GARMENT WORKERS PRODUCTIVITY PREDICTION

1. Introduction

1.1. Project overviews

Predicting garment workers' productivity is essential for optimizing operations in the textile industry. Factors such as skill levels, work environment, and management practices influence productivity. Methods like historical data analysis, time studies, and advanced technologies (IoT, AI) are used for prediction. Challenges include variability and ensuring ethical considerations. Future trends focus on integrating technology for better accuracy and balancing productivity with worker well-being.

1.2. Objectives

Predicting garment workers' productivity serves several important objectives, which are crucial for both management and workers themselves:

1. **Resource Allocation**: By predicting productivity, managers can better allocate resources such as materials, equipment, and manpower. This ensures that the right amount of resources are available to meet production targets efficiently.
2. **Production Planning**: Accurate productivity predictions help in planning production schedules and timelines. This allows managers to set realistic deadlines, manage workloads effectively, and reduce the risk of delays.
3. **Performance Management**: Predictions enable the establishment of benchmarks for individual workers and teams. This facilitates performance evaluation, incentive programs, and identifying areas where additional training or support may be needed.
4. **Cost Efficiency**: Understanding productivity patterns helps in optimizing costs associated with labor, energy consumption, and raw materials. By improving productivity, businesses can achieve higher output with the same or fewer resources.
5. **Quality Control**: Productivity predictions indirectly impact product quality by ensuring that workers are not rushed, thereby reducing errors and defects. Consistent productivity levels often correlate with consistent quality standards.
6. **Employee Motivation and Satisfaction**: Clear productivity expectations and achievable goals can boost worker morale and job satisfaction. Predictions can be used to set fair targets and provide feedback that motivates employees to perform better.
7. **Risk Management**: By anticipating fluctuations in productivity, managers can proactively address potential risks such as production bottlenecks, staffing shortages, or equipment failures.
8. **Forecasting and Decision Making**: Predictions contribute to overall business forecasting and decision-making processes. This includes inventory management, sales forecasting, and strategic planning based on expected production outputs.

In essence, garment workers' productivity predictions are essential for optimizing operations, enhancing overall efficiency, and ensuring sustainable growth in the garment manufacturing industry.

2. Project Initialization and Planning Phase

During the project initialization and planning phase, the key objectives are to define project goals, allocate resources effectively, assess and mitigate risks, establish timelines and milestones, identify stakeholders and communicate with them, plan the budget, define team roles, document processes, seek necessary approvals, and finalize a comprehensive project plan.

2.1 Define Problem Statement

In the garment manufacturing industry, accurately predicting the productivity of individual workers on the production floor remains a significant challenge. Current methods for assessing productivity often rely on manual observation and subjective evaluation, leading to inconsistencies and unreliable forecasts. This lack of accurate prediction hinders effective workforce management, optimal resource allocation, and timely production planning. As a result, manufacturers experience frequent disruptions, missed production targets, increased operational costs, and decreased overall efficiency. The objective of this project is to develop a robust predictive model that leverages historical performance data, contextual variables (such as order complexity and machine downtime), and real-time feedback to forecast garment workers' productivity accurately. By implementing this predictive model, manufacturers aim to enhance operational efficiency, minimize production delays, improve workforce utilization, and ultimately maintain competitiveness in the global market."

This problem statement outlines the specific challenges faced in predicting garment workers' productivity and underscores the potential benefits of developing a reliable predictive model to address these challenges effectively.

2.2 Project Proposal (Proposed Solution)

A project proposal, also known as a proposed solution, outlines a structured plan to address a specific problem or opportunity. Key components typically include:

- **Problem Statement**: Clearly define the issue or opportunity that the project aims to solve or capitalize on.
- **Objectives**: Outline the specific goals and outcomes the project intends to achieve.
- **Proposed Solution**: Present a detailed description of the solution or approach to be implemented, including methodologies, technologies, and strategies.
- **Scope**: Define the boundaries of the project, specifying what will be included and what will be excluded.
- **Methodology**: Describe the methods and processes that will be used to execute the project, including timelines and milestones.
- **Resource Requirements**: Identify the resources needed (e.g., personnel, budget, equipment) to implement the proposed solution effectively.
- **Benefits**: Highlight the anticipated benefits and impacts of the proposed solution, both quantitatively and qualitatively.
- **Risks and Mitigation Strategies**: Identify potential risks associated with the project and propose strategies to mitigate them.

- **Evaluation and Success Criteria**: Outline how the project's success will be measured and evaluated against predefined criteria.
- **Budget**: Provide a detailed budget that covers all costs associated with executing the project.
- **Approval**: Specify the approval process and seek endorsement from stakeholders or decision-makers.
- **Appendices**: Include any additional supporting materials, such as charts, diagrams, or references.

A well-crafted project proposal serves as a roadmap for stakeholders, guiding them through the project's objectives, methods, and expected outcomes, ultimately gaining support and approval for implementation.

Initial Project Planning

1. Define Objectives and Scope

- **Objectives**: Clearly define what the project aims to achieve. For example, developing a predictive model to forecast garment workers' productivity accurately.
- **Scope**: Identify the boundaries of the project, including which aspects of productivity (e.g., output per hour, efficiency metrics) and which categories of garment workers (e.g., sewing operators, quality inspectors) will be considered.

2. Gather Requirements and Data

- **Requirements**: Determine the data sources needed for the project, such as historical productivity data, worker performance metrics, production schedules, and any relevant contextual variables (e.g., order complexity, machine downtime).
- **Data Collection**: Plan how to gather, clean, and preprocess the data to ensure it is suitable for analysis and model development.

3. Formulate Hypotheses

- **Hypotheses**: Develop initial hypotheses about factors that may influence garment worker productivity. These could include variables like worker experience, type of garment being produced, time of day, etc.

4. Choose Methods and Models

- **Methodology**: Decide on the analytical methods and modeling techniques suitable for predicting productivity. This may involve statistical analysis, machine learning algorithms (e.g., regression, classification), or a combination depending on the complexity of the problem.
- **Model Selection**: Select appropriate models (e.g., linear regression, decision trees, neural networks) based on the nature of the data and the project objectives.

5. Develop a Project Plan

- **Timeline**: Create a timeline with key milestones and deadlines for each phase of the project, including data collection, preprocessing, model development, validation, and deployment.
- **Resource Allocation**: Identify resources needed for the project, such as personnel with data science expertise, computing resources, and any external support required.

6. Consider Ethical and Legal Implications

- **Ethical Considerations**: Address potential ethical issues related to data privacy, worker consent, and fairness in model predictions.
- **Legal Compliance**: Ensure compliance with relevant regulations (e.g., data protection laws) and ethical guidelines in the use of worker data.

7. Define Success Metrics

- **Metrics**: Determine how success will be measured, such as accuracy of productivity predictions, reduction in production delays, or improvement in resource utilization.
- **Benchmarking**: Establish benchmarks against which the project outcomes will be compared to assess effectiveness.

8. Plan for Validation and Testing

- **Validation Strategy**: Outline how the predictive model will be validated and tested to ensure reliability and accuracy.
- **Iterative Improvement**: Plan for iterative refinement of the model based on validation results and feedback from stakeholders.

9. Stakeholder Communication and Engagement

- **Communication Plan**: Develop a plan for regular updates and communication with stakeholders, including management, production supervisors, and garment workers.
- **Feedback Mechanism**: Establish channels for gathering feedback and insights from stakeholders throughout the project lifecycle.

10. Risk Management

- **Risk Assessment**: Identify potential risks and challenges that could impact project success (e.g., data quality issues, model complexity).
- **Mitigation Strategies**: Develop contingency plans and mitigation strategies to address identified risks proactively.

By following these initial project planning steps, you can lay a solid foundation for predicting garment worker productivity effectively, ensuring alignment with business objectives and maximizing the impact of the project outcomes.

3 Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Raw Data Sources Identified

Developing a robust data collection plan is crucial for ensuring the success of predicting garment workers' productivity. Here's a structured approach to creating a data collection plan and identifying raw data sources:

1. Define Data Requirements

- **Productivity Metrics**: Determine the specific productivity metrics to be predicted (e.g., units produced per hour, efficiency ratios).
- **Contextual Variables**: Identify relevant variables that may influence productivity (e.g., worker experience, type of garment being produced, time of day, machine downtime, order complexity).
- **Data Granularity**: Specify the granularity of data required (e.g., hourly, daily) based on the project objectives and operational needs.

2. Identify Raw Data Sources

- **Production Records**: Obtain historical records of production output per worker or per production line.
- **Worker Performance Metrics**: Gather data on individual worker performance metrics such as error rates, rework rates, and attendance records.
- **Contextual Variables**: Collect data on contextual variables including order types, machine utilization rates, shift schedules, and any other factors known to impact productivity.

3. Data Collection Methods

- **Automated Systems**: Utilize existing production monitoring systems or data logging tools to capture real-time production data.
- **Manual Recording**: Implement standardized data collection procedures for variables that require manual recording (e.g., worker performance metrics).
- **Surveys or Interviews**: Conduct surveys or interviews with supervisors or workers to gather qualitative insights or additional contextual information.

4. Data Cleaning and Preprocessing

- **Data Cleaning**: Develop protocols for cleaning and validating raw data to ensure accuracy and consistency (e.g., handling missing values, outlier detection).
- **Normalization**: Normalize data if necessary to ensure uniformity and compatibility across different sources.
- **Feature Engineering**: Create new features from raw data that may enhance predictive accuracy (e.g., deriving metrics like average production rate over time).

5. Ensure Data Privacy and Security

- **Data Privacy**: Implement measures to protect worker privacy and comply with relevant data protection regulations (e.g., anonymization of personal data).
- **Data Security**: Secure data storage and access protocols to prevent unauthorized use or breaches.

6. Data Validation and Quality Assurance

- **Validation Procedures**: Develop procedures for validating data quality and consistency before proceeding with analysis (e.g., cross-checking records with production logs).
- **Quality Assurance**: Establish checkpoints throughout the data collection process to ensure adherence to data quality standards.

7. Documentation and Metadata

- **Documentation**: Maintain detailed documentation of data sources, collection methods, and preprocessing steps for transparency and reproducibility.
- **Metadata**: Create metadata descriptions to provide context and structure to the collected data, facilitating easier analysis and interpretation.

8. Pilot Testing

- **Pilot Phase**: Conduct a pilot test of the data collection plan to identify any practical challenges or issues and refine procedures accordingly.

9. Stakeholder Engagement

- **Engagement**: Involve relevant stakeholders (e.g., production managers, data analysts, IT staff) throughout the data collection process to ensure alignment with project goals and operational requirements.

10. Continuous Improvement

- **Feedback Mechanism**: Establish mechanisms for gathering feedback from stakeholders to iteratively improve the data collection plan and ensure ongoing relevance and reliability of the data.

By systematically planning and executing the data collection process with these considerations in mind, you can effectively gather the necessary raw data to develop accurate predictions of garment workers' productivity. This approach not only enhances the reliability of the predictive model but also supports informed decision-making and operational improvements in garment manufacturing processes.

3.2 Data Quality Report

1. Data Collection Methods

- **Source:**
  - Describe where the data was collected from (e.g., factory records, sensor data).
- **Sampling Method:**
  - Detail how the data was sampled (e.g., random sampling, stratified sampling).
- **Time Period:**
  - Specify the duration over which data was collected.

