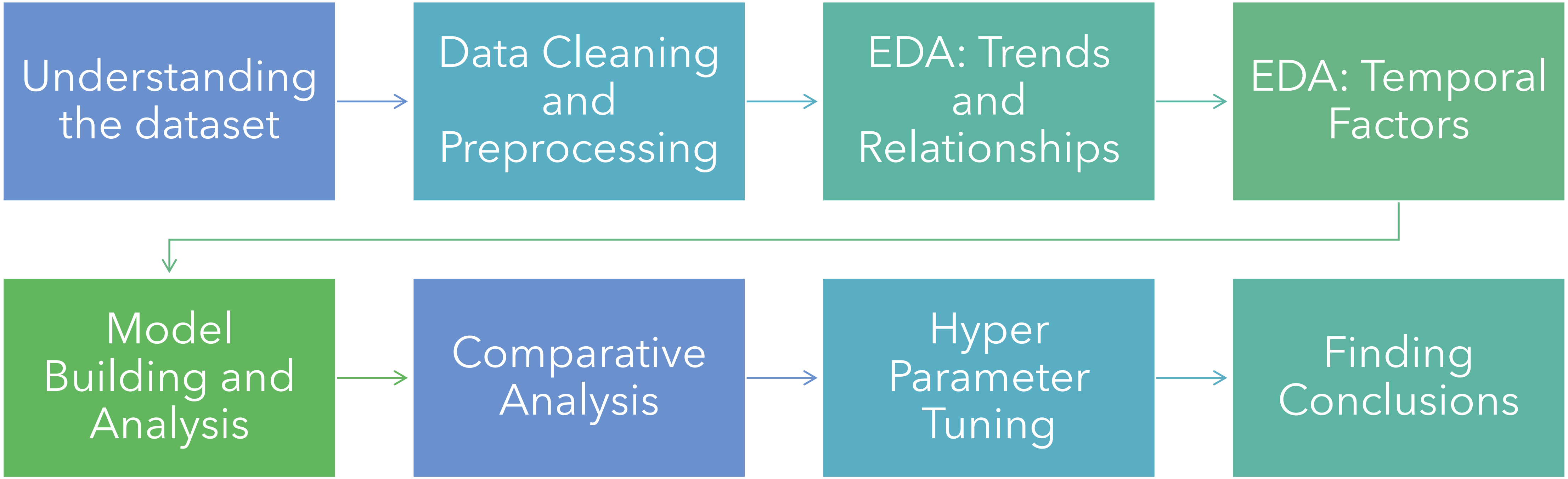
# Introduction to the Analysis

"OBJECTIVE: TO ANALYZE PRODUCTIVITY LEVELS IN THE GARMENT INDUSTRY."

"IMPORTANCE: UNDERSTANDING FACTORS INFLUENCING WORKER EFFICIENCY."

"IMPACT: INSIGHTS TO IMPROVE PRODUCTIVITY AND OPERATIONAL EFFICIENCY."

# Project Workflow



# Overview of the Dataset

Dataset Scope: "Data spans from January 1, 2015, to March 11, 2015, focusing on garment worker productivity."

Size and Structure: "Contains 1197 entries across 15 features, including date, department, team, and productivity metrics."

Key Variables Highlight:

"Targeted vs. Actual Productivity: Central to analyzing performance efficiency."

"Over Time and Incentive: Insights into workload and motivational factors."

"WIP (Work In Progress): Indicates operational workflow and capacity."

Data Integrity: "Most features are complete; however, 'wip' (work in progress) has 506 missing entries, handled in preprocessing."

Analytical Focus: "Emphasis on understanding factors that impact actual productivity, team performance, and departmental efficiency."

# Data Cleaning and Preprocessing

**Data Consistency:**

Ensured uniform data types and formats across all features for accurate analysis.

**Handling Missing Data:**

Focused on features like 'wip' (work in progress) with missing values.

**Encoding Categorical Data:**

Transformed categorical variables into a format that could be easily analyzed.

**Normalization:**

- Mean Normalization: Adjusted features to have a mean of 0 and a standard deviation of 1.
- Min-Max Normalization: Scaled features to a range between 0 and 1.

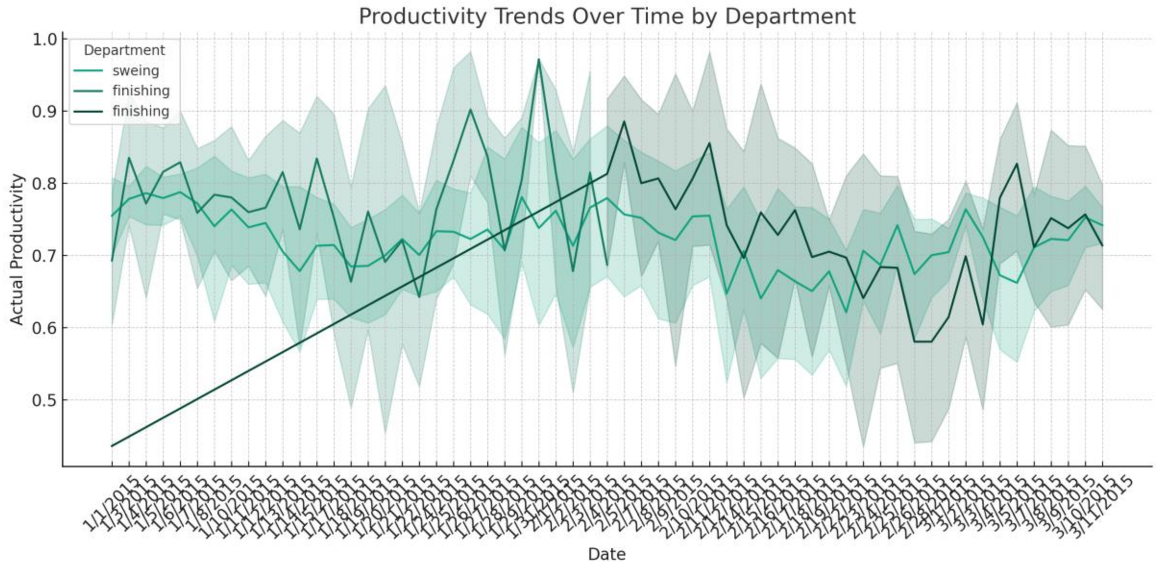
# Exploratory Data Analysis: Trends and Relationships

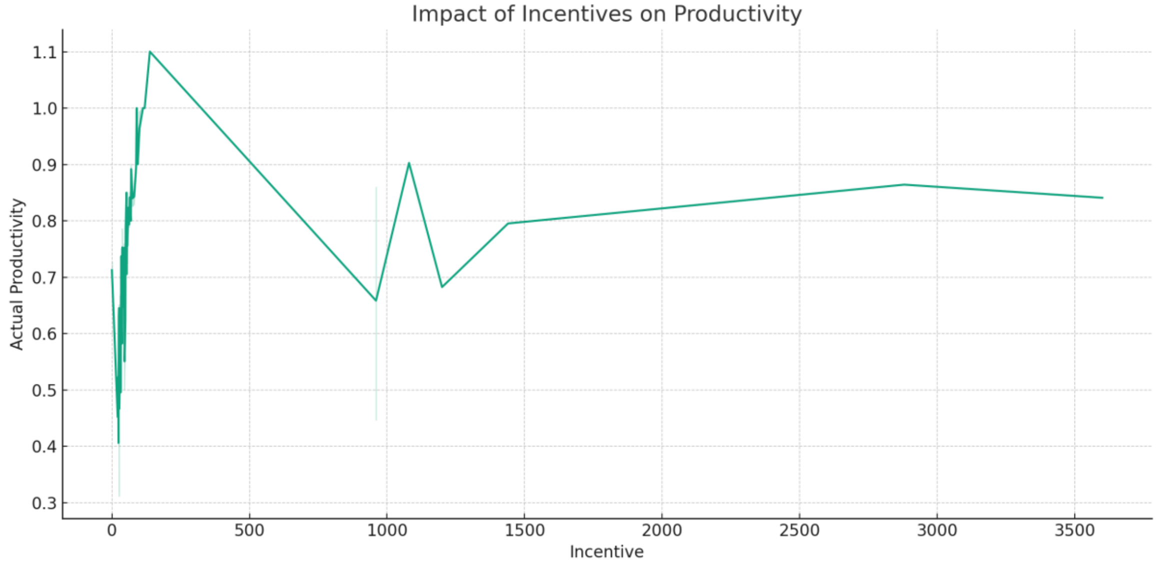
- **Graph 1: "Productivity Trends Over Time by Department"**

Insight: "Examined daily productivity variations across departments."

- **Graph 2: "Impact of Incentives on Productivity"**

Insight: "A clear upward trend indicates that departments generally experience increased productivity with higher incentives."



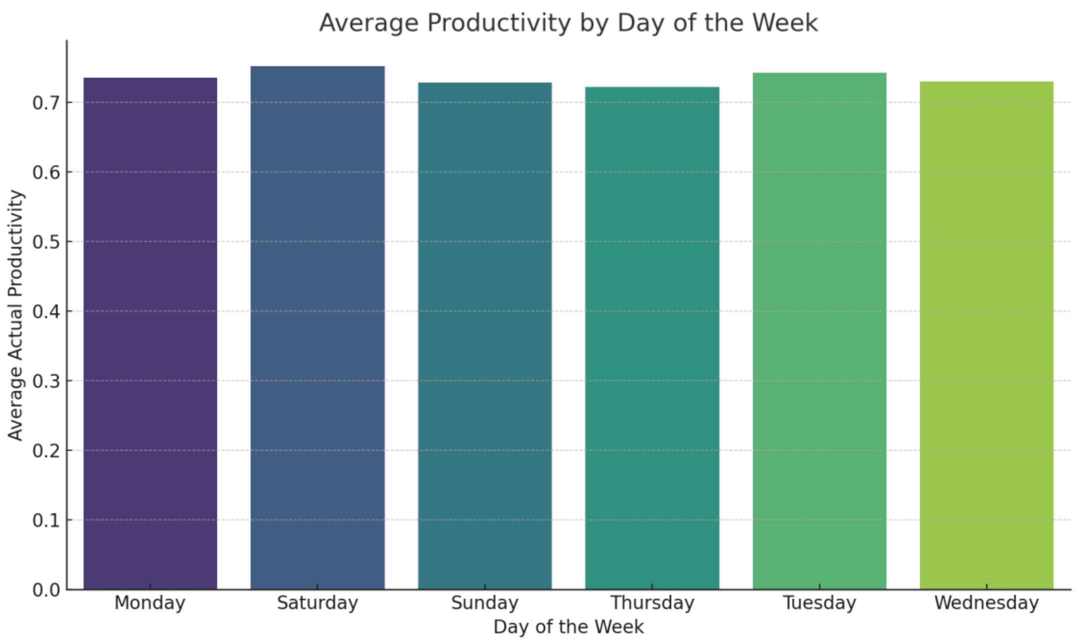
# Exploratory Data Analysis: Temporal Factors

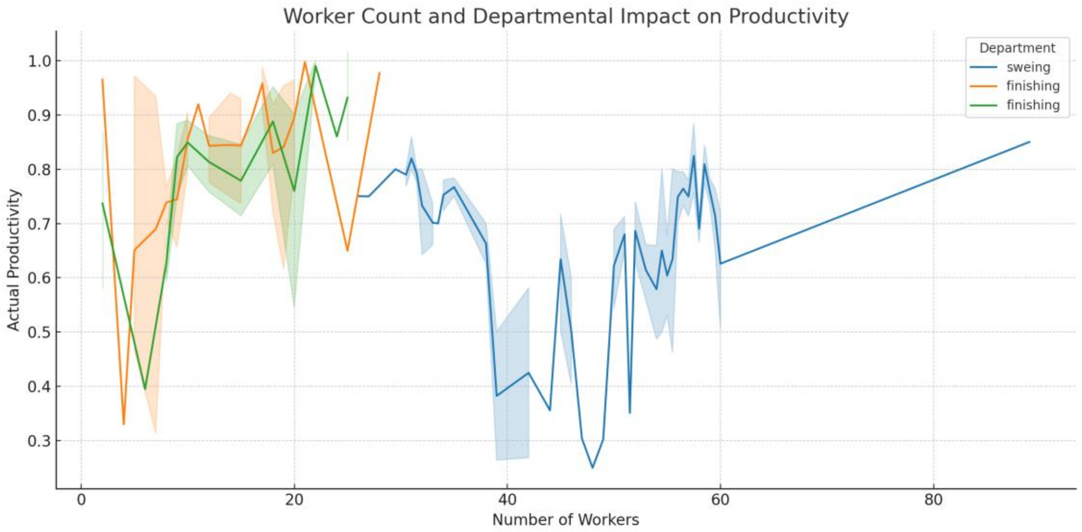
- **Graph 3: Productivity by Day of the Week**

Insight: "Productivity remains relatively consistent across all days, with no significant variations observed."

- **Graph 4: Worker Count and Departmental Impact**

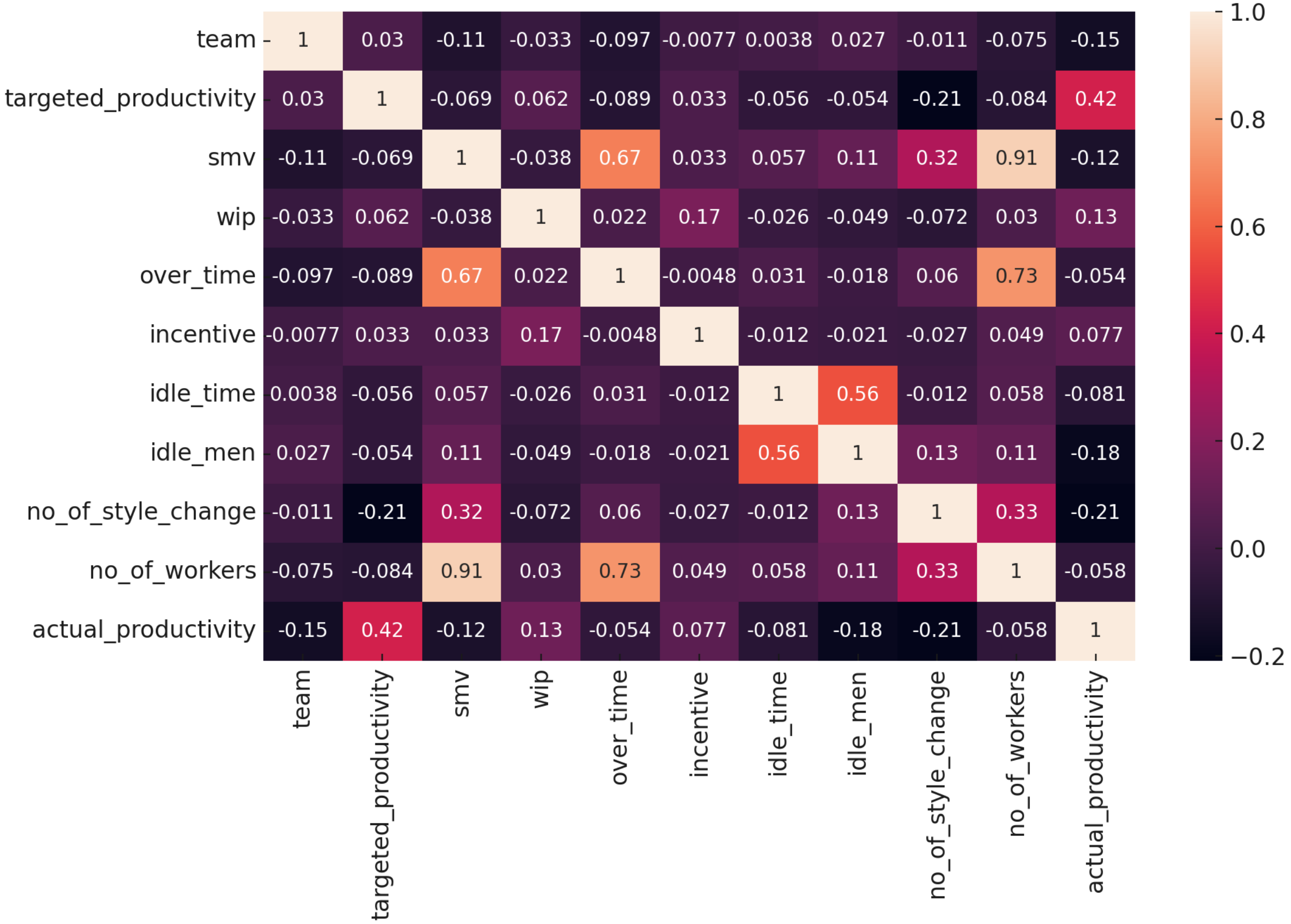
Insight: "Productivity generally increases with the number of workers up to a certain threshold, after which it plateaus."



# Visualization of Data Correlations

- **Worker Numbers and Overtime:** There is a significant positive correlation between no_of_workers and over_time. This suggests that as the workforce size increases, the amount of overtime tends to increase as well.
- **Productivity Influencers:** actual_productivity is positively correlated with incentive and no_of_workers, indicating that higher incentives and more workers generally lead to better productivity.
- Conversely, there's a negative correlation with idle_time and idle_men, highlighting that increased idle times are detrimental to productivity.
- **Target vs. Actual Productivity**: A positive but moderate correlation between targeted_productivity and actual_productivity suggests that while targets provide a guideline, achieving *actual productivity is influenced by multiple other factors.*



# Multi Collinearity Checks

- Variance inflation factor was used to check
- Columns with Multi Collinearity are:
  - Targeted Productivity
  - SMV
  - Number of workers
  - Department Sweing

| | VIF |
|---|---|
| team | 4.741576 |
| targeted_productivity | 15.592947 |
| smv | 18.059788 |
| over_time | 7.379474 |
| incentive | 1.131791 |
| idle_time | 1.504608 |
| idle_men | 1.570623 |
| no_of_style_change | 1.579334 |
| no_of_workers | 48.667939 |
| quarter_Quarter2 | 1.949833 |
| quarter_Quarter3 | 1.617029 |
| quarter_Quarter4 | 1.800280 |
| quarter_Quarter5 | 1.240162 |
| department_finishing | 2.083363 |
| department_sweing | 23.614834 |
| day_Saturday | 2.025713 |
| day_Sunday | 2.003934 |
| day_Thursday | 2.042090 |
| day_Tuesday | 1.966888 |
| day_Wednesday | 2.006785 |

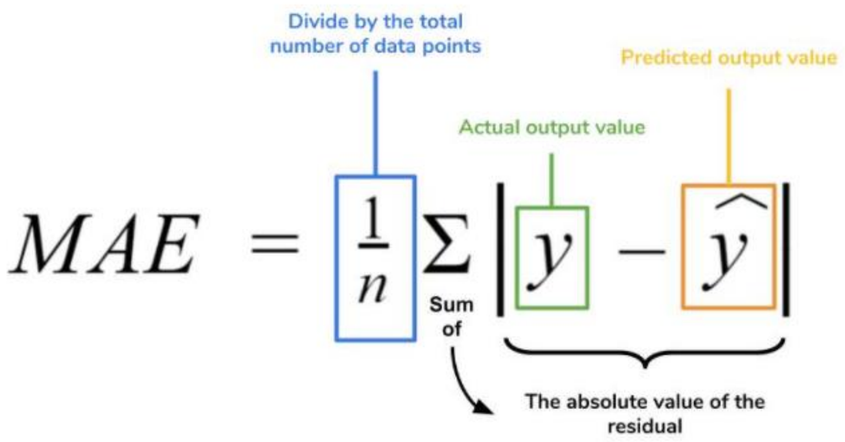
# Model Building and Analysis

- Decision Tree
- Random Forest
- Linear Regression
- Ridge Regression
- K-Nearest Neighbors
- Support Vector Regression
- XGBoost

# Choosing Evaluation Metrics

## MAE was chosen as the metric of choice for the following reasons

- **Simple and Intuitive:** MAE provides a straightforward measure of average error in the same units as productivity, making it easy for stakeholders to understand.
- **Robust to Outliers:** Unlike squared error metrics, MAE is less sensitive to rare but extreme productivity deviations, ensuring a more balanced model evaluation.
- **Directly Relevant:** MAE's direct error averaging aligns well with the dataset's continuous productivity values, offering a clear perspective on model accuracy.
- **Consistent Scale:** MAE matches the scale of the target variable, allowing for a clear, direct interpretation of the prediction error magnitude.



# Decision Tree Regression

| Train Size (%) | Test Size (%) | Mean Absolute Error |
| ---: | ---: | ---: |
| 90 | 10 | 0.08844 |
| **85** | **15** | **0.07699** |
| 80 | 20 | 0.09396 |
| 75 | 25 | 0.10454 |
| 70 | 30 | 0.09589 |
| 65 | 35 | 0.10331 |
| 60 | 40 | 0.10293 |
| 55 | 45 | 0.09753 |
| 50 | 50 | 0.09710 |

# Random Forest Regression

| Train Size (%) | Test Size (%) | Mean Absolute Error |
|---:|---:|---:|
| **90** | **10** | **0.065430075** |
| 85 | 15 | 0.070192716 |
| 80 | 20 | 0.073252402 |
| 75 | 25 | 0.079894848 |
| 70 | 30 | 0.078871584 |
| 65 | 35 | 0.082222543 |
| 60 | 40 | 0.08213174 |
| 55 | 45 | 0.07741973 |
| 50 | 50 | 0.079720101 |

# Linear Regression

| Train Size (%) | Test Size (%) | Mean Absolute Error |
|---|---|---|
| **90** | **10** | **0.105937861** |
| 85 | 15 | 0.106423333 |
| 80 | 20 | 0.108614417 |
| 75 | 25 | 0.111214394 |
| 70 | 30 | 0.110823908 |
| 65 | 35 | 0.128172574 |
| 60 | 40 | 0.127065411 |
| 55 | 45 | 0.126445408 |
| 50 | 50 | 0.127547967 |

# Support Vector Regression

| Train Size (%) | Test Size (%) | Mean Absolute Error |
|---:|---:|---:|
| **90** | **10** | **0.116384593** |
| 85 | 15 | 0.117117239 |
| 80 | 20 | 0.122901735 |
| 75 | 25 | 0.123751111 |
| 70 | 30 | 0.127805978 |
| 65 | 35 | 0.131143129 |
| 60 | 40 | 0.128962518 |
| 55 | 45 | 0.128996364 |
| 50 | 50 | 0.12866834 |

# XGBoost Regression

| Train Size (%) | Test Size (%) | Mean Absolute Error |
|---|---|---|
| **90** | **10** | **0.072753529** |
| 85 | 15 | 0.074613518 |
| 80 | 20 | 0.078751224 |
| 75 | 25 | 0.087170226 |
| 70 | 30 | 0.081963369 |
| 65 | 35 | 0.082820627 |
| 60 | 40 | 0.083250973 |
| 55 | 45 | 0.08313448 |
| 50 | 50 | 0.085131545 |

# KNN Regression

| Train Size (%) | Test Size (%) | Mean Absolute Error |
|---:|---:|---:|
| **90** | **10** | **0.105660144** |
| 85 | 15 | 0.107386416 |
| 80 | 20 | 0.108698153 |
| 75 | 25 | 0.111652415 |
| 70 | 30 | 0.112255811 |
| 65 | 35 | 0.116633859 |
| 60 | 40 | 0.114772785 |
| 55 | 45 | 0.112367598 |
| 50 | 50 | 0.112973774 |

# Ridge Regression

| Train Size (%) | Test Size (%) | Mean Absolute Error |
|---:|---:|---:|
| **90** | **10** | **0.106521157** |
| 85 | 15 | 0.107147829 |
| 80 | 20 | 0.109317151 |
| 75 | 25 | 0.112086198 |
| 70 | 30 | 0.111510312 |
| 65 | 35 | 0.128768105 |
| 60 | 40 | 0.127660796 |
| 55 | 45 | 0.126813299 |
| 50 | 50 | 0.127837744 |

# Comparative Analysis of Model Performance

| Model | Mean Absolute Error (MAE) | Train-Test Split |
|---|---|---|
| Decision Tree | 0.076991 | 85:15 |
| **Random Forest** | **0.065430** | **90:10** |
| Linear Regression | 0.105660 | 90:10 |
| Support Vector Regression | 0.116385 | 90:10 |
| Gradient Boosting (XGBoost) | 0.072754 | 90:10 |
| KNN Regression | 0.105660 | 90:10 |
| Ridge Regression | 0.106521 | 90:10 |

# Hyperparameter tuning and Grid Search

**What is Hyperparameter Tuning?**

Optimizing the parameters of a machine learning model that are not learned from the data but set before the training process.

It's crucial for enhancing model performance by finding the most effective combination of parameters.

**Focus on Efficient Models:**

Applied to Random Forest and XGBoost, the two models that showed promising performance in initial evaluations.

Aim to further refine these models to achieve the best possible predictive accuracy.

**Grid Search Explained:**

Involves defining a 'grid' of various hyperparameter values and exhaustively searching through them.

The best combination is selected based on performance metrics, ensuring optimal model settings.

**Outcome Expectation:**

Anticipate improved accuracy and robustness in the selected models.

Hyperparameter tuning allows for a more tailored and precise modeling approach

# Hyperparameter Tuning of Random Forest

**Purpose:**

- Enhance Random Forest model performance by optimizing its parameters.

**Tuning Parameters:**

- Focused on 'n_estimators', 'max_depth', 'min_samples_split', 'min_samples_leaf'.
- Explored combinations to find the best fit for the dataset.

**Method:**

- Employed Grid Search with cross-validation.
- Ensured thorough exploration and robust validation of parameter sets.

**Results:**

- Determined the optimal configuration for Random Forest.
- Noted performance improvement over default settings.

# Hyperparameter Tuning of XGBoost

**Objective:**

- Optimize XGBoost parameters for superior predictive accuracy.

**Tuning Focus:**

- Tuned n_estimators, learning_rate, max_depth, colsample_bytree.
- Sought the best balance of speed and accuracy.

**Approach:**

- Utilized Grid Search with cross-validation strategy.
- Comprehensive parameter exploration for model refinement.

**Outcomes:**

- Identified the most effective parameter combination for XGBoost.
- Enhanced model's predictive performance post-tuning.

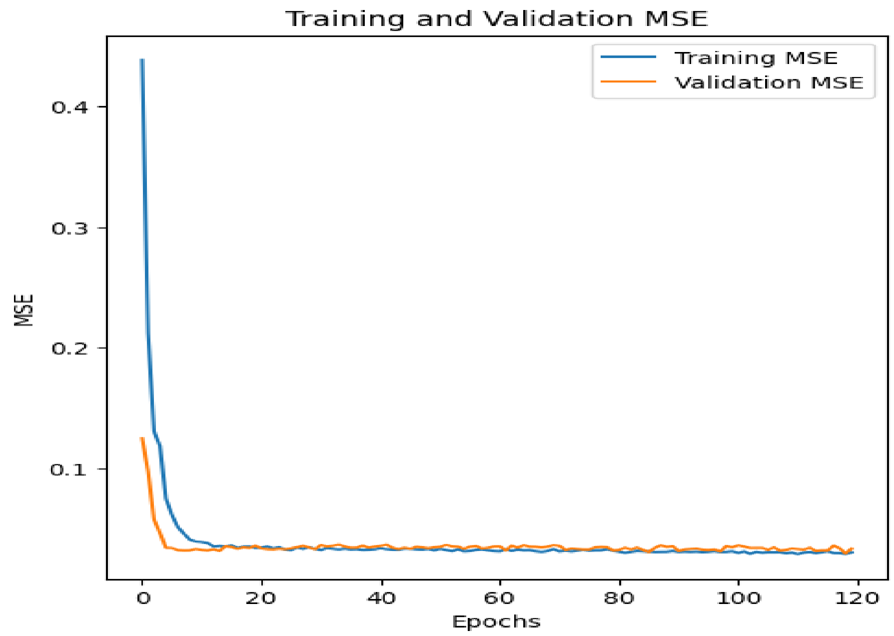
# Results of Hyperparameter tuning

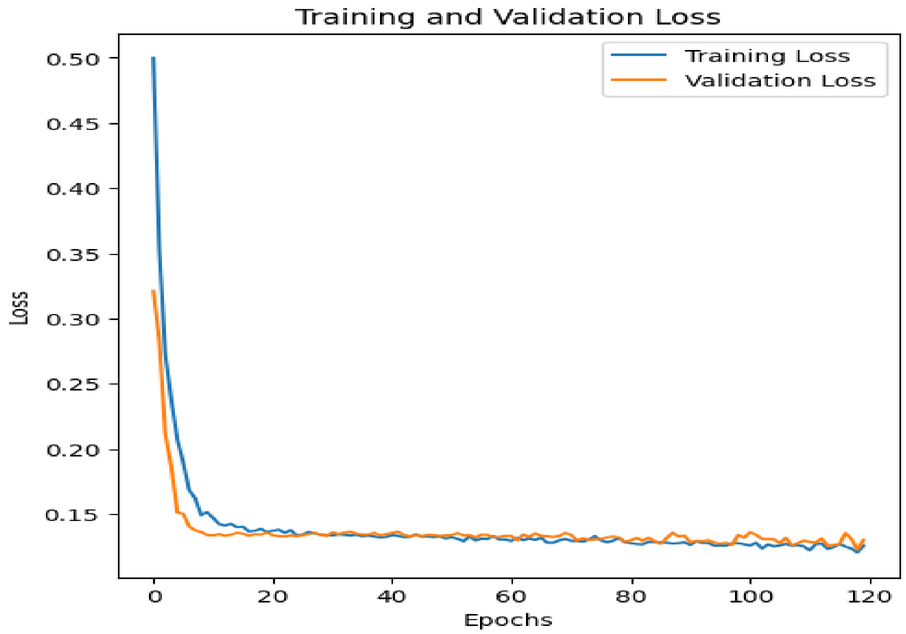
The best-fit results for Random Forest and XGBoost are:

| Model | Mean Absolute Error (MAE) |
|---|---|
| **Random Forest** | **0.0796** |
| **Gradient Boosting (XGBoost)** | **0.0788** |

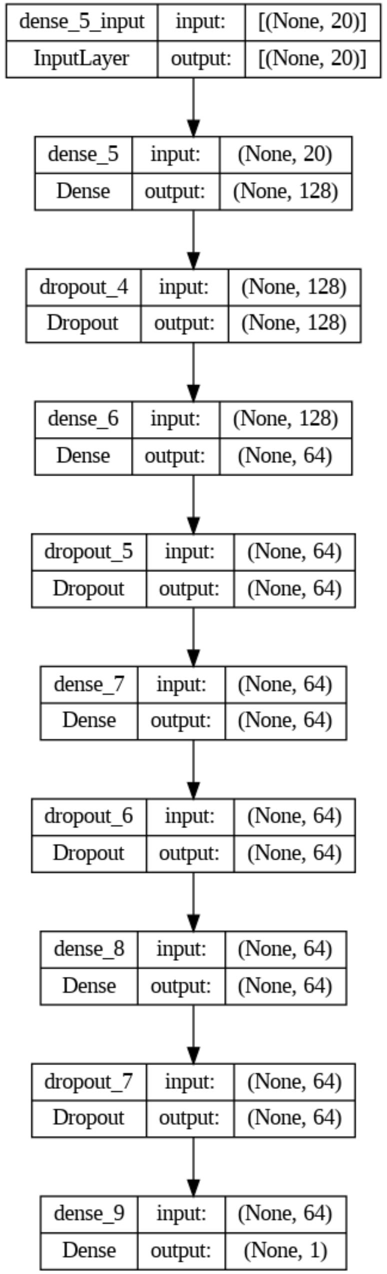
# Neural Network Regression

- A deep neural network was trained over 120 epochs to predict Productivity
- Parameters
  - Optimizer: Stochastic Gradient Descent
  - Loss function: Binary Mean Absolute Error
  - Trainable Parameters: 19329
- **Loss: 0.1243**
- **MSE: 0.0292**



# Model Architecture



# Key Findings

**Productivity Insights:**

Key factors impacting productivity identified: incentives, number of workers, and idle time.

Overtime and worker count are closely related, affecting overall productivity.

**Model Performance:**

Random Forest model showed promising initial results.

Hyperparameter tuning yielded marginal improvements due to the already high efficiency of the base models.

**Data-Driven Decision Making:**

The analysis provides actionable insights for optimizing workforce management and setting realistic productivity targets.

# Future Recommendations and Limitations

### Recommendations for Improvement

Invest in enhancing computational resources for more nuanced model tuning and potentially better predictive accuracy.

Explore additional data points or features that might further elucidate productivity dynamics.

### Project Limitations

Computational constraints limited the extent of hyperparameter tuning.

The current analysis focuses on existing features; additional data could reveal more insights.

### Next Steps

Implement findings in workforce management strategies.

Consider a longitudinal study to track the effectiveness of implemented changes over time.

# Thank you

# Appendix

# Outcomes and Computational Constraints

**Tuning Results:** Random Forest and XGBoost tuning yielded MAE close to the base models, indicating initial efficiency.

**Efficiency vs. Resources:** The base models' low MAE suggests high initial optimization.

**Project Scope:** Further tuning is constrained by the project's computational limits.

# Link to Collab Notebook