PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

The garment industry stands as one of the global economy's pillars, and the productivity of garment workers is a decisive factor influencing the success and profitability of companies within this sector. In this project, our primary objective is to harness the power of machine learning to construct a robust predictive model capable of estimating garment worker productivity. By utilizing a comprehensive dataset encompassing diverse attributes related to garment production, we aim to uncover patterns and relationships that contribute to workforce efficiency.

1.2 Purpose :-

The primary purpose of this project is to develop a predictive model for garment worker productivity using machine learning techniques. The significance of this endeavor lies in its potential to revolutionize decision-making processes within the garment industry. The project serves several key purposes:

Optimizing Workforce Efficiency:

Identify and understand the factors that influence garment worker productivity.
Enable companies to optimize workforce efficiency by addressing specific challenges revealed through data analysis.

Cost Reduction:

Provide insights into areas where costs can be minimized without compromising productivity.
Assist in resource allocation, reducing unnecessary expenses and improving overall cost-effectiveness.

Strategic Decision-Making:

Empower decision-makers with a data-driven approach to strategic planning.
Enable the formulation of targeted interventions to address productivity bottlenecks.

Competitiveness Enhancement:

Enhance the competitiveness of companies within the garment industry.
Enable organizations to adapt quickly to market dynamics by improving operational efficiency.

Industry Advancement:

Contribute to the advancement of the garment industry through the application of cutting-edge technology.
Showcase the potential of machine learning in solving real-world challenges within manufacturing and supply chain management.

Human Resource Management:

Improve human resource management by understanding how various factors impact worker productivity.
Provide a basis for implementing policies that enhance employee satisfaction and engagement.

Supply Chain Optimization:

Facilitate better integration of production processes within the supply chain.
Contribute to the optimization of manufacturing workflows to meet demand efficiently.

Data-Driven Insights:

Promote a culture of data-driven decision-making within the garment industry.
Encourage the utilization of analytics to continuously refine and enhance production processes.

2. LITERATURE SURVEY

2.1 Existing Problem:-

The garment industry faces several existing challenges that impact the productivity of its workforce. Inefficient workflow processes limited predictive capabilities, and challenges in resource allocation contribute to suboptimal productivity and increased operational costs. Human resource management issues, including motivation and workload distribution, further compound these challenges. The industry often struggles with incomplete utilization of available data, hindering comprehensive insights into productivity drivers. Global competition and dynamic market demands add pressure, necessitating continuous adaptation for sustained competitiveness. Maintaining a high level of workforce satisfaction and engagement is crucial, as a disengaged workforce may compromise overall operational efficiency. Additionally, the industry's susceptibility to supply chain disruptions underscores the need for greater resilience and adaptability in addressing unknown challenges. Addressing these existing problems is imperative for the garment industry to enhance productivity, reduce costs, and maintain competitiveness in an evolving market.

2.2 References:-

1. https://rpubs.com/jialing2511/wqd7004occ2group7
2. https://www.researchgate.net/publication/369147058_Interpretable_Garment_Workers'_Productivity_Prediction_in_Bangladesh_Using_Machine_Learning_Algorithms_and_Explainable_AI

2.3 Problem Statement Definition:-

The problem at hand revolves around the suboptimal productivity and operational inefficiencies within the garment industry. Inadequate workflow processes, resource misallocation, and a lack of predictive capabilities hinder the industry's capacity to respond effectively to dynamic market demands. Challenges in human resource management, coupled with incomplete data utilization, contribute to increased operational costs. The overarching issue is the industry's struggle to adapt swiftly to changing circumstances, compromising its competitiveness. To address this problem, there is a critical need for a comprehensive solution that integrates advanced predictive modeling, optimized resource allocation, and enhanced human resource strategies to elevate overall productivity and ensure sustained competitiveness in the garment sector.

Key Points in the Problem Statement:

Inefficient Workflow Processes:

Workflow inefficiencies within the garment industry contribute to suboptimal productivity.
Lack of streamlined processes may impede the timely completion of tasks, affecting overall efficiency.

Suboptimal Resource Allocation:

Challenges in accurately allocating resources, including manpower and time, result in underutilization or overburdening of workers.
Inefficient resource allocation contributes to increased operational costs.

Lack of Predictive Capabilities:

The industry lacks sophisticated tools for predicting and mitigating factors affecting garment worker productivity.
Reactive decision-making is prevalent due to a deficiency in predictive analytics.

Human Resource Management Challenges:

Issues in managing human resources effectively, including problems related to motivation, workload distribution, and overall workforce satisfaction.
These challenges impact the well-being and productivity of garment workers.

Incomplete Data Utilization:

The industry is not fully leveraging available data on garment production.
Incomplete utilization of data prevents comprehensive insights into productivity drivers.

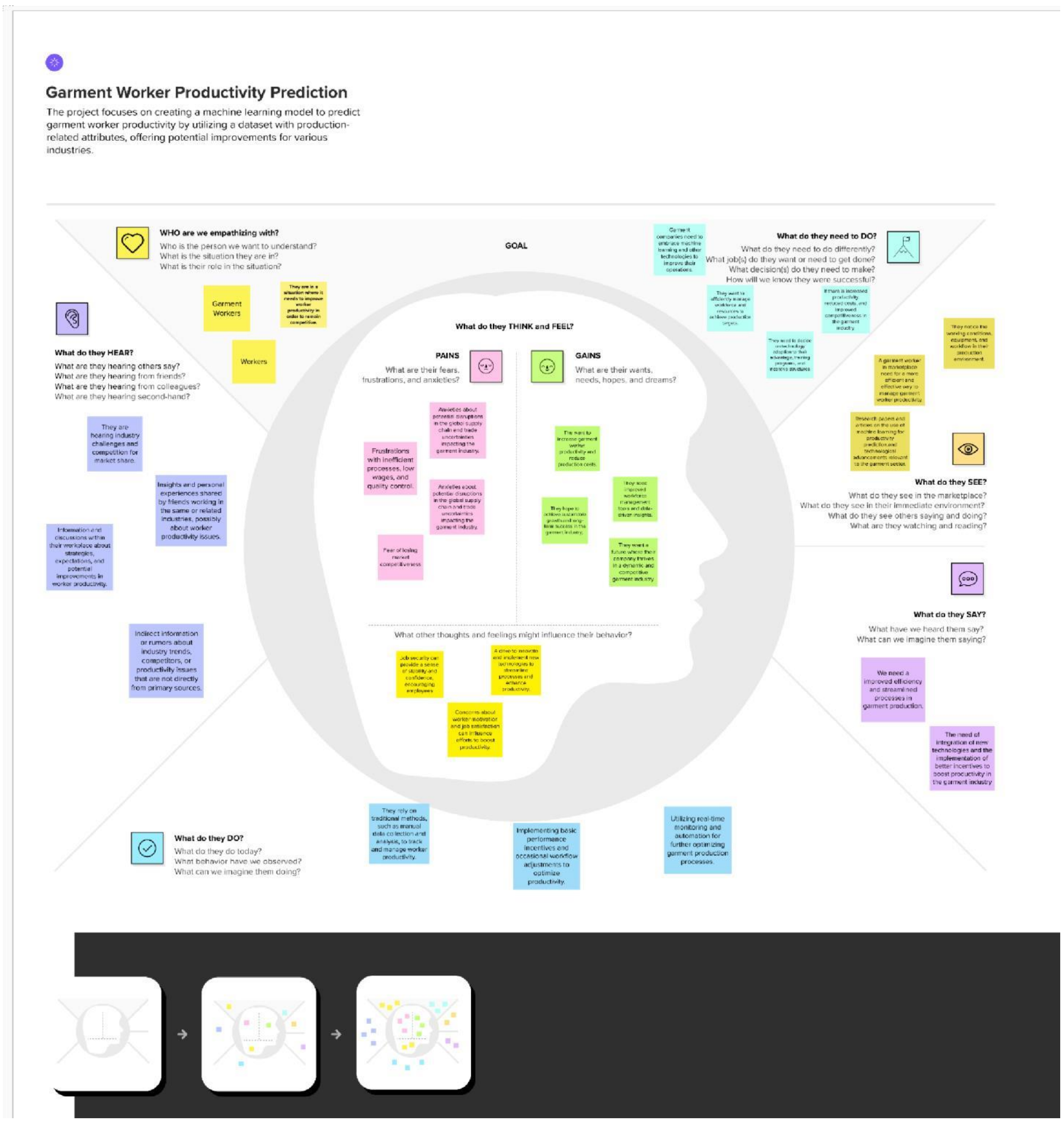
Ideation Phase
Empathize & Discover

Date	20 October 2023
Team ID	592923
Project Name	Garment Worker Productivity Prediction
Maximum Marks	4 Marks

Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.



Garment Worker Productivity Prediction

The project focuses on creating a machine learning model to predict garment worker productivity by utilizing a dataset with production-related attributes, offering potential improvements for various industries.

WHO are we empathizing with?

Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

Garment Workers

They are in a situation where it needs to improve worker productivity in order to remain competitive.

Workers

GOAL

Garment companies need to embrace machine learning and other technologies to improve their operations.

What do they need to DO?

What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

They want to efficiently manage workforce and resources to achieve production targets.

If there is increased productivity, reduced costs, and improved competitiveness in the garment industry.

They need to decide on technology adoption to their advantage, training programs, and incentive structures.

What do they HEAR?

What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

They are hearing industry challenges and competition for market share.

Insights and personal experiences shared by friends working in the same or related industries, possibly about worker productivity issues.

Information and discussions within their workplace about strategies, expectations, and potential improvements in worker productivity.

Indirect information or rumors about industry trends, competitors, or productivity issues that are not directly from primary sources.

What do they THINK and FEEL?

PAINS

What are their fears, frustrations, and anxieties?

Anxieties about potential disruptions in the global supply chain and trade uncertainties impacting the garment industry.

Frustrations with inefficient processes, low wages, and quality control.

Anxieties about potential disruptions in the global supply chain and trade uncertainties impacting the garment industry.

Fear of losing market competitiveness

GAINS

What are their wants, needs, hopes, and dreams?

They want to increase garment worker productivity and reduce production costs.

They need improved workforce management tools and data-driven insights.

They hope to achieve sustainable growth and long-term success in the garment industry.

They want a future where their company thrives in a dynamic and competitive garment industry

What other thoughts and feelings might influence their behavior?

Job security can provide a sense of stability and confidence, encouraging employees

A drive to innovate and implement new technologies to streamline processes and enhance productivity.

Concerns about worker motivation and job satisfaction can influence efforts to boost productivity.

They notice the working conditions, equipment, and workflow in their production environment.

A garment worker in marketplace need for a more efficient and effective way to manage garment worker productivity

Research papers and articles on the use of machine learning for productivity prediction and technological advancements relevant to the garment sector.

What do they SEE?

What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

What do they SAY?

What have we heard them say?
What can we imagine them saying?

We need a improved efficiency and streamlined processes in garment production.

The need of integration of new technologies and the implementation of better incentives to boost productivity in the garment industry

What do they DO?

What do they do today?
What behavior have we observed?
What can we imagine them doing?

They rely on traditional methods, such as manual data collection and analysis, to track and manage worker productivity.

Implementing basic performance incentives and occasional workflow adjustments to optimize productivity.

Utilizing real-time monitoring and automation for further optimizing garment production processes.

Link:
https://app.mural.co/t/hemesh4466/m/hemesh4466/1698167564025/0e6a9603eddb3497e306ce4fa939c34e4c1f6b0b?sender=u5e36ee1238fb2dd2d1428088
]

Ideation Phase
Brainstorm & Idea Prioritization Template

Date	20 October 2023
Team ID	592923
Project Name	Garment Worker Productivity Prediction
Maximum Marks	4 Marks

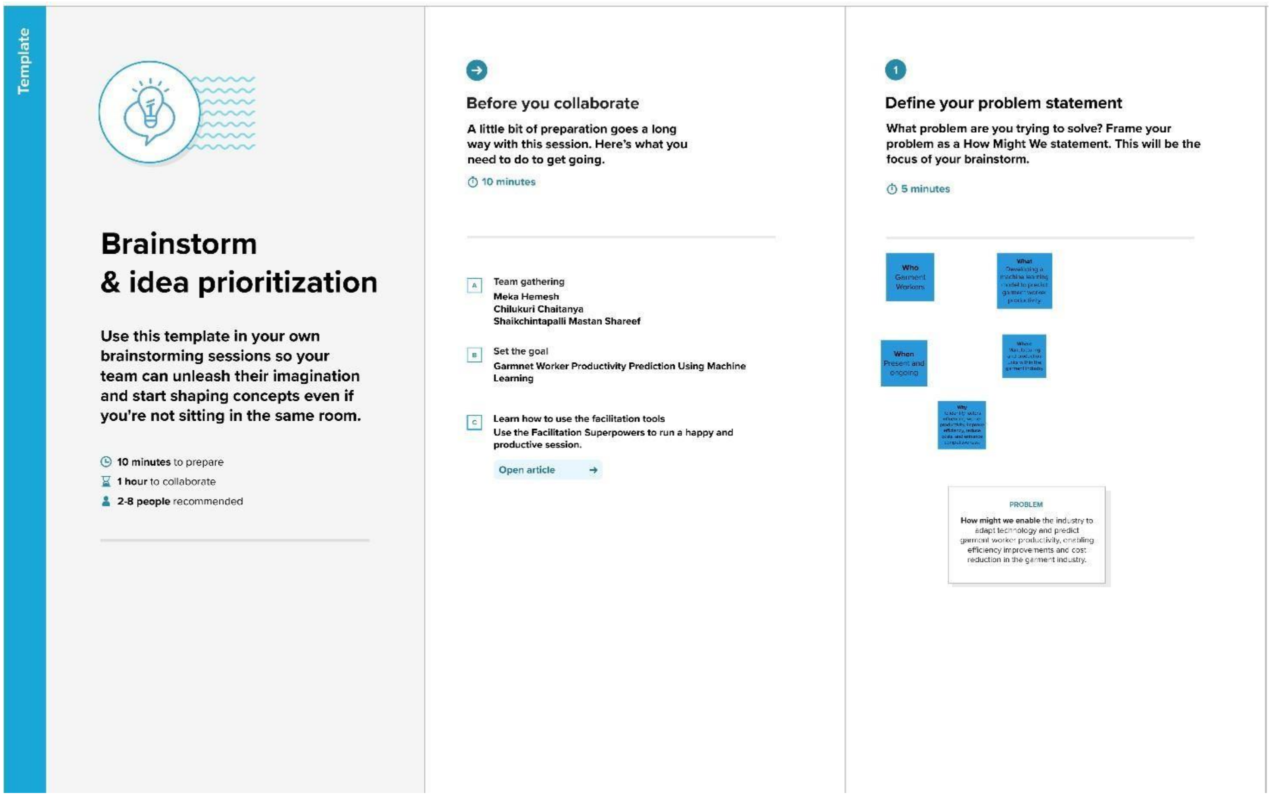
Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: https://www.mural.co/templates/empathy-map-canvas

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A. Team gathering
Meka Hemesh
Chilukuri Chaitanya
Shaikchintapalli Mastan Shareef

B. Set the goal
Garmnet Worker Productivity Prediction Using Machine Learning

C. Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

Who: Garment Workers

What: Developing a machine learning model to predict garment worker productivity

When: Present and ongoing

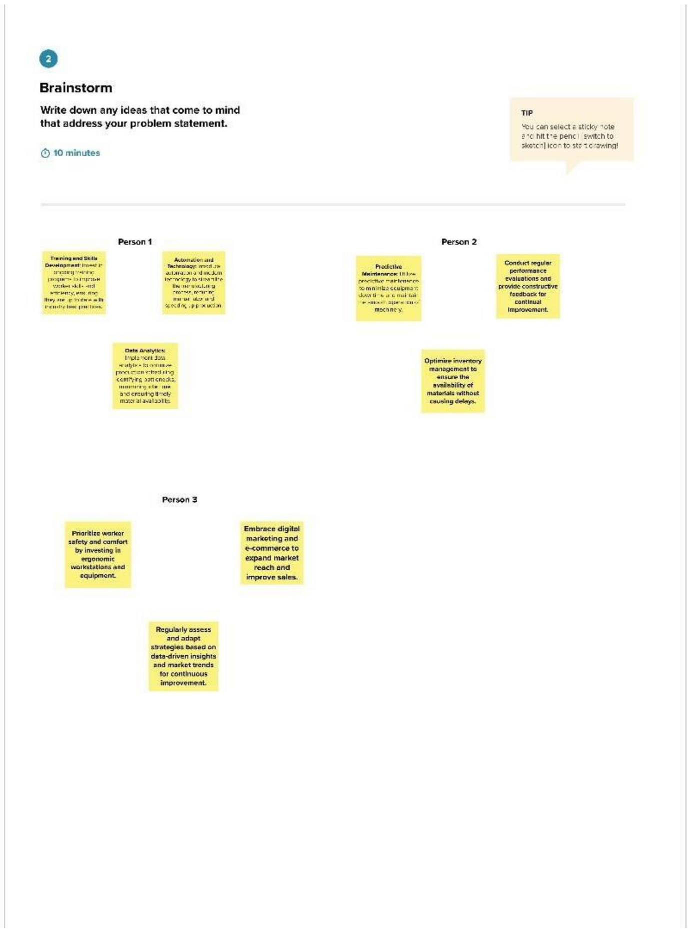
Where: Manufacturing and production units within the garment industry

Why: [illegible]

PROBLEM

How might we enable the industry to adapt technology and predict garment worker productivity, enabling efficiency improvements and cost reduction in the garment industry.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

Training and Skills Development: Invest in ongoing training programs to improve worker skills and efficiency, ensuring they are up to date with industry best practices.

Automation and Technology: Introduce automation and modern technology to streamline the manufacturing process, reducing manual labor and speeding up production.

Data Analytics: Implement data analytics to optimize production scheduling, identifying bottlenecks, minimizing idle time and ensuring timely material availability.

Person 2

Predictive Maintenance: Utilize predictive maintenance to minimize equipment downtime and maintain the smooth operation of machinery.

Conduct regular performance evaluations and provide constructive feedback for continual improvement.

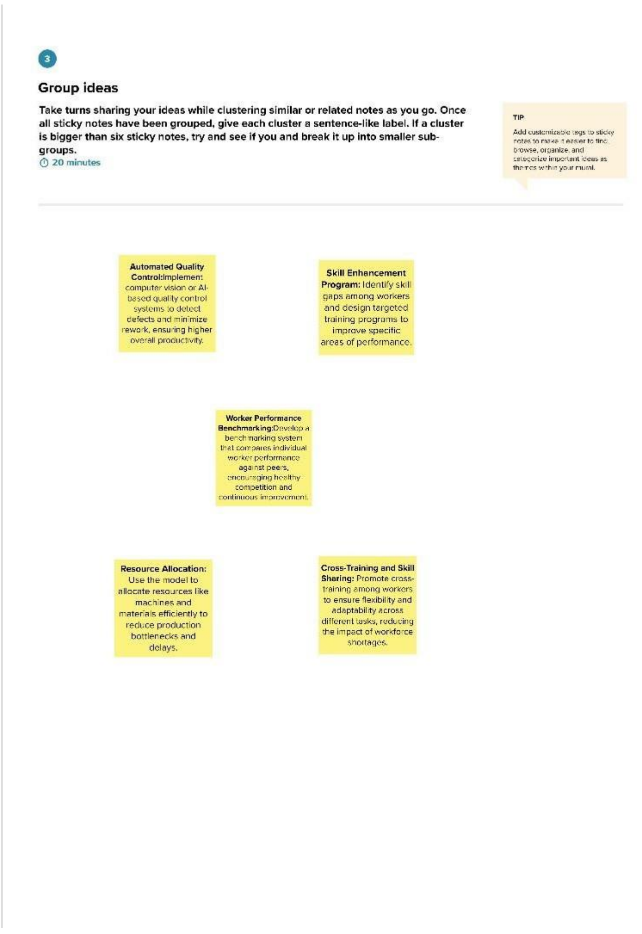
Optimize inventory management to ensure the availability of materials without causing delays.

Person 3

Prioritize worker safety and comfort by investing in ergonomic workstations and equipment.

Embrace digital marketing and e-commerce to expand market reach and improve sales.

Regularly assess and adapt strategies based on data-driven insights and market trends for continuous improvement.



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Automated Quality Control: Implement computer vision or AI-based quality control systems to detect defects and minimize rework, ensuring higher overall productivity.

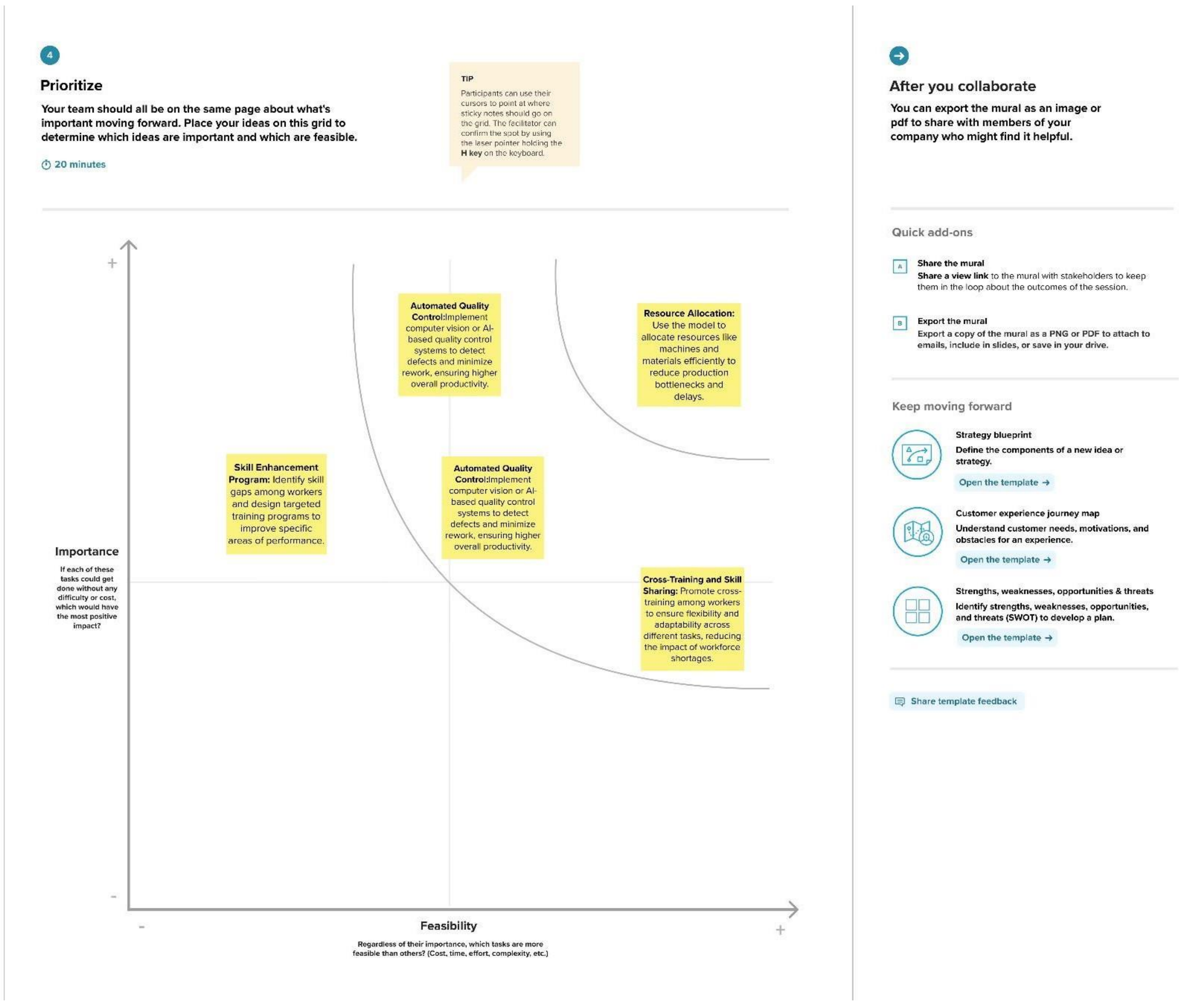
Skill Enhancement Program: Identify skill gaps among workers and design targeted training programs to improve specific areas of performance.

Worker Performance Benchmarking: Develop a benchmarking system that compares individual worker performance against peers, encouraging healthy competition and continuous improvement.

Resource Allocation: Use the model to allocate resources like machines and materials efficiently to reduce production bottlenecks and delays.

Cross-Training and Skill Sharing: Promote cross-training among workers to ensure flexibility and adaptability across different tasks, reducing the impact of workforce shortages.

Step-3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Automated Quality Control: Implement computer vision or AI-based quality control systems to detect defects and minimize rework, ensuring higher overall productivity.

Resource Allocation: Use the model to allocate resources like machines and materials efficiently to reduce production bottlenecks and delays.

Skill Enhancement Program: Identify skill gaps among workers and design targeted training programs to improve specific areas of performance.

Automated Quality Control: Implement computer vision or AI-based quality control systems to detect defects and minimize rework, ensuring higher overall productivity.

Cross-Training and Skill Sharing: Promote cross-training among workers to ensure flexibility and adaptability across different tasks, reducing the impact of workforce shortages.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

Link to our mural:-

https://app.mural.co/t/hemesh4466/m/hemesh4466/1698219675137/cbcfde60a5c6d9ad7e4d26a6ef7090f80c4ab5f a?sender=u8cfb11d97d35957cef318933

REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional requirements define the specific features and functionalities that a system, software, or product must possess to meet the needs of its users. These requirements outline what the system is expected to do, its capabilities, and how users will interact with it. In the context of developing a predictive model for garment worker productivity, functional requirements might include:

Data Collection and Input:

The system should be capable of collecting and inputting relevant data from various sources, including information on quarters, departments, days, team numbers, time allocated, unfinished items, overtime, incentives, idle time, idle men, style changes, and the number of workers.

Prediction Model:

Develop a machine learning model capable of predicting garment worker productivity based on the provided dataset.
The model should be able to generalize patterns and relationships within the data to make accurate predictions on new, unseen data.

User Interface:

Create a user-friendly interface that allows users to input new data for prediction and view the results.
Ensure that the interface is intuitive, facilitating ease of use for individuals with varying levels of technical expertise.

Scalability:

Design the system to be scalable, allowing for an increasing volume of data as the dataset grows over time.
Ensure that the model's performance remains consistent and efficient as the system handles larger datasets.

Integration:

Integrate the predictive model into existing workflows or systems within the garment industry seamlessly.
Ensure compatibility with data storage and retrieval systems commonly used in the industry.

Accuracy and Reliability:

Specify the level of accuracy required from the predictive model, considering the impact of inaccurate predictions on decision-making processes.
Ensure that the model is reliable and can consistently produce accurate results.

4.2 Non-Functional requirements

Non-functional requirements define the characteristics and constraints that the system must adhere to. These requirements focus on aspects such as performance, security, usability, and scalability. For a predictive model in the garment industry, non-functional requirements might include:

Performance:

Specify the response time for generating predictions to ensure timely decision-making.

Define the throughput, indicating the number of predictions the system should handle concurrently.

Usability:

Define user interface requirements to ensure a positive and efficient user experience.
Consider factors such as accessibility, responsiveness, and adaptability to different devices.
Security:

Implement measures to secure the data used for training the model and the predictions made by the system.
Specify user authentication and authorization protocols to control access to sensitive information.’

Scalability:

Define how the system will handle an increased volume of users or data.
Ensure that the system can scale up without compromising performance.

Reliability:

Specify the system's reliability requirements, including its uptime and the ability to recover from failures gracefully.
Implement mechanisms for data backup and recovery to prevent data loss.

Maintainability:

Define how updates, improvements, or modifications to the system will be managed.
Ensure that the system can be easily maintained and upgraded without significant disruptions.

Compliance:

Specify any regulatory or industry-specific compliance requirements that the system must adhere to.
Ensure that the system complies with data protection and privacy regulations.

Training and Documentation:

Develop comprehensive documentation and training materials for users and administrators.
Ensure that users have the necessary resources to understand and utilize the system effectively.

Project Design
Phase-ISolution
Architecture

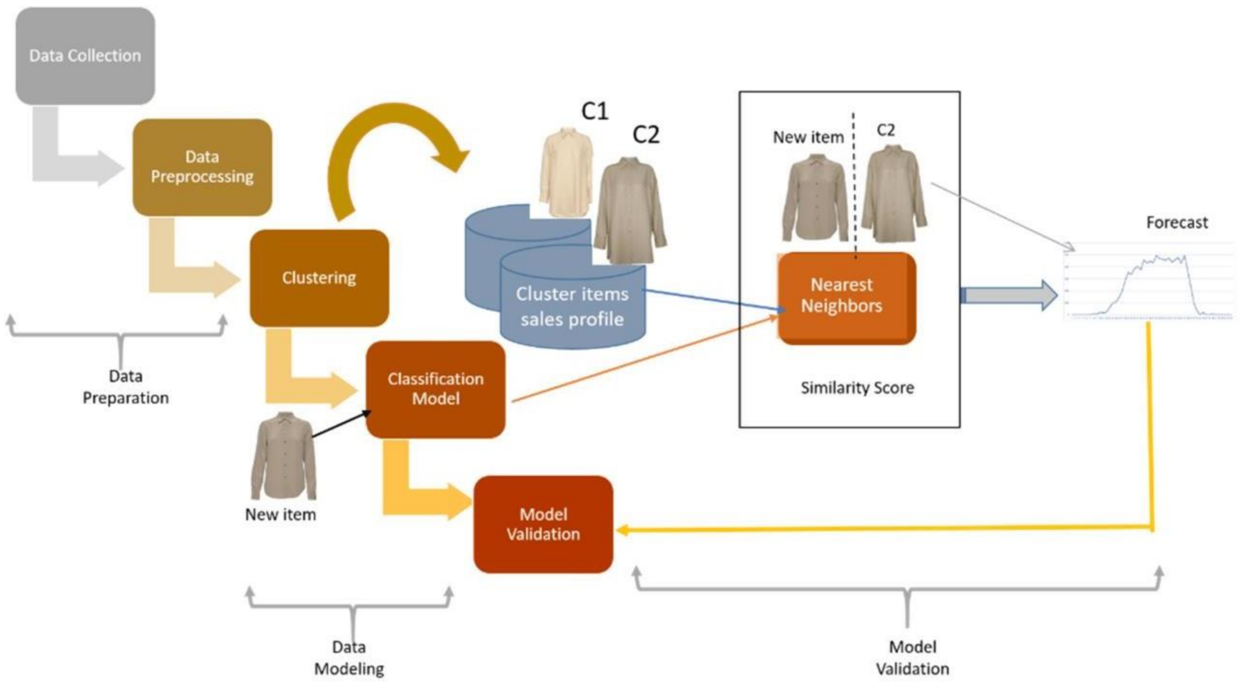
Date	31 October 2023
Team ID	Team-592923
Project Name	Garment Worker Productivity Prediction
Maximum Marks	4 Marks

Solution Architecture:

The Solution Architecture Diagram shows the key components of the garment worker productivity prediction model and how they interact with each other. The data collection and preparation component is responsible for collecting and preparing the data that will be used totrain the model. The feature engineering component is responsible for creating new features from the existing data and transforming the features to different scales.

The model selection and training component is responsible for selecting a machine learning algorithm and training the model on the prepared data. The model evaluation component is responsible for evaluating the performance of the trained model on a held-out test set. The
model deployment component is responsible for deploying the trained model to production sothat it can be used to predict worker productivity in real time.

Solution Architecture Diagram:



Key Components:

1. **Data Ingestion:** Ingest and preprocess the dataset containing various attributes ofgarment production.
2. **Feature Engineering:** Transform and engineer features to enhance the predictive powerof the model.
3. **Model Development:** Implement machine learning algorithms for training andprediction.
4. **Evaluation Module:** Assess the model's performance using various metrics.
5. **Interpretation & Insights:** Analyze and interpret the results to understand factorsaffecting productivity.
6. **Deployment:** Integrate the trained model into a usable system or application.

Key Techniques:

- **Machine learning algorithms:** There are a variety of machine learning algorithms that can be used to predict garment worker productivity. Some popular choices include linearregression, logistic regression, decision trees, random forests, and gradient boosting machines.

- **Feature engineering:** Feature engineering is a powerful technique that can be used to improve the performance of machine learning models. By creating new features from theexisting data, we can provide the model with more information to learn from.
- **Cross-validation:** Cross-validation is a technique that can be used to evaluate the performance of a machine learning model on new data. It involves splitting the trainingdata into multiple folds and training the model on each fold while evaluating it on the remaining folds.
- **Hyperparameter tuning:** Hyperparameter tuning is the process of finding the optimal values for the hyperparameters of a machine learning algorithm. Hyperparameters areparameters that control the training process and the behaviour of the algorithm.

Benefits:

- **Improved productivity:** By identifying the factors that affect worker productivity,companies can take corrective actions to improve efficiency and reduce costs.

- **Enhanced competitiveness:** By using a garment worker productivity prediction model,companies can better plan their production and supply chain, which can give them a competitive advantage in the market.

- **Improved decision-making**: A garment worker productivity prediction model can help companies make more informed decisions about labor allocation, resource allocation,and production planning.

- **Improve their environmental performance:** By reducing waste and optimizing production processes, companies can reduce their environmental impact.

- **Improve their supply chain management:** By better understanding the factors that affect worker productivity, companies can make more informed decisions about their supply chain partners.

- **Improve their brand reputation:** By demonstrating their commitment to social responsibility and ethical business practices, companies can improve their brand reputation and attract more customers.

- **Cost Reduction:** Optimizing resource allocation based on productivity factors.

In summary, our solution architecture integrates transfer learning and convolutional neural networks (CNNs) to create a Garment Worker Productivity Prediction. It operates with continuous learning and adaptation, guaranteeing real-time accuracy and making significant contributions to garment production industries.

Project Design Phase-II

Data Flow Diagram & User Stories

Date	31 october 2023
Team ID	592923
Project Name	Garment Worker Productivity Prediction
Maximum Marks	4 MARKS

Data Flow Diagram :

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat andclear DFD can depict the right

amount of the system requirement graphically. It shows how data enters and leaves the system, what changes theinformation, and where data is store

EXAMPLE :

Flow

Start

 |

 V

Data Collection

 |

 V

Data Preprocessing

 |

 v

Model Training

 |

 v

Model Evaluation

 |

 v

Productivity Prediction

 |

En

d

DATA FLOW DIAGRAM

Data Collection: Collect data on various attributes of garment production.

Data Preprocessing: Handle missing values, outliers, and perform feature engineering if necessary. Model Training: Train the machine learning model using the preprocessed dataset.

Model Evaluation: Evaluate the model’s performance using a separate test dataset.

Productivity Prediction: Use the trained model to predict worker productivity based on given features.

Customer dataset

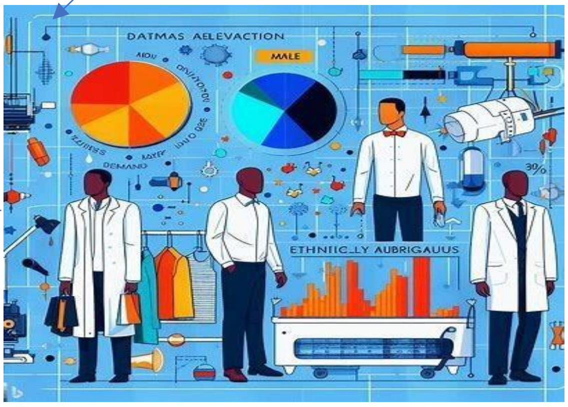
DATA SET

DATA – PREPROCESSING

MODEL TRAINING

PRODUCTIVITY PREDICTION

MODEL EVALUATION



USER STORIES

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story /Task	Acceptance criteria	Priority
Data scientist	Data preparation	DS-1	As a data scientist, I can preprocess the dataset to handle missing values and outliers	Dataset is clean and ready for model training	High
Data scientist	Model Training	DS-2	As a data scientist, I can Train a machine learning model using the preprocessed dataset	Model is trained and performance metrics are available	High
Data scientist	Model evaluation	DS-3	As a data scientist, I can Evaluate the model using a test dataset	Models performance on the test dataset is known	High
Company Management	Productivity Prediction	CM-1	As a company manager ,I can use the model to predict worker productivity based on given	Productivity prediction is available for decision making	High

			features		

Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	26 October 2023
Team ID	592923
Project Name	Garment Woker Productivity Prediction
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

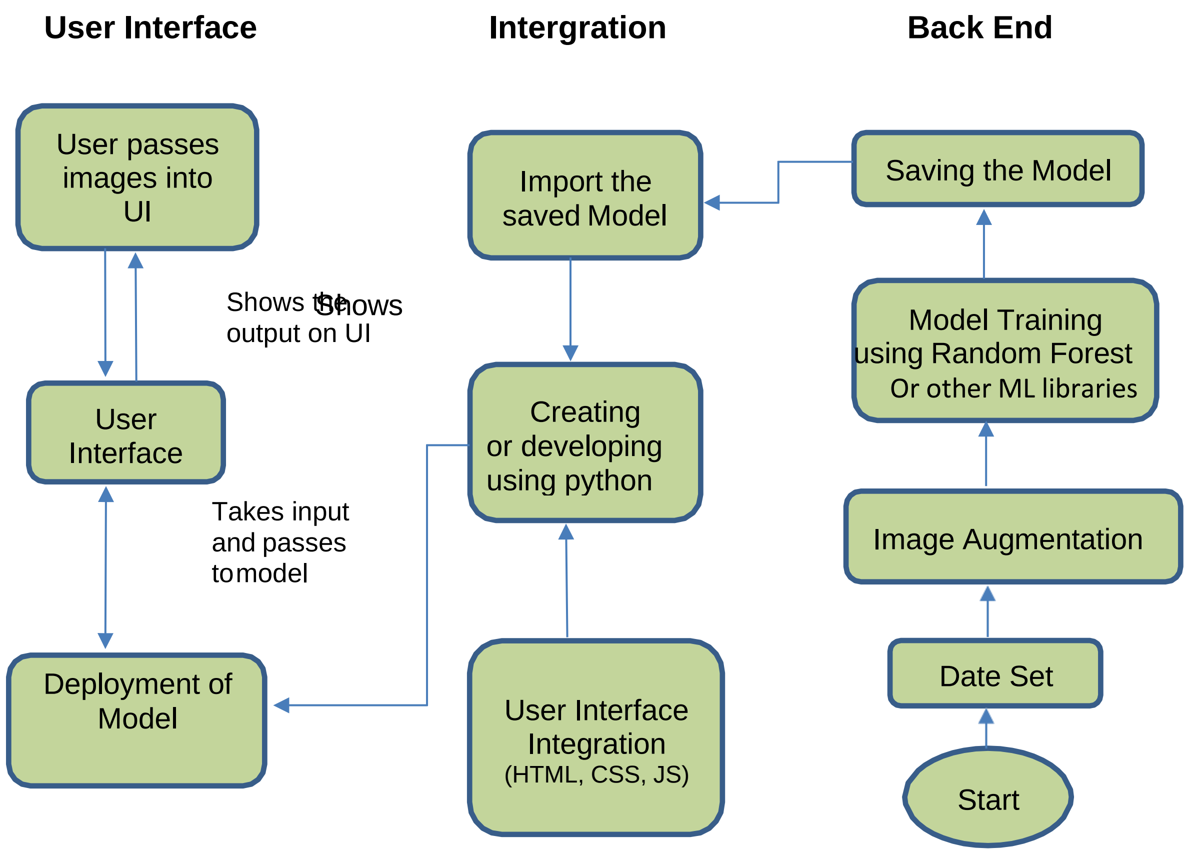

User Interface
Intergration
Back End
User passes images into UI
Shows the output on UI
Shows
User Interface
Takes input and passes to model
Deployment of Model
Import the saved Model
Creating or developing using python
User Interface Integration (HTML, CSS, JS)
Saving the Model
Model Training using Random Forest Or other ML libraries
Image Augmentation
Date Set
Start

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The graphical interface for user interaction and model input.	HTML, CSS, JavaScript
2.	Application Logic-1	Implements the application logic for user interactions, handles requests from the UI, and processes inputs.	Python ,JavaScript
3.	Database	Stores and manages data related to garmentproduction attributes.	Csv File and images
4.	File Storage	Manages storage of files, such as dataset files ormodel weights.	Local Filesystem ,Cloud Storage
5.	Frame Work	Provides tools and utilities for developing, training, and deploying machine learning models	Scikit-learn, TensorFlow, PyTorch, XGBoost
6.	Machine Learning Model	Implements a machine learning model for predicting garment worker productivity.	Scikit-learn, XGBoost, RandomForest, or other ML libraries

Table-2: Application Characteristics:

S.No	Component	Description	Technology
1.	Open-Source Frameworks	Utilizes frameworks that are open-source, fostering collaboration and flexibility.	Google Colab, Jupyter Notebook,VS Code
2.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	TensorFlow and Pytorch
3.	Availability	Ensures high availability with minimal downtime and robust fault tolerance.	Redundant servers, Failover mechanisms, Cloud-based infrastructure
4.	Performance	Optimized for efficient and responsive operation under varying workloads.	Caching mechanisms, Load balancing,Performance monitoring tools

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	31 October 2023
Team ID	Team-592923
Project Name	Garment Worker Productivity Prediction
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Project setup & Infrastructure	USN-1	Set up the development environment with the required tools and frameworks to start the garment worker productivity prediction project.	3	High	Hemesh
Sprint 1	Data Collection	USN-2	Gather data related to garment workers, including factors like working hours, machine usage, and other relevant parameters for training the prediction model.	2	High	Shareef
Sprint 2	Data Preprocessing	USN-3	Preprocess the collected data, handle missing values, scale features, and prepare it for training the productivity prediction model.	2	High	Hemesh
Sprint 2	Feature Engineering	USN-4	Explore and engineer relevant features that can contribute to the accuracy of the productivity prediction model.	3	High	Chaitanya
Sprint 3	Model Development	USN-5	Train a machine learning model using the preprocessed data to predict garment worker productivity based on the selected features.	3	High	Chaitanya
Sprint 3	Model Evaluation	USN-6	Evaluate the model's performance using appropriate metrics and fine-tune it for better accuracy.	2	Medium	Shareef

Sprint 4	Model Deployment & Integration	USN-7	Deploy the trained model as an API or service for predicting garment worker productivity. Integrate the model into a user-friendly interface for users to input relevant parameters and receive predictions.	2	Medium	Chaitanya
Sprint 5	Testing & Quality Assurance	USN-8	Conduct thorough testing of the model and interface to identify and address any issues. Optimize the model based on user feedback and testing results.	3	Medium	Hemesh

Project Tracker, Velocity & Burndown Chart: (4 Marks)

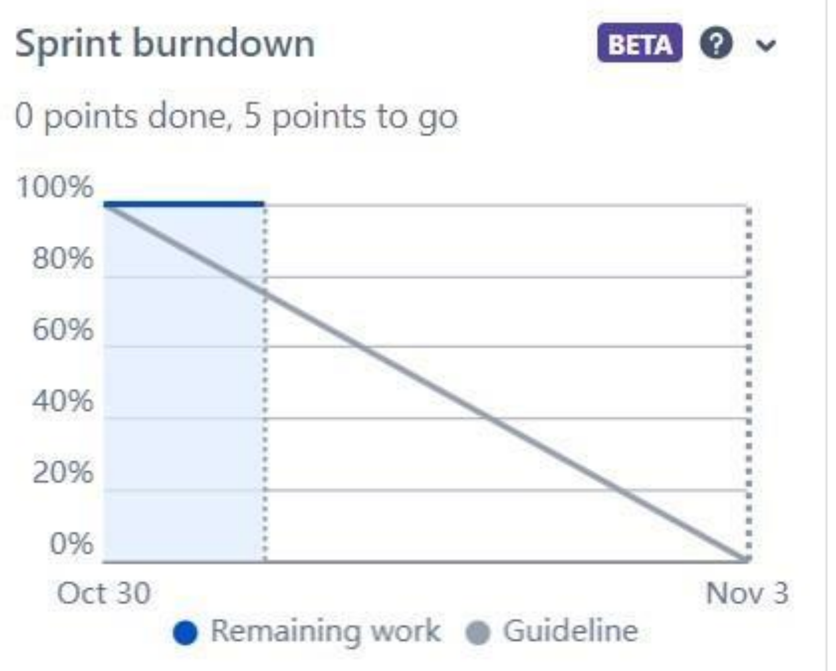
Sprint	Total Story Points	Duration - Sprint Start Date Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	5	3 Days 26 Oct 2023 29 Oct 2023	5	29 Oct 2023
Sprint-2	5	3 Days 29 Oct 2023 31 Oct 2023	5	31 Oct 2023
Sprint-3	5	2 Days 1 Nov 2023 3 Nov 2023	5	3 Nov 2023
Sprint-4	2	2 Days 3 Nov 2023 5 Nov 2023	2	5 Nov 2023
Sprint-5	3	1 Days 5 Nov 2023 6 Nov 2023	3	6 Nov 2023

Velocity:

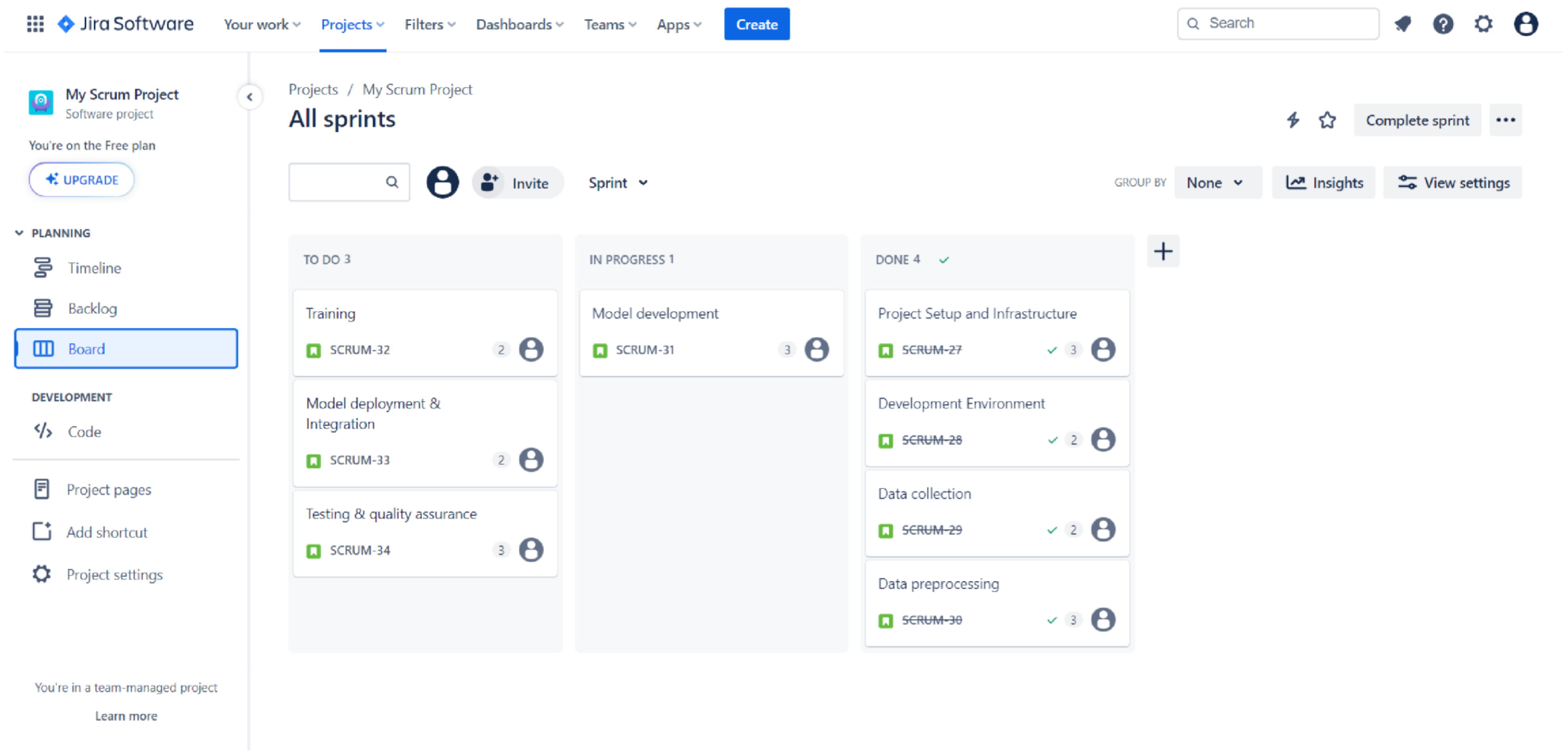
Average Velocity = Sprint Duration / Velocity = 11/5 = 2.2

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time.



Board Section



TO DO 3

- Training — SCRUM-32 — 2
- Model deployment & Integration — SCRUM-33 — 2
- Testing & quality assurance — SCRUM-34 — 3

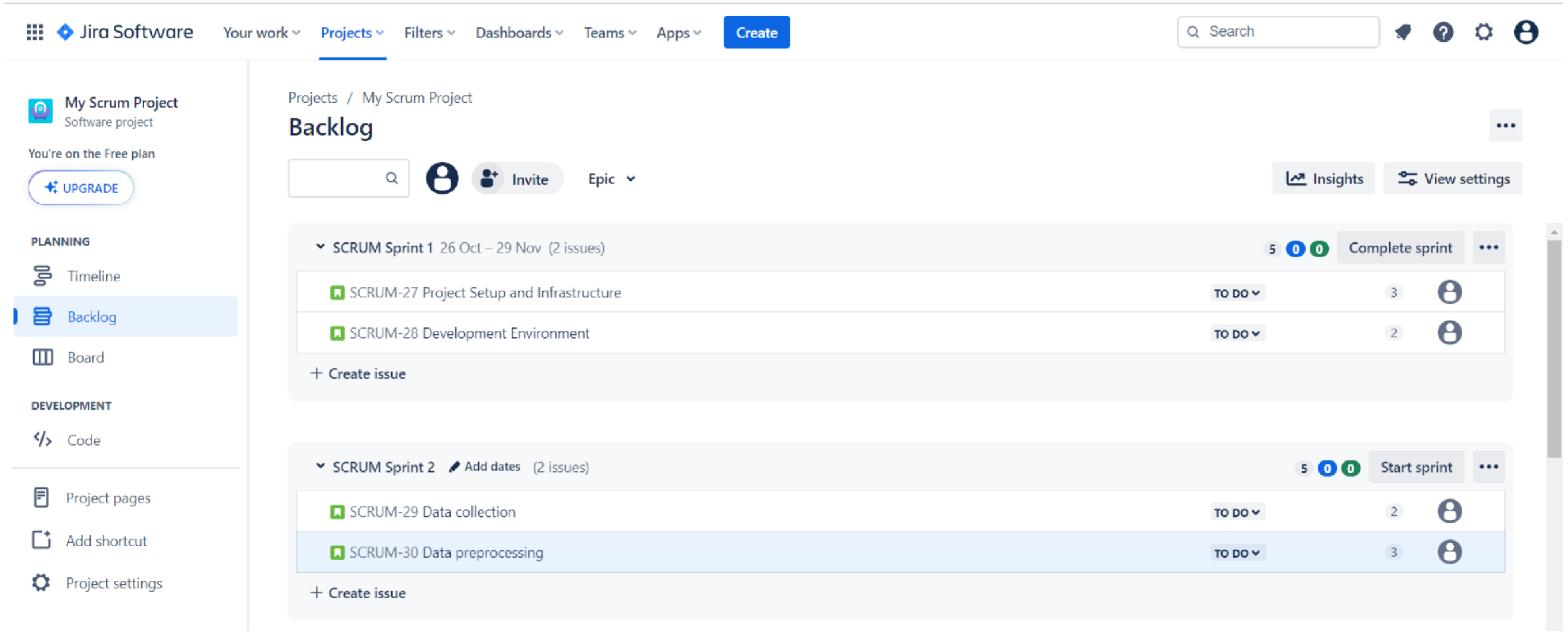
IN PROGRESS 1

- Model development — SCRUM-31 — 3

DONE 4 ✓

- Project Setup and Infrastructure — ~~SCRUM-27~~ ✓ 3
- Development Environment — ~~SCRUM-28~~ ✓ 2
- Data collection — ~~SCRUM-29~~ ✓ 2
- Data preprocessing — ~~SCRUM-30~~ ✓ 3

Backlog section


Jira Software · Your work · Projects · Filters · Dashboards · Teams · Apps · Create · Search

My Scrum Project
Software project
You're on the Free plan
UPGRADE

PLANNING
Timeline
Backlog
Board

DEVELOPMENT
Code

Project pages
Add shortcut
Project settings

Projects / My Scrum Project
Backlog

Invite · Epic · Insights · View settings

SCRUM Sprint 1 26 Oct – 29 Nov (2 issues) 5 0 0 Complete sprint