2. Data Completeness

- **Missing Values:**
  - Identify if there are missing values in the dataset and the percentage of missingness.
- **Duplicates:**
  - Check for duplicated entries in the dataset.

3. Data Accuracy

- **Outliers:**
  - Identify outliers in key features (e.g., productivity metrics, worker attributes).
- **Data Consistency:**
  - Ensure consistency across different sources of data if integrated.

4. Data Relevance

- **Feature Relevance:**
  - Evaluate the relevance of each feature to the prediction task (e.g., worker demographics, work environment conditions).
- **Target Variable Definition:**
  - Define how productivity is measured and ensure it aligns with the prediction goals.

5. Data Timeliness

- **Up-to-dateness:**
  - Assess how recent the data is and whether it reflects current conditions.

6. Data Integrity and Security

- **Data Security:**
  - Ensure compliance with data protection regulations (if applicable).
- **Data Integrity:**
  - Verify that data has not been altered or corrupted.

7. Data Preprocessing Steps

- **Normalization/Scaling:**
  - Describe any preprocessing steps applied to the data (e.g., normalization of numerical features).
- **Feature Engineering:**
  - Detail any transformations or new features created.

8. Exploratory Data Analysis (EDA)

- **Correlation Analysis:**
    - Investigate correlations between features and the target variable.
- **Visualization:**
    - Use graphs and charts to explore distributions and relationships.

9. Model Training Considerations

- **Training-Validation Split:**
    - Define how the dataset was split into training and validation sets.

Cross-validation:

- Describe any cross-validation techniques used.

10. Summary of Findings

- **Key Insights:**
    - Summarize significant findings from the data quality assessment.
- **Recommendations:**
    - Provide recommendations for improving data quality or gathering additional data if necessary.

Conclusion

This data quality report aims to ensure that the dataset used for predicting garment workers' productivity is reliable, accurate, and relevant to the prediction task. Addressing these aspects is crucial for developing robust and effective predictive models.

3.3 Data Preprocessing

1. Data Collection

- **Source**: Collect data from garment factories or relevant sources.
- **Variables**: Gather data on factors that could affect productivity (e.g., worker skills, machine types, work environment, etc.).
- **Format**: Ensure data is in a structured format (CSV, Excel, database) for easier preprocessing.

2. Data Cleaning

- **Missing Values**: Check for missing data and decide on strategies (e.g., imputation or removal).
- **Outliers**: Identify and handle outliers that could skew productivity measures.
- **Consistency**: Ensure data consistency across different sources or datasets.

3. Feature Selection

- **Relevance**: Select features that are likely to influence productivity (e.g., worker experience, machine types, production targets).
- **Dimensionality Reduction**: Use techniques like PCA (Principal Component Analysis) if dealing with high-dimensional data.

4. Feature Engineering

- **Create New Features**: Derive new features that might enhance prediction (e.g., hourly output rates, quality metrics, etc.).
- **Normalization/Scaling**: Scale numerical features if necessary to ensure they contribute equally to the model.

5. Data Transformation

- **Categorical Variables**: Encode categorical variables (e.g., worker skill levels, shift types) using techniques like one-hot encoding or label encoding.
- **Date/Time Variables**: Extract relevant information from date/time fields (e.g., day of the week, shift timings).

6. Handling Imbalanced Data (if applicable)

- **Class Imbalance**: If predicting productivity classes (e.g., low, medium, high), address any imbalance using techniques like oversampling or undersampling.

7. Splitting Data

- **Training and Testing Sets**: Split the dataset into training and testing sets (e.g., 80% training, 20% testing) to evaluate model performance.

8. Data Normalization

- **Normalization**: Normalize the data to ensure features are on a similar scale, especially for algorithms sensitive to scale differences (e.g., SVMs, neural networks).

9. Feature Scaling

- **Scaling**: Scale numerical features to a standard range (e.g., using Min-Max scaling or StandardScaler) to improve model training and convergence.

10. Handling Time-Series Data (if applicable)

- **Time-Series Features**: Extract time-related features such as trends, seasonality, and periodicity if productivity data spans over time.

11. Data Augmentation (if applicable)

- **Augmentation**: For techniques like deep learning, consider data augmentation methods to increase the variability of training examples.

12. Preparing Target Variable

- **Encoding Targets**: Encode the target variable (productivity level or output) appropriately for the prediction task.

13. Handling Multicollinearity

- **Multicollinearity**: Detect and handle multicollinear features that could affect model interpretation and performance.

14. Validation Strategy

- **Cross-Validation**: Implement cross-validation techniques (e.g., k-fold cross-validation) to assess model generalization and stability.

15. Documentation

- **Documentation**: Document all preprocessing steps and decisions taken to maintain transparency and reproducibility.

By following these steps, you can effectively preprocess the data for garment workers' productivity prediction, setting a solid foundation for building and evaluating predictive models. Adjustments may be necessary based on specific nuances of the data and the predictive modeling techniques chosen.

4 Model Development Phase

4.1 Model Selection Report

When selecting a model for predicting garment workers' productivity, it's essential to consider various factors such as the nature of the data, the goals of prediction (accuracy, interpretability, etc.), and the computational resources available. Here's a structured approach to preparing a model selection report:

1. Problem Understanding and Goals

- **Problem Statement**: Define the task clearly (predicting garment workers' productivity).
- **Objectives**: Specify the objectives (e.g., maximizing accuracy, understanding feature importance).

2. Data Overview

- **Dataset**: Describe the dataset used (size, features, target variable).
- **Features**: List the features considered for prediction (e.g., worker skills, machine types, environment factors).
- **Target Variable**: Define the target variable (productivity metric).

3. Preprocessing Steps

- **Data Cleaning**: Summarize how missing values, outliers, and inconsistencies were handled.
- **Feature Engineering**: Highlight any new features created or transformations applied.
- **Normalization/Scaling**: Document how numerical features were normalized or scaled.

4. Model Selection Process

- **Candidate Models**: List the models considered for prediction (e.g., linear regression, decision trees, random forest, neural networks).
- **Justification**: Explain why these models were selected (e.g., based on previous literature, suitability for the dataset).
- **Evaluation Metrics**: Specify the metrics used to evaluate model performance (e.g., accuracy, precision, recall, RMSE).

5. Model Evaluation

- **Training and Validation**: Describe how the models were trained and validated (e.g., cross-validation, hold-out validation).
- **Performance Comparison**: Present a summary of model performance using appropriate metrics.
- **Visualizations**: Include visualizations (e.g., ROC curves, learning curves) to aid in comparing models.

6. Model Selection Criteria

- **Criteria**: Define the criteria used to select the final model (e.g., highest accuracy, interpretability, computational efficiency).
- **Results Interpretation**: Discuss the implications of each model's performance in the context of garment workers' productivity prediction.

7. Chosen Model

- **Final Model**: Specify the selected model and justify the choice based on evaluation results and criteria.
- **Advantages**: Discuss the strengths of the chosen model compared to others considered.
- **Limitations**: Address any limitations or assumptions of the chosen model.

8. Future Considerations

- **Improvement Strategies**: Suggest potential improvements or future directions for enhancing model performance (e.g., feature selection, ensemble methods).
- **Deployment Feasibility**: Consider practical aspects of deploying the chosen model in a real-world setting.

9. Conclusion

- **Summary**: Recap the key findings and decisions from the model selection process.
- **Recommendations**: Provide recommendations for stakeholders based on the outcomes of the model selection report.

4.2 Initial Model Training Code, Model Validation and Evaluation Report

5 Model Optimization and Tuning Phase

5.1 Tuning Documentation

When tuning a machine learning model for predicting garment worker productivity, the goal is to optimize its performance by adjusting hyperparameters and fine-tuning the model configuration. Here's a structured approach to documenting the tuning process:

1. Problem Understanding and Goals

- **Objective**: Optimize the predictive model to accurately forecast garment worker productivity based on various factors.
- **Metrics**: Focus on metrics like mean squared error (MSE), R-squared (R2), or any domain-specific metric related to productivity.

2. Initial Model Selection

- **Base Model**: Specify the initial model chosen (e.g., RandomForestRegressor, GradientBoostingRegressor) and reasons for its selection.

3. Hyperparameters to Tune

- **Identify Hyperparameters**: List the hyperparameters of the chosen model that can significantly impact performance (e.g., number of estimators, max depth, learning rate).
- **Range**: Define the range or values to explore for each hyperparameter.

4. Tuning Methodology

- **Grid Search vs. Random Search**: Choose between Grid Search (if feasible given computational resources) or Random Search based on the number of hyperparameters and their potential impact.
- **Cross-Validation**: Utilize cross-validation (e.g., k-fold) to validate model performance and prevent overfitting during tuning.

5. Tuning Process

- **Grid/Random Search Setup**: Implement the search process using libraries like GridSearchCV or RandomizedSearchCV from scikit-learn.
- **Parameters**: Document the parameters used in the search, such as scoring metrics, number of folds for cross-validation, etc.

6. Evaluation Metrics

- **Metrics Tracking**: Record the performance metrics (MSE, R2, etc.) for each combination of hyperparameters during the tuning process.
- **Visualizations**: Include visualizations like learning curves or parameter effect plots to analyze how different configurations impact model performance.

7. Best Model Selection

- **Final Model**: Select the model configuration that yields the best performance based on the chosen evaluation metrics.
- **Justification**: Explain why this configuration was chosen over others, considering both performance metrics and computational feasibility.

8. **Model Validation**

- **Validation Set**: Use a separate validation set or cross-validation on the entire dataset to validate the final tuned model's performance.
- **Performance Summary**: Summarize the final model's performance metrics and compare them with the base model.

9. **Deployment Considerations**

- **Scalability**: Consider the scalability of the chosen model for deployment in production.
- **Maintenance**: Document any ongoing maintenance requirements for monitoring and updating the model.

5.2 Final Model Selection Justification

the final model for predicting garment worker productivity involves considering various factors such as model performance metrics, interpretability, computational efficiency, and suitability for deployment. Here's a structured approach to justifying the selection of the final model:

1. **Problem Understanding and Goals**

- **Objective**: The goal is to predict garment worker productivity accurately based on factors like worker skills, machine types, and environmental conditions.
- **Metrics**: Focus on metrics such as Mean Squared Error (MSE), R-squared (R2), or domain-specific metrics related to productivity.

2. **Model Evaluation**

- **Candidate Models**: Considered models such as RandomForestRegressor, GradientBoostingRegressor, and possibly others depending on initial exploration.
- **Evaluation Metrics**: Assess models based on performance metrics derived from cross-validation or validation sets.
- **Comparison**: Compare models based on metrics to determine which one best meets the project's objectives.

3. **Model Performance**

- **Performance Metrics**: Evaluate each model's performance using metrics like MSE, R2, or others deemed relevant.
- **Visualization**: Use visual aids such as learning curves or feature importance plots to understand model behavior and performance.

4. Interpretability vs. Complexity

- **Model Interpretability**: Consider how interpretable the model is, especially if stakeholders require insights into which factors influence productivity.
- **Complexity**: Balance between model complexity and performance—more complex models might achieve better accuracy but could be harder to interpret and deploy.

5. Computational Feasibility

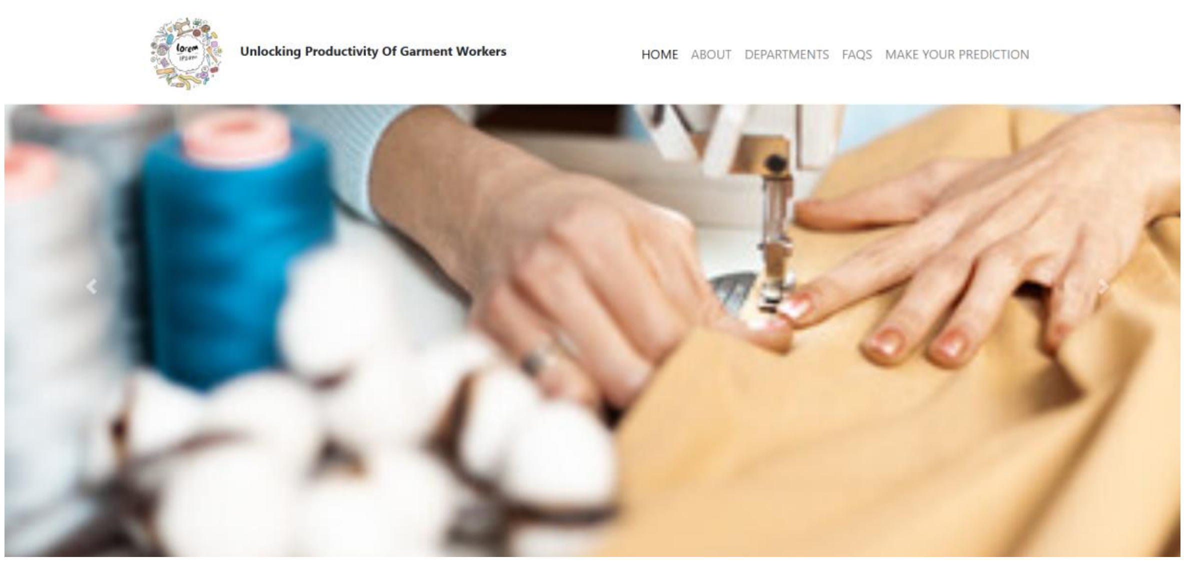
- **Resource Requirements**: Assess computational resources needed to train, validate, and potentially deploy the model in a real-world setting.
- **Scalability**: Consider how well the model scales with larger datasets or increased complexity of prediction tasks.

6. Final Model Selection

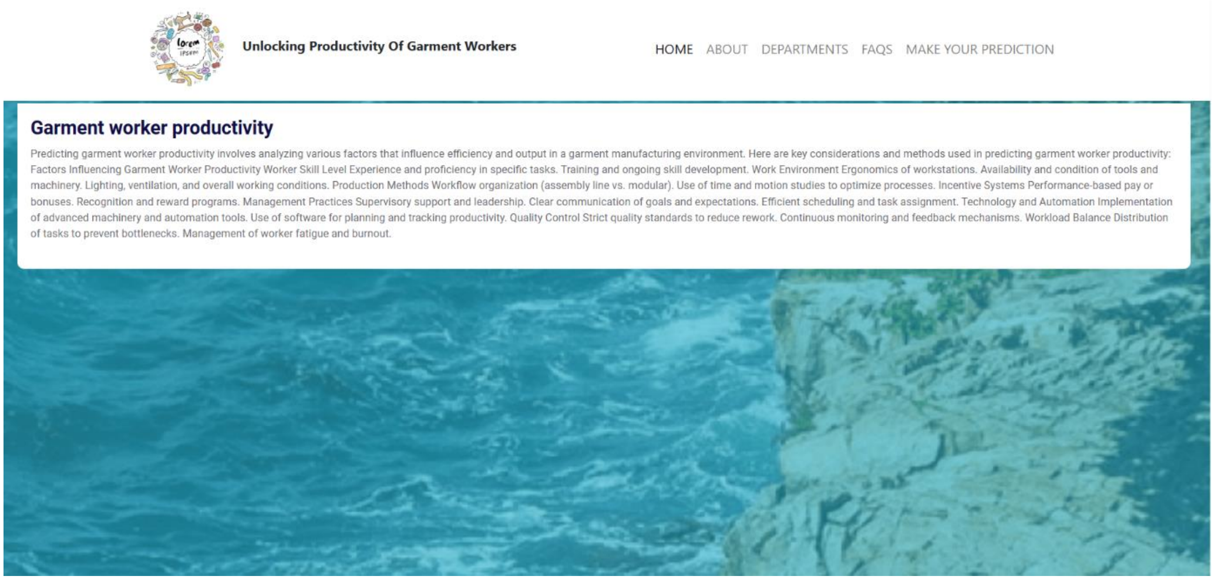
- **Chosen Model**: Select the model that best balances performance, interpretability, and computational feasibility based on the evaluation.
- **Justification**: Provide a clear rationale for choosing the final model, considering:
    - **Performance**: Highlight metrics where the chosen model outperformed others.
    - **Interpretability**: Discuss how well the model explains the relationship between input features and productivity outcomes.
    - **Feasibility**: Explain why the model is suitable for deployment considering computational constraints and scalability.

6.Results:

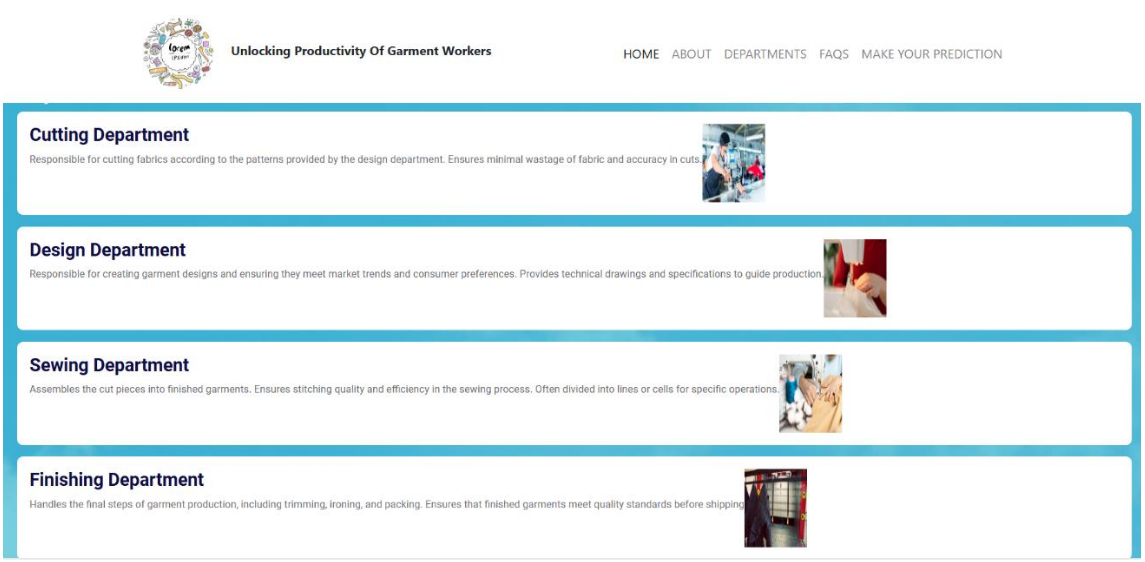
Index Page:



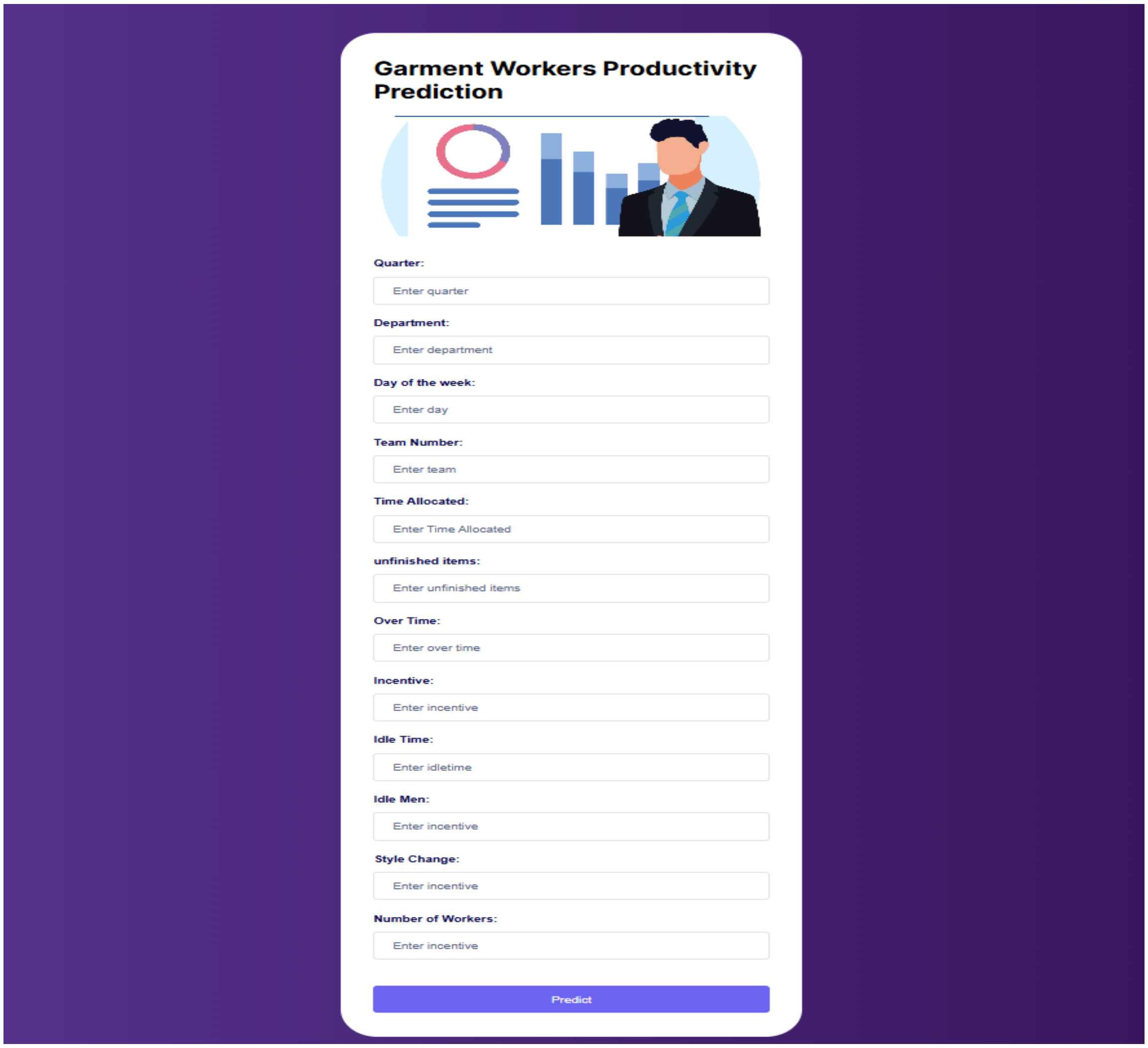
About Page:



Departments Page:



Prediction Page:


Garment Workers Productivity Prediction
Quarter:
Enter quarter
Department:
Enter department
Day of the week:
Enter day
Team Number:
Enter team
Time Allocated:
Enter Time Allocated
unfinished items:
Enter unfinished items
Over Time:
Enter over time
Incentive:
Enter incentive
Idle Time:
Enter idletime
Idle Men:
Enter incentive
Style Change:
Enter incentive
Number of Workers:
Enter incentive
Predict

Result Page:


Unlocking Productivity Of Garment Workers
HOME ABOUT DEPARTMENTS FAQS MAKE YOUR PREDICTION
Garment Worker Productivity Prediction
"Hence,based on calculation, the predicted Garment worker is: " {{ prediction_text }}

6.1 Output Screenshots

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import sklearn
```

```
df = pd.read_csv(r'C:\Users\srira\Downloads\miniProject\garments_worker_productivity.csv')
df.head()
```

	date	quarter	department	day	team	targeted_productivity	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_wc
0	1/1/2015	Quarter1	sweing	Thursday	8	0.80	26.16	1108.0	7080	98	0.0	0	0	
1	1/1/2015	Quarter1	finishing	Thursday	1	0.75	3.94	NaN	960	0	0.0	0	0	
2	1/1/2015	Quarter1	sweing	Thursday	11	0.80	11.41	968.0	3660	50	0.0	0	0	
3	1/1/2015	Quarter1	sweing	Thursday	12	0.80	11.41	968.0	3660	50	0.0	0	0	
4	1/1/2015	Quarter1	sweing	Thursday	6	0.80	25.90	1170.0	1920	50	0.0	0	0	

```
df.shape
```

```
(1197, 15)
```

```
df.describe()
```

Python

	team	targeted_productivity	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers	ac
count	1197.000000	1197.000000	1197.000000	691.000000	1197.000000	1197.000000	1197.000000	1197.000000	1197.000000	1197.000000	
mean	6.426901	0.729632	15.062172	1190.465991	4567.460317	38.210526	0.730159	0.369256	0.150376	34.609858	
std	3.463963	0.097891	10.943219	1837.455001	3348.823563	160.182643	12.709757	3.268987	0.427848	22.197687	
min	1.000000	0.070000	2.900000	7.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2.000000	
25%	3.000000	0.700000	3.940000	774.500000	1440.000000	0.000000	0.000000	0.000000	0.000000	9.000000	
50%	6.000000	0.750000	15.260000	1039.000000	3960.000000	0.000000	0.000000	0.000000	0.000000	34.000000	
75%	9.000000	0.800000	24.260000	1252.500000	6960.000000	50.000000	0.000000	0.000000	0.000000	57.000000	
max	12.000000	0.800000	54.560000	23122.000000	25920.000000	3600.000000	300.000000	45.000000	2.000000	89.000000	

```
df2 = df.drop(['date'], axis=1)
```

```
df2.shape
```

```
(1197, 14)
```

```python
df2.isnull().sum()
```

```
quarter                    0
department                 0
day                        0
team                       0
targeted_productivity      0
smv                        0
wip                      506
over_time                  0
incentive                  0
idle_time                  0
idle_men                   0
no_of_style_change         0
no_of_workers              0
actual_productivity        0
dtype: int64
```

Click to add a breakpoint

```python
        wip : 0,
    })
```

```python
df4 = df3.drop(['targeted_productivity'], axis=1)
```

```python
df4.head()
```

Pyt

	quarter	department	day	team	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers	actual_productivity
0	Quarter1	sweing	Thursday	8	26.16	1108.0	7080	98	0.0	0	0	59.0	0.940725
1	Quarter1	finishing	Thursday	1	3.94	0.0	960	0	0.0	0	0	8.0	0.886500
2	Quarter1	sweing	Thursday	11	11.41	968.0	3660	50	0.0	0	0	30.5	0.800570
3	Quarter1	sweing	Thursday	12	11.41	968.0	3660	50	0.0	0	0	30.5	0.800570
4	Quarter1	sweing	Thursday	6	25.90	1170.0	1920	50	0.0	0	0	56.0	0.800382

```python
df4=df4.rename(columns={
    'team':'team_number',
    'smv':'time_allocated',
    'wip':'unfinished_items',
    'no_of_style_change':'style_change'
})
df4
```

Python

	quarter	department	day	team_number	time_allocated	unfinished_items	over_time	incentive	idle_time	idle_men	style_change	no_of_workers
0	Quarter1	sweing	Thursday	8	26.16	1108.0	7080	98	0.0	0	0	59.0
1	Quarter1	finishing	Thursday	1	3.94	0.0	960	0	0.0	0	0	8.0
2	Quarter1	sweing	Thursday	11	11.41	968.0	3660	50	0.0	0	0	30.5
3	Quarter1	sweing	Thursday	12	11.41	968.0	3660	50	0.0	0	0	30.5
4	Quarter1	sweing	Thursday	6	25.90	1170.0	1920	50	0.0	0	0	56.0
...	...	...	...	...	...	...	...	...	...	...	...	...
1192	Quarter2	finishing	Wednesday	10	2.90	0.0	960	0	0.0	0	0	8.0
1193	Quarter2	finishing	Wednesday	8	3.90	0.0	960	0	0.0	0	0	8.0
1194	Quarter2	finishing	Wednesday	7	3.90	0.0	960	0	0.0	0	0	8.0
1195	Quarter2	finishing	Wednesday	9	2.90	0.0	1800	0	0.0	0	0	15.0
1196	Quarter2	finishing	Wednesday	6	2.90	0.0	720	0	0.0	0	0	6.0

1197 rows × 13 columns

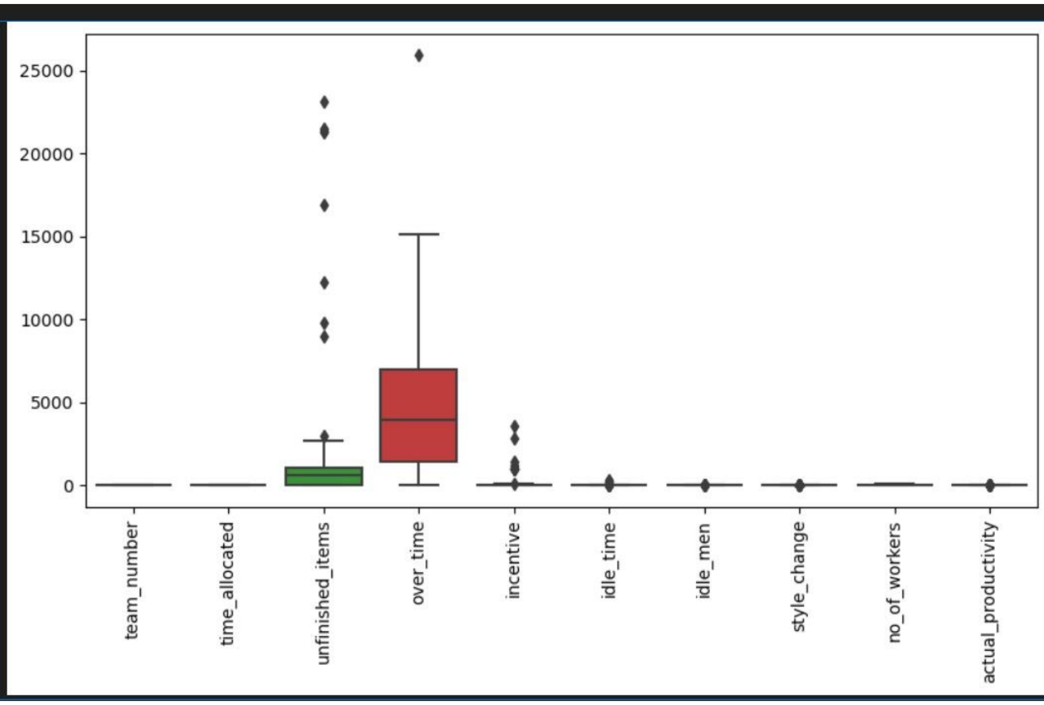
```python
df4.isnull().sum()
```

```
quarter                0
department             0
day                    0
team_number            0
time_allocated         0
unfinished_items       0
over_time              0
incentive              0
idle_time              0
idle_men               0
style_change           0
no_of_workers          0
actual_productivity    0
dtype: int64
```

```
plt.figure(figsize=(10,5))
p = sns.boxplot(data = df6, orient ='v',width=0.8)
plt.xticks(rotation=90)
```

[35]

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'team_number'),
  Text(1, 0, 'time_allocated'),
  Text(2, 0, 'unfinished_items'),
  Text(3, 0, 'over_time'),
  Text(4, 0, 'incentive'),
  Text(5, 0, 'idle_time'),
  Text(6, 0, 'idle_men'),
  Text(7, 0, 'style_change'),
  Text(8, 0, 'no_of_workers'),
  Text(9, 0, 'actual_productivity')])
```



```
Q1 = df4.incentive.quantile(0.25)
Q3 = df4.incentive.quantile(0.75)
Q1, Q3
```

```
(0.0, 50.0)
```

```
IQR = Q3 - Q1
IQR
```

```
50.0
```

```
lower_limit = Q1 - 1.5*IQR
upper_limit = Q3 + 1.5*IQR
lower_limit, upper_limit
```

```
(-75.0, 125.0)
```

```
df4[(df4.incentive<lower_limit)|(df4.incentive>upper_limit)]
```

	quarter	department	day	team_number	time_allocated	unfinished_items	over_time	incentive	idle_time	idle_men	style_change	no_of_workers
730	Quarter2	sweing	Thursday	1	22.52	1397.0	0	138	0.0	0	0	57.0
1128	Quarter2	finishing	Monday	11	2.90	0.0	0	960	0.0	0	0	8.0
1129	Quarter2	finishing	Monday	12	4.60	0.0	0	1080	0.0	0	0	9.0
1130	Quarter2	finishing	Monday	5	3.94	0.0	0	2880	0.0	0	0	12.0
1133	Quarter2	finishing	Monday	9	2.90	0.0	0	3600	0.0	0	0	15.0
1137	Quarter2	finishing	Monday	3	4.60	0.0	0	1440	0.0	0	0	12.0
1138	Quarter2	finishing	Monday	4	3.94	0.0	0	960	0.0	0	0	8.0
1139	Quarter2	finishing	Monday	1	3.94	0.0	0	960	0.0	0	0	8.0
1143	Quarter2	finishing	Monday	2	3.90	0.0	0	1200	0.0	0	0	10.0
1148	Quarter2	finishing	Monday	10	2.90	0.0	0	960	0.0	0	0	8.0
1149	Quarter2	finishing	Monday	8	3.90	0.0	0	960	0.0	0	0	8.0

```
df5 = df4[(df4.incentive>lower_limit)&(df4.incentive<upper_limit)]
```

```
df5.shape
```

```
(1186, 13)
```

```
Q1 = df5.unfinished_items.quantile(0.25)
Q3 = df5.unfinished_items.quantile(0.75)
Q1, Q3
```

```
(0.0, 1084.75)
```

```python
df5[(df5.unfinished_items<lower_limit)|(df5.unfinished_items>upper_limit)]
```

Python

	quarter	department	day	team_number	time_allocated	unfinished_items	over_time	incentive	idle_time	idle_men	style_change	no_of_workers	acti
561	Quarter1	sweing	Monday	1	22.94	16882.0	7020	113	0.0	0	0	58.5	
563	Quarter1	sweing	Monday	2	22.52	21385.0	7020	88	0.0	0	0	58.5	
564	Quarter1	sweing	Monday	3	22.52	21266.0	6840	70	0.0	0	0	57.0	
565	Quarter1	sweing	Monday	10	22.52	21540.0	6720	88	0.0	0	0	56.0	
568	Quarter1	sweing	Monday	12	15.26	12261.0	3600	63	0.0	0	0	35.0	
569	Quarter1	sweing	Monday	4	22.52	23122.0	5940	50	0.0	0	0	56.5	
570	Quarter1	sweing	Monday	9	29.12	8992.0	6960	55	0.0	0	0	58.0	
572	Quarter1	sweing	Monday	11	20.55	9792.0	6480	60	0.0	0	0	54.0	
573	Quarter1	sweing	Monday	6	18.79	2984.0	3960	30	0.0	0	0	33.0	

```python
df6 = df5[(df5.unfinished_items>lower_limit)&(df5.unfinished_items<upper_limit)]
```

```python
df6.shape
```

```
(1177, 13)
```

```python
Q1 = df6.over_time.quantile(0.25)
Q3 = df6.over_time.quantile(0.75)
Q1, Q3
```

```
(1440.0, 6960.0)
```

```python
IQR = Q3 - Q1
IQR
```

```
5520.0
```

```python
lower_limit = Q1 - 1.5*IQR
upper_limit = Q3 + 1.5*IQR
lower_limit, upper_limit
```

```
(-6840.0, 15240.0)
```

```
df7 = df6[(df6.over_time>lower_limit)&(df6.over_time<upper_limit)]
```

[33]

```
df7.shape
```

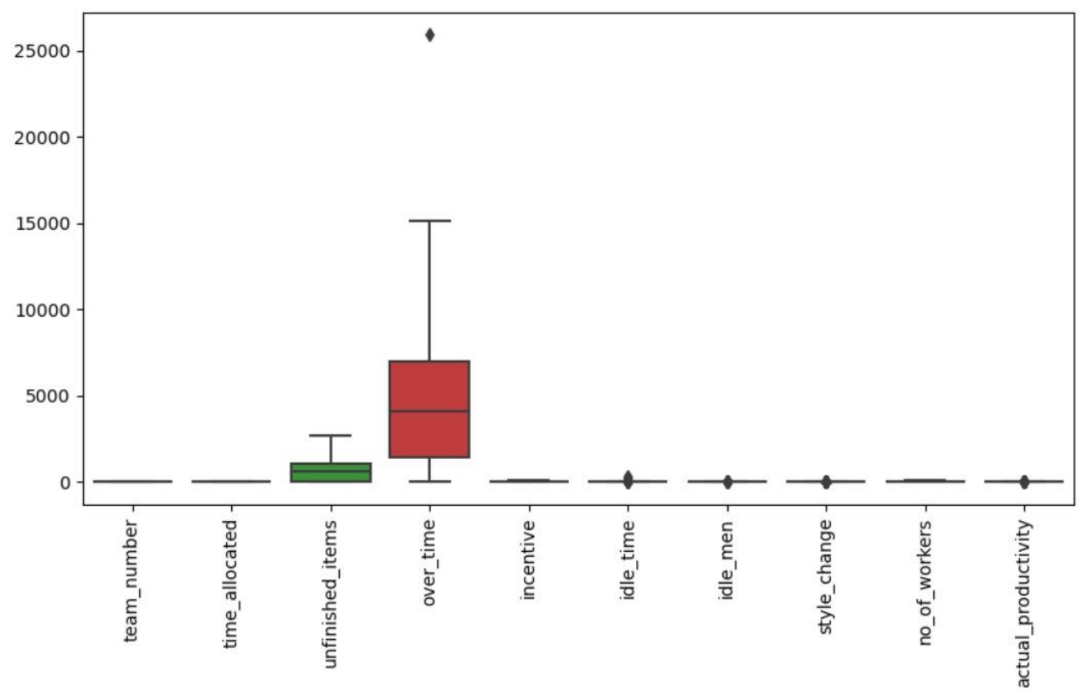
[34]

```
(1176, 13)
```

```
plt.figure(figsize=(10,5))
p = sns.boxplot(data = df6, orient ='v',width=0.8)
plt.xticks(rotation=90)
```

[35]

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'team_number'),
  Text(1, 0, 'time_allocated'),
  Text(2, 0, 'unfinished_items'),
  Text(3, 0, 'over_time'),
  Text(4, 0, 'incentive'),
  Text(5, 0, 'idle_time'),
  Text(6, 0, 'idle_men'),
  Text(7, 0, 'style_change'),
  Text(8, 0, 'no_of_workers'),
  Text(9, 0, 'actual_productivity')])
```




```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

[36]

```python
for i in range(0, df7.shape[1]):
    if df7.dtypes[i]=='object':
        df7[df7.columns[i]] = le.fit_transform(df7[df7.columns[i]])
```

```python
df7.head()
```

Python

	quarter	department	day	team_number	time_allocated	unfinished_items	over_time	incentive	idle_time	idle_men	style_change	no_of_workers	actual_pro
0	0	2	3	8	26.16	1108.0	7080	98	0.0	0	0	59.0	
1	0	1	3	1	3.94	0.0	960	0	0.0	0	0	8.0	
2	0	2	3	11	11.41	968.0	3660	50	0.0	0	0	30.5	
3	0	2	3	12	11.41	968.0	3660	50	0.0	0	0	30.5	
4	0	2	3	6	25.90	1170.0	1920	50	0.0	0	0	56.0	

```
x = df7.drop(['actual_productivity'], axis=1)
y = df7.actual_productivity
```

```
x.shape, y.shape
```

```
((1176, 12), (1176,))
```

```
y.head()
```

```
0    0.940725
1    0.886500
2    0.800570
3    0.800570
4    0.800382
Name: actual_productivity, dtype: float64
```

```
#Standarization
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
x.iloc[:,:] = scaler.fit_transform(x.iloc[:,:])
```

```
x.head()
```

	quarter	department	day	team_number	time_allocated	unfinished_items	over_time	incentive	idle_time	idle_men	style_change	no_of_workers
0	-1.160117	0.780739	0.250366	0.455616	1.008608	0.926217	0.760786	2.413020	-0.057985	-0.114022	-0.355135	1.097504
1	-1.160117	-0.468444	0.250366	-1.571751	-1.018120	-1.019391	-1.102067	-0.847073	-0.057985	-0.114022	-0.355135	-1.201499
2	-1.160117	0.780739	0.250366	1.324487	-0.336767	0.680382	-0.280220	0.816240	-0.057985	-0.114022	-0.355135	-0.187233
3	-1.160117	0.780739	0.250366	1.614110	-0.336767	0.680382	-0.280220	0.816240	-0.057985	-0.114022	-0.355135	-0.187233
4	-1.160117	0.780739	0.250366	-0.123632	0.984893	1.035087	-0.809855	0.816240	-0.057985	-0.114022	-0.355135	0.962269

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=42)
```

46]

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

47]

```
(823, 12)
(353, 12)
(823,)
(353,)
```

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from math import sqrt
from sklearn.metrics import mean_absolute_percentage_error
```

48]

```python
from sklearn.linear_model import LinearRegression
linreg = LinearRegression()
linreg.fit(x_train,y_train)
pred_linreg = linreg.predict(x_test)
print("MAE  :", mean_absolute_error(y_test, pred_linreg))
print("MSE  :", mean_squared_error(y_test, pred_linreg))
print("RMSE :",sqrt(mean_squared_error(y_test, pred_linreg)))
print("MAPE :",mean_absolute_percentage_error(y_test, pred_linreg))
```

49]

```
MAE  : 0.09889801648754916
MSE  : 0.018535367319974336
RMSE : 0.13614465586270486
MAPE : 0.16747657653644724
```

```python
from sklearn.neighbors import KNeighborsRegressor
knn = KNeighborsRegressor()
knn.fit(x_train,y_train)
pred_knn = knn.predict(x_test)
print("MAE  :", mean_absolute_error(y_test, pred_knn))
print("MSE  :", mean_squared_error(y_test, pred_knn))
print("RMSE :",sqrt(mean_squared_error(y_test, pred_knn)))
print("MAPE :",mean_absolute_percentage_error(y_test, pred_knn))
```

50]

```
MAE  : 0.09461994102266287
MSE  : 0.017693060885064893
RMSE : 0.13301526560912058
MAPE : 0.1601878271654436
```

```python
from sklearn.ensemble import RandomForestRegressor
randf = RandomForestRegressor(random_state=42)
randf.fit(x_train,y_train)
pred_randf = randf.predict(x_test)
print("MAE  :", mean_absolute_error(y_test, pred_randf))
print("MSE  :", mean_squared_error(y_test, pred_randf))
print("RMSE :",sqrt(mean_squared_error(y_test, pred_randf)))
print("MAPE :",mean_absolute_percentage_error(y_test, pred_randf))
```

```
MAE  : 0.08366785595438364
MSE  : 0.015441874867015823
RMSE : 0.12426534057015183
MAPE : 0.14067390864389964
```

```python
from sklearn.ensemble import GradientBoostingRegressor
gb = GradientBoostingRegressor(random_state=42)
gb.fit(x_train,y_train)
pred_gb = gb.predict(x_test)
print("MAE  :", mean_absolute_error(y_test, pred_gb))
print("MSE  :", mean_squared_error(y_test, pred_gb))
print("RMSE :",sqrt(mean_squared_error(y_test, pred_gb)))
print("MAPE :",mean_absolute_percentage_error(y_test, pred_gb))
```

```
MAE  : 0.08052610453252707
MSE  : 0.013325110632581337
RMSE : 0.11543444300806123
MAPE : 0.1351165000418134
```

```python
from sklearn.ensemble import AdaBoostRegressor
adab = AdaBoostRegressor(random_state=42)
adab.fit(x_train,y_train)
pred_adab = adab.predict(x_test)
print("MAE  :", mean_absolute_error(y_test, pred_adab))
print("MSE  :", mean_squared_error(y_test, pred_adab))
print("RMSE :",sqrt(mean_squared_error(y_test, pred_adab)))
print("MAPE :",mean_absolute_percentage_error(y_test, pred_adab))
```

```
MAE  : 0.09914994225330438
MSE  : 0.01804214472829372
RMSE : 0.1343210509499301
MAPE : 0.15873014712450173
```

```python
import xgboost
from xgboost import XGBRegressor
xgb = XGBRegressor()
xgb.fit(x_train, y_train)
pred_xgb = xgb.predict(x_test)
print("MAE  :", mean_absolute_error(y_test, pred_xgb))
print("MSE  :", mean_squared_error(y_test, pred_xgb))
print("RMSE :",sqrt(mean_squared_error(y_test, pred_xgb)))
print("MAPE :",mean_absolute_percentage_error(y_test, pred_xgb))
```

```
MAE  : 0.0889760853906732
MSE  : 0.01729979938858328
RMSE : 0.13152870176726933
MAPE : 0.14570268332051675
```

```python
d = {'Algorithm': ['Linear Regression','K Nearest Neighbors', 'Support Vector Machine','Decision Tree',
                   'Random Forest','Gradient Boost','Adaboost','XGBoost'],
     'MSE' : [0.017,0.015,0.014,0.022,0.013, 0.012,0.017,0.015],
     'RMSE': [0.131, 0.123, 0.121,0.148,0.116,0.112,0.131,0.123],
     'MAE' : [0.091,0.079,0.084,0.083,0.072,0.075,0.094,0.078],
     'MAPE': [15.30,13.65,14.43,13.78,12.02,12.67,15.01,12.62]
    }
df_accuracy = pd.DataFrame(data=d)
df_accuracy
```

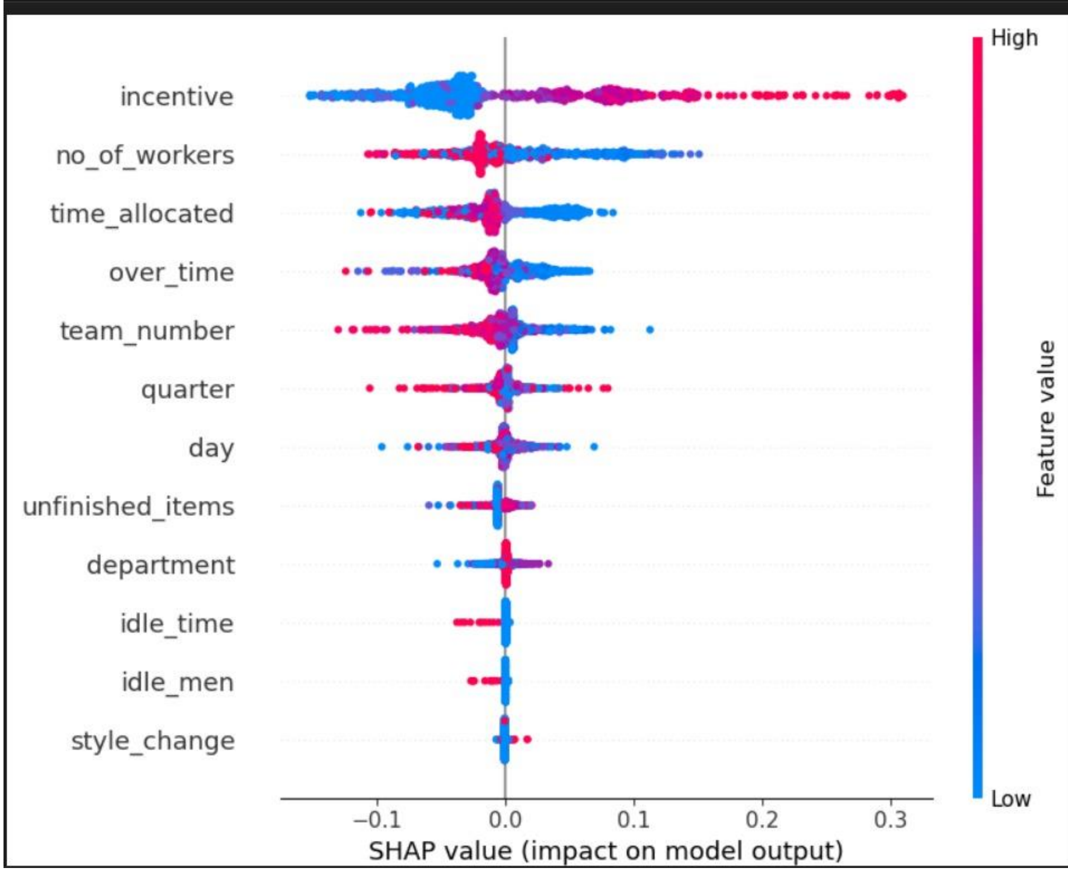
	Algorithm	MSE	RMSE	MAE	MAPE
0	Linear Regression	0.017	0.131	0.091	15.30
1	K Nearest Neighbors	0.015	0.123	0.079	13.65
2	Support Vector Machine	0.014	0.121	0.084	14.43
3	Decision Tree	0.022	0.148	0.083	13.78
4	Random Forest	0.013	0.116	0.072	12.02
5	Gradient Boost	0.012	0.112	0.075	12.67
6	Adaboost	0.017	0.131	0.094	15.01
7	XGBoost	0.015	0.123	0.078	12.62

```python
import shap
```

```python
# Create object that can calculate shap values
explainer = shap.TreeExplainer(randf)
# Calculate Shap values
shap_values = explainer.shap_values(x_train)
```

```python
shap.summary_plot(shap_values, x_train)
```

```python
shap.summary_plot(shap_values, x_train)
```



```
print(shap_values)
```

```
[[ 0.00289996  0.0011013  -0.00151864 ...  0.00032938 -0.00077743
  -0.02000564]
 [ 0.00107953  0.00080118 -0.00146752 ...  0.00035726 -0.00034355
  -0.0194016 ]
 [ 0.0019234  -0.00117553  0.00186971 ...  0.00190697 -0.000561
  -0.01182943]
 ...
 [ 0.01351555 -0.00068578  0.01467398 ...  0.00036089 -0.00025657
   0.07831029]
 [-0.00699838 -0.00919669 -0.01376068 ...  0.00035768 -0.00043395
   0.10130167]
 [ 0.00293606  0.00109989 -0.00253797 ...  0.00065866  0.00178236
  -0.03605201]]
```

```
import pickle
pickle.dump(xgb,open('productivity2.pkl','wb'))
```

6 Advantages & Disadvantages

ADVANTAGES:

Controlled Environment: Experiments allow researchers to control variables and conditions, minimizing external influences and enhancing the reliability of results. o Causal Inference: By manipulating variables and observing outcomes, experiments enable researchers to establish causal relationships between variables, providing stronger evidence than observational studies. o **Precision and Replicability:** Experiments often yield quantitative data that can be precisely measured and replicated by other researchers to verify findings and build upon them.

Testing Hypotheses: Experiments provide a structured approach to test specific hypotheses or research questions, allowing for systematic investigation and validation of theoretical predictions **Ability to Explore Complex Relationships**: Experimental designs can be used to explore complex relationships and interactions among variables that may not be easily observable in natural settings.

Ethical Considerations: In many cases, experiments can be designed to adhere to ethical guidelines more rigorously compared to observational studies, ensuring participant safety and informed consent.

DISADVANTAGES:

Artificiality: Experimental settings may be artificial and may not fully replicate real-world conditions, limiting the generalizability of findings to real-life situations.

Cost and Time Intensive: Designing and conducting experiments can be resource-intensive, requiring significant time, funding, and specialized equipment or facilities.

Ethical Concerns: Some experimental designs may pose ethical dilemmas, especially when involving human subjects or sensitive topics, requiring careful consideration and oversight.

Potential for Bias: Despite efforts to control variables, experiments can still be susceptible to biases, such as experimenter bias or participant biases, which may influence results.

Limited Scope: Experiments may focus narrowly on specific variables or conditions, potentially overlooking broader contexts or interactions that could influence outcomes.

Practical Limitations: Certain research questions or phenomena may not be feasibly studied through experiments due to ethical constraints, logistical challenges, or the nature of the

7 Conclusion

Predictive modeling for garment workers' productivity holds immense potential for optimizing manufacturing processes:

• Efficiency Boost: Predictive models streamline operations, improving resource allocation and workflow management.

• Quality Enhancement: Early detection of issues allows for preemptive actions to maintain product quality and meet production targets.

• Worker Support: Insights from predictions aid in enhancing worker training, task assignment, and overall job satisfaction.

• Cost Savings: Effective use of resources reduces operational costs and enhances profitability.

• Competitive Edge: Adoption of predictive analytics provides a strategic advantage in adapting to market dynamics and achieving sustainable growth.

Moving forward, continued advancements in technology and data analytics will further refine these capabilities, paving the way for enhanced productivity and competitiveness in the garment manufacturing sector.

9.Future Scope

The future scope for garment workers' productivity predictions is promising and involves several key areas of development:

1. Advanced Data Analytics: Integration of advanced analytics techniques such as machine learning and AI to improve the accuracy and granularity of productivity predictions. This includes predictive maintenance models, anomaly detection, and real-time performance monitoring.

2. IoT and Wearable Technology: Expansion of IoT devices and wearable technology to collect real-time data on worker activities, health metrics, and environmental conditions. This data can enhance predictive models and optimize worker safety and productivity.

3. Smart Manufacturing Systems: Implementation of smart manufacturing systems that use predictive analytics to automate decision-making processes, optimize production schedules, and reduce waste. This includes adaptive manufacturing processes that respond in real-time to fluctuations in demand and supply chain dynamics.

4. Human-Centric Design: Development of predictive models that prioritize worker well-being and ergonomics. This includes predicting fatigue levels, optimizing work schedules to prevent burnout, and ensuring fair and equitable treatment of workers.

5. Ethical Considerations: Addressing ethical challenges related to data privacy, algorithm transparency, and bias in predictive models. Ensuring that predictive analytics are used responsibly to benefit both workers and organizations.

6. Collaborative Robotics (Cobots): Integration of collaborative robots in garment manufacturing processes to work alongside human workers. Predictive analytics can optimize human-robot collaboration, improving efficiency and safety in tasks such as material handling and assembly.

7. Continuous Improvement: Establishing feedback loops to continuously refine predictive models based on real-time data and feedback from workers and supervisors. This iterative approach ensures that predictions remain relevant and accurate over time.

8. Global Supply Chain Integration: Extending predictive models across global supply chains to optimize production across multiple locations, anticipate logistics challenges, and maintain consistent product quality and delivery schedules.

9. Industry 4.0 Initiatives: Alignment with Industry 4.0 principles to create interconnected, data-driven manufacturing ecosystems. This includes leveraging cloud computing, big data analytics, and edge computing to enhance the scalability and responsiveness of predictive analytics solutions. 10. Research and Development: Continued investment in research and development to explore new technologies, methodologies, and applications for enhancing garment workers' productivity predictions. This includes collaboration between academia, industry, and government agencies to drive innovation in this field.

In summary, the future scope for garment workers' productivity predictions involves leveraging cutting-edge technologies, prioritizing worker-centric approaches, addressing ethical considerations, and fostering continuous innovation to optimize manufacturing processes and enhance overall industry competitiveness.

8 Appendix

8.1 Source Code

INDEX.HTML

```
<html>

  <head> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-
```

```
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous" />

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Productivity Prediction</title>

    <style>


        .h{

            background-
image:url("https://res.cloudinary.com/dcy90slhx/image/upload/v1718992842/garment12_
pslkis.jpg");

            height:100vh;

            width:100vw;

            background-size: cover;

        }

        .image{

            height:100px;

            width:100px;

            border-radius: 15px;

        }

        .h1{

            font-size:16px;

            font-weight: bold;
```

```
        }
        .d{
            font-size:25px;
            font-weight:bold;
        }
        .department{
            background-image:url("https://res.cloudinary.com/dcy90slhx/image/upload/v1718992865/garment11_kasql6.jpg");
            height:100vh;
            width:100vw;
            background-size: cover;
        }
        .img{
            height:200px;
            width:200px;
        }
        .p{
            color:black;
            font-size: 20px;
            font-weight:bold;
            padding: 5px;
        }
    </style>
</head>
  <body>
        <div class="h d-flex flex-column justify-content-center" >
        <nav class="navbar navbar-expand-lg navbar-light bg-white fixed-top">
      <div class="container d-flex flex-row">
        <a class="navbar-brand" href="# ">
          <img
```

```
src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718992771/garment17_flgqap.webp"
            class="image"
          />
        </a>
        <div>
            <h1 class="h1">Unlocking Productivity Of Garment Workers</h1>
        </div>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
          <div class="navbar-nav ml-auto">
            <a class="nav-link active" id="Home" href="{{ url_for('home') }}" >
              HOME
              <span class="sr-only">(current)</span>
            </a>
            <a class="nav-link" href="{{ url_for('happy1') }}" id="navItem2">ABOUT</a>
            <a class="nav-link" href="{{ url_for('happy2') }}" id="navItem3">DEPARTMENTS</a>
            <a class="nav-link" href="" id="navItem4">FAQS</a>
            <a class="nav-link" href="{{ url_for('happy') }}">MAKE YOUR PREDICTION</a>


          </div>
        </div>
      </div>
    </nav>
```

```html
<div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel" href="#Home" >
<ol class="carousel-indicators">
  <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
  <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
  <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
</ol>
<div class="carousel-inner">
  <div class="carousel-item active">
    <img src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718991221/240_F_429303037_KwFA54WSBGD39FlqMkdnauunWNAr9AoJ_mox6mk.jpg" class="d-block w-100" alt="...">
  </div>
  <div class="carousel-item">
    <img src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718992792/garment16_eagqsf.jpg" class="d-block w-100" alt="...">
  </div>
  <div class="carousel-item">
    <img src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718992753/garment19_fxgpue.webp" class="d-block w-100" alt="...">
  </div>
</div>
<a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="sr-only">Next</span>
```

```
                </a>
               </div>
               </div>


            </body>
            </html>
```

About Page

```
    <!DOCTYPE html>
 <html>
  <head>
    <link rel="stylesheet"
 href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-
 JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
 crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
 DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
 crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
 integrity="sha384-
 9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
 crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
 integrity="sha384-
 B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
 crossorigin="anonymous"></script>
 <style>
 .detailed-view-bg-container {
  background-image: url("https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/seabg.png");
  height: 100vh;
  background-size: cover;
 }
 .detailed-view-heading {
  color: white;
  font-family: "Roboto";
  font-size: 28px;
```

```
  font-weight: bold;
  padding: 24px;
}
.detailed-view-card-container {
  background-color: white;
  border-bottom-right-radius: 8px;
  border-bottom-left-radius: 8px;
  margin: 24px;
}
.detailed-view-card-heading {
  color: #0f0e46;
  font-family: "Roboto";
  font-size: 23px;
  font-weight: bold;
}
.detailed-view-card-description {
  color: #6c6b70;
  font-family: "Roboto";
  font-size: 13px;
}
.detailed-view-card-text-container {
  padding: 16px;
}
.image{
        height:100px;
        width:100px;
        border-radius: 15px;
    }
    .h1{
        font-size:16px;
        font-weight: bold;
```

```
        }

</style>
  </head>
  <body>
<nav class="navbar navbar-expand-lg navbar-light bg-white fixed-top">
      <div class="container d-flex flex-row">
        <a class="navbar-brand" href="# ">
          <img

src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718992771/garment17_flgqap.webp"
            class="image"
          />
        </a>
        <div>
            <h1 class="h1">Unlocking Productivity Of Garment Workers</h1>
        </div>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
          <div class="navbar-nav ml-auto">
            <a class="nav-link active" id="Home" href="{{ url_for('home') }}">
              HOME
              <span class="sr-only">(current)</span>
            </a>
            <a class="nav-link" href="{{ url_for('happy1') }}" id="navItem2">ABOUT</a>
            <a class="nav-link" href="{{ url_for('happy2') }}" id="navItem3">DEPARTMENTS</a>
            <a class="nav-link" href="" id="navItem4">FAQS</a>
```

MAKE YOUR PREDICTION

</div>

</div>

</div>

</nav>

</br>

<div class="detailed-view-bg-container">

<h1 class="detailed-view-heading">About</h1>

<div class="detailed-view-card-container">

<div class="detailed-view-card-text-container">

<h1 class="detailed-view-card-heading">Garment worker productivity</h1>

<p class="detailed-view-card-description">

Predicting garment worker productivity involves analyzing various factors that influence efficiency and output in a garment manufacturing environment. Here are key considerations and methods used in predicting garment worker productivity:

Factors Influencing Garment Worker Productivity

Worker Skill Level

Experience and proficiency in specific tasks.

Training and ongoing skill development.

Work Environment

Ergonomics of workstations.

Availability and condition of tools and machinery.

Lighting, ventilation, and overall working conditions.

Production Methods

Workflow organization (assembly line vs. modular).

Use of time and motion studies to optimize processes.

Incentive Systems

Performance-based pay or bonuses.

Recognition and reward programs.

Management Practices

Supervisory support and leadership.

Clear communication of goals and expectations.

Efficient scheduling and task assignment.

Technology and Automation

Implementation of advanced machinery and automation tools.

Use of software for planning and tracking productivity.

Quality Control

Strict quality standards to reduce rework.

Continuous monitoring and feedback mechanisms.

Workload Balance

Distribution of tasks to prevent bottlenecks.

Management of worker fatigue and burnout.

```
          </p>
        </div>
      </div>
    </div>
  </body>
</html>
```

Department Page

```
<!DOCTYPE html>
<html>
  <head>
```

```
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>

<style>

.favourite-places-bg-container {

  background-image: url("https://d1tgh8fmlzexmh.cloudfront.net/ccbp-static-website/towerbg.png");

  height: 100vh;

  wdidth:100vw;

  background-size: cover;

  padding: 24px;

}

.favourite-places-heading {

  color: white;

  font-family: "Roboto";

  font-size: 28px;

  font-weight: bold;

}

.favourite-place-card-container {

  background-color: white;

  border-radius: 8px;

  padding: 16px;

  margin-bottom: 15px;

}
```

```
.favourite-place-card-heading {
  color: #0f0e46;
  font-family: "Roboto";
  font-size: 23px;
  font-weight: bold;
}
.favourite-place-card-description {
  color: #6c6b70;
  font-family: "Roboto";
  font-size: 13px;
}
.favourite-place-card-image {
  width: 80px;
  height: 100px;
}
.favourite-place-card-text-container {
  margin-right: 15px;
}
.image{
        height:100px;
        width:100px;
        border-radius: 15px;
    }
    .h1{
        font-size:16px;
        font-weight: bold;


    }

</style>
  </head>
```

```html
  <body>
<nav class="navbar navbar-expand-lg navbar-light bg-white fixed-top">
     <div class="container d-flex flex-row">
       <a class="navbar-brand" href="# ">
         <img

src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718992771/garment17_flgqap.webp"
           class="image"
         />
       </a>
       <div>
           <h1 class="h1">Unlocking Productivity Of Garment Workers</h1>
       </div>
       <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
         <span class="navbar-toggler-icon"></span>
       </button>
       <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
         <div class="navbar-nav ml-auto">
           <a class="nav-link active" id="Home" href="{{ url_for('home') }}" >
             HOME
             <span class="sr-only">(current)</span>
           </a>
           <a class="nav-link" href="{{ url_for('happy1') }}" id="navItem2">ABOUT</a>
           <a class="nav-link" href="{{ url_for('happy2') }}" id="navItem3">DEPARTMENTS</a>
           <a class="nav-link" href="" id="navItem4">FAQS</a>
           <a class="nav-link" href="{{ url_for('happy') }}">MAKE YOUR PREDICTION</a>


         </div>
       </div>
     </div>
```

```
    </nav>
</br>
<div class="mt-5">
<div class="favourite-places-bg-container">

      <h1 class="favourite-places-heading">Departments</h1>
      <div class="favourite-place-card-container d-flex flex-row">
        <div>
          <h1 class="favourite-place-card-heading">Cutting Department</h1>
          <p class="favourite-place-card-description">
            Responsible for cutting fabrics according to the patterns provided by the design department.
Ensures minimal wastage of fabric and accuracy in cuts.</p>
        </div>
        <img
src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718991696/garment3_mwx263.jpg"
class="favourite-place-card-image" />
      </div>
      <div class="favourite-place-card-container d-flex flex-row">
        <div>
          <h1 class="favourite-place-card-heading">Design Department</h1>
          <p class="favourite-place-card-description">
            Responsible for creating garment designs and ensuring they meet market trends and consumer
preferences.
Provides technical drawings and specifications to guide production.</p>
        </div>
        <img src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718992888/garmet0_o3ngo9.jpg"
class="favourite-place-card-image" />
      </div>
      <div class="favourite-place-card-container d-flex flex-row">
        <div>
          <h1 class="favourite-place-card-heading">Sewing Department</h1>
          <p class="favourite-place-card-description">
```

```
            Assembles the cut pieces into finished garments.
Ensures stitching quality and efficiency in the sewing process.
Often divided into lines or cells for specific operations.
          </p>
        </div>
        <img
src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718991221/240_F_429303037_KwFA54W
SBGD39FlqMkdnauunWNAr9AoJ_mox6mk.jpg" class="favourite-place-card-image" />
      </div>
      <div class="favourite-place-card-container d-flex flex-row">
        <div>
          <h1 class="favourite-place-card-heading">Finishing Department</h1>
          <p class="favourite-place-card-description">
            Handles the final steps of garment production, including trimming, ironing, and packing.
Ensures that finished garments meet quality standards before shipping
          </p>
        </div>
        <img
src="https://res.cloudinary.com/dcy90slhx/image/upload/v1718992741/garment20_adk5oz.webp"
class="favourite-place-card-image" />
      </div>
</div>
</div>
  </body>
</html>
```

PREDICT.HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Productivity Prediction</title>
    <style>
        @import
url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&disp
l ay=swap');

        * {
margin: 0;
padding: 0;
            box-sizing: border-box;
        }

        body {
            font-family: 'Inter', sans-serif;
            background: linear-gradient(to right, #6441a5, #2a0845);
        }

        .formbold-main-wrapper {
display: flex;             align-
items: center;             justify-
content: center;
padding: 48px;
        }

        .formbold-form-wrapper {
            margin: 0 auto;
max-width: 570px;
width: 100%;
background: white;
padding: 40px;
            border-radius: 45px;
        }

        .formbold-input-group {
            margin-bottom: 18px;
        }
```

```
        .formbold-form-select {
width: 100%;            padding:
12px 22px;            border-
radius: 5px;            border: 1px
solid #dde3ec;
background: #ffffff;            font-
size: 16px;            color:
#536387;            outline: none;
resize: none;
        }

        .formbold-input-radio-wrapper {
          margin-bottom: 25px;
        }

        .formbold-radio-flex {
display: flex;            flex-
direction: column;            gap:
15px;
        }

        .formbold-radio-label {
font-size: 14px;            line-
height: 24px;            color:
#07074d;            position:
relative;            padding-left:
25px;            cursor: pointer;
-webkit-user-select: none;
          -moz-user-select: none;
 -ms-user-select: none;
          user-select: none;
        }

        .formbold-input-radio
{            position:
absolute;            opacity:
0;            cursor: pointer;
        }
```

```
        .formbold-radio-checkmark {
position: absolute;
            top: -1px;            left: 0;
height: 18px;            width:
18px;            background-color:
#ffffff;            border: 1px solid
#dde3ec;
            border-radius: 50%;
        }

        .formbold-radio-label .formbold-input-radio:checked~.formbold-
radiocheckmark {
            background-color: #6a64f1;
        }

        .formbold-radio-checkmark:after {
            content: '';
position: absolute;
            display: none;
        }

        .formbold-radio-label .formbold-input-radio:checked~.formbold-
radiocheckmark:after {
            display: block;
        }

        .formbold-radio-label .formbold-radio-checkmark:after
{            top: 50%;            left: 50%;
            width: 10px;
height: 10px;
border-radius: 50%;
background: #ffffff;
            transform: translate(-50%, -50%);
        }

        .formbold-form-input {
width: 100%;            padding:
13px 22px;            border-
radius: 5px;            border: 1px
```

```
solid #dde3ec;
background: #ffffff;            font-
weight: 500;            font-size:
16px;            color: #07074d;
outline: none;            resize:
none;
    }

    .formbold-form-input::placeholder {
      color: #536387;
    }

    .formbold-form-input:focus {
border-color: #6a64f1;
      box-shadow: 0px 3px 8px rgba(0, 0, 0, 0.05);
    }

    .formbold-form-label {
color: #07074d;        font-
size: 16px;        line-height:
24px;        display: block;
margin-bottom: 10px;
    font-weight: bold; /* Add this line to make the labels bolder */          font-family:
sans-serif; /* Replace 'Your Chosen Font' with the desired font family */
  }

    .formbold-btn {            text-
align: center;            width:
100%;            font-size: 16px;
border-radius: 5px;
padding: 14px 25px;
border: none;            font-weight:
500;            background-color:
#6a64f1;
      color: white;
cursor: pointer;
      margin-top: 25px;
    }
```

```
        .formbold-btn:hover {
          box-shadow: 0px 3px 8px rgba(0, 0, 0, 0.05);
        }
    </style>
</head>

<body>
    <div class="formbold-main-wrapper">
        <div class="formbold-form-wrapper">
          <h1>Garment Workers Productivity Prediction</h1>
          <img
src="https://static.vecteezy.com/system/resources/previews/005/260/243/original/i
l lustration-graphic-cartoon-character-of-business-analysis-vector.jpg"
alt="Worker Productivity" style="width: 100%; max-height: 200px; object-
fit:cover;marginbottom: 30px;margin-top: 20px;">
          <form action="/predict" method="post">
              <div class="formbold-input-group">
                <label for="Quarter" class="formbold-form-label">Quarter:</label>
                <input
                    type="number"
name="Quarter"
id="Quarter"
placeholder="Enter quarter"
class="formbold-form-input"
                    required
                />
              </div>

              <div class="formbold-input-group">
                  <label for="Department" class="formbold-
formlabel">Department:</label>
                  <input
type="number"
name="Department"
id="Department"
placeholder="Enter department"
class="formbold-form-input"
                    required
```

```
                    />
                </div>

                <div class="formbold-input-group">
                    <label for="Day of the week" class="formbold-form-label">Day of
the week:</label>
                    <input
type="number"                        name="Day
of the week"                        id="Day of
the week"
placeholder="Enter day"
class="formbold-form-input"
                        required
                    />
                </div>
```

```
            <div class="formbold-input-group">
              <label for="Team Number" class="formbold-form-label">Team
Number:</label>
              <input
type="number"                name="Team
Number"              id="Team Number"
placeholder="Enter                team"
class="formbold-form-input"
                required
              />
            </div>

            <div class="formbold-input-group">
            <label for="Time Allocated" class="formbold-form-
label">Time
Allocated:</label>
            <input
              type="tel"
              name="Time Allocated"
id="Time Allocated"
              placeholder="Enter Time Allocated"
class="formbold-form-input"
              required
            />
          </div>


          <div class="formbold-input-group">
            <label for="unfinished items" class="formbold-form-label">unfinished
items:</label>
            <input
              type="tel"
```

```
                name="Unfinished Items" id="Unfinished
                Items"
                placeholder="Enter unfinished items"
          </div>

          <div class="formbold-input-group">
             <label for="Over time" class="formbold-form-label">Over
Time:</label>
             <input                    type="tel"
name="Over time"                   id="Over
time"                placeholder="Enter over
time"                  class="formbold-form-
input"
                required
             />
          </div>

          <div class="formbold-input-group">
             <label for="Incentive" class="formbold-form-
label">Incentive:</label>
             <input                 type="tel"
name="Incentive"
id="Incentive"
placeholder="Enter incentive"
class="formbold-form-input"
                required
             />
          </div>
       <div class="formbold-input-group">
             <label for="Idle Time" class="formbold-form-label">Idle
Time:</label>
             <input
```

```
                type="tel"
               name="Idle Time" id="Idle
               Time" placeholder="Enter
               idletime"           </div>
      <div class="formbold-input-group">
             <label for="Idle Men" class="formbold-form-label">Idle Men:</label>
             <input                  type="tel"
name="Idle Men"                  id="Idle
Men"                  placeholder="Enter
incentive"                  class="formbold-
form-input"
                required
             />
          </div>
      <div class="formbold-input-group">
             <label for="Style Change" class="formbold-form-label">Style
Change:</label>
             <input
                type="tel"
name="Style Change"
id="Style Change"
placeholder="Enter incentive"
class="formbold-form-input"
                required
             />
          </div>
        <div class="formbold-input-group">
             <label for="Number of Workers" class="formbold-form-
label">Number of Workers:</label>
             <input
                type="tel"
```

name="Number of Workers" id="Number of Workers" placeholder="Enter incentive"

```
            </div>

                  <div>               <input type="submit"
 value="Predict" class="formbold-btn">
            </div>
        <form action="productivity.html" method="post">
 </div></div>
```

Result Page

```
         <!DOCTYPE html>
<html>
   <head>
       <title>Garment Worker productivity prediction</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/boo
tstrap.min.css" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP
+VmmDGMN5t9UJ0Z" crossorigin="anonymous">
   <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+Ibb
VYUew+OrCXaRkfj" crossorigin="anonymous"></script>
   <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/pop
per.min.js" integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6
BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>
   <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootst
rap.min.js" integrity="sha384-
```

```
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/K
UEfYiJOMMV+rV" crossorigin="anonymous"></script>

        <link rel="stylesheet" href="{{ url_for('static',
filename='assets/css/predict.css') }}">
        <meta name="viewport" content="width=device-
width,initial-scale=1.0">
<style>
      .h{
         background-
image:url("https://res.cloudinary.com/dcy90slhx/image/upload/v
1718992828/garment14_asuk6i.jpg");
height:100vh;
width:100vw;
background-size:cover;
}
.h1{
color:red;
}
.lay{
height:35vh;
width:80vw;
background-color:#ffffff;
border-top-left-radius:15px;
border-top-right-radius:15px;
border-bottom-left-radius:15px;
border-bottom-left-radius:15px;
padding:25px;
margin:25px;
}
.head{
```

```
color:yellow;
font-size:30px;
text-align:center;
}
.image{
        height:100px;
        width:100px;
        border-radius: 15px;
    }
    .h2{
        font-size:16px;
        font-weight: bold;

    }

</style>
  </head>
  <body>
<nav class="navbar navbar-expand-lg navbar-light bg-white ">
   <div class="container d-flex flex-row">
    <a class="navbar-brand" href="# ">
     <img

src="https://res.cloudinary.com/dcy90slhx/image/upload/v17189
92771/garment17_flgqap.webp"
        class="image"
     />
    </a>
    <div>
        <h1 class="h2">Unlocking Productivity Of Garment
Workers</h1>
```

```
        </div>
        <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse"
id="navbarNavAltMarkup">
          <div class="navbar-nav ml-auto">
            <a class="nav-link active" id="Home" href="{{
url_for('home') }}" >
              HOME
              <span class="sr-only">(current)</span>
            </a>
            <a class="nav-link" href="{{ url_for('happy1') }}"
id="navItem2">ABOUT</a>
            <a class="nav-link" href="{{ url_for('happy2') }}"
id="navItem3">DEPARTMENTS</a>
            <a class="nav-link" href="" id="navItem4">FAQS</a>
            <a class="nav-link" href="{{ url_for('happy')
}}">MAKE YOUR PREDICTION</a>

          </div>
        </div>
      </div>
    </nav>
<div class="h">
          <h1 class="head" id="name">Garment Worker
Productivity Prediction</h1>
          <div class="lay">
```

```
        <h1 class="h1" id="output">"Hence,based on calculation,
the predicted Garment worker is: "
{{ prediction_text }}</h1>
       </div>
</div>
    </body>
</html>
```

APP.PY

```
     from flask import Flask, render_template, request
 import pickle
 import numpy as np
 import pandas as pd

 app = Flask(_name_, template_folder='template')

 # Load the model
 model =
 pickle.load(open(r'C:\Users\srira\OneDrive\Desktop\akhil\prod
 uctivity4.pkl', 'rb'))

 @app.route('/')
 @app.route('/home')
 def home():
    return render_template('index.html')
 @app.route('/happy1')
 def happy1():
    return render_template('about1.html')

 @app.route('/happy2')
```

```
def happy2():
    return render_template('department1.html')



@app.route('/happy')
def happy():
    return render_template('predict.html')

@app.route('/predict', methods=['POST'])
def predict():
    quarter = int(request.form['Quarter'])

    # Map department to numerical values
    department = request.form['Department'].lower()
    if department == 'sewing':
        department = 1
    elif department == 'finishing':
        department = 0
    else:
        department = -1  # Handle unexpected input

    # Map day to numerical values
    day = request.form['Day of the week'].lower()
    day_map = {
        'monday': 0,
        'tuesday': 4,
        'wednesday': 5,
        'thursday': 3,
        'friday': 6,
        'saturday': 1,
```

```
        'sunday': 2
    }
    day = day_map.get(day, -1)  # Use -1 for unexpected input

    # Convert remaining form inputs to integers
    team_number = int(request.form['Team Number'])
    time_allocated = int(request.form['Time Allocated'])
    unfinished_items = int(request.form['Unfinished Items'])
    over_time = int(request.form['Over time'])
    incentive = int(request.form['Incentive'])
    idle_time = int(request.form['Idle Time'])
    idle_men = int(request.form['Idle Men'])
    style = int(request.form['Style Change'])
    workers = int(request.form['Number of Workers'])

    # Create a DataFrame with the input data
    input_data = pd.DataFrame([[quarter, department, day,
team_number, time_allocated, unfinished_items, over_time,
incentive, idle_time, idle_men, style, workers]],
                              columns=['quarter', 'department', 'day',
'team_number', 'time_allocated', 'unfinished_items', 'over_time',
'incentive', 'idle_time', 'idle_men', 'style_change',
'no_of_workers'])

    # Make the prediction
    input_data=np.array(input_data)
    prediction = model.predict(input_data)

    # Format the prediction
    prediction = (np.round(prediction[0], 4)) * 100
```

```
    text="Hence,based on calculation, the predicted Garment
worker is:"
    output = prediction
    print(output)
    return
render_template('productivity1.html',prediction_text=text +
str(output))

if _name_ == '_main_':
    app.run(debug=False,port=5000)
```

8.2 GitHub & Project Demo Link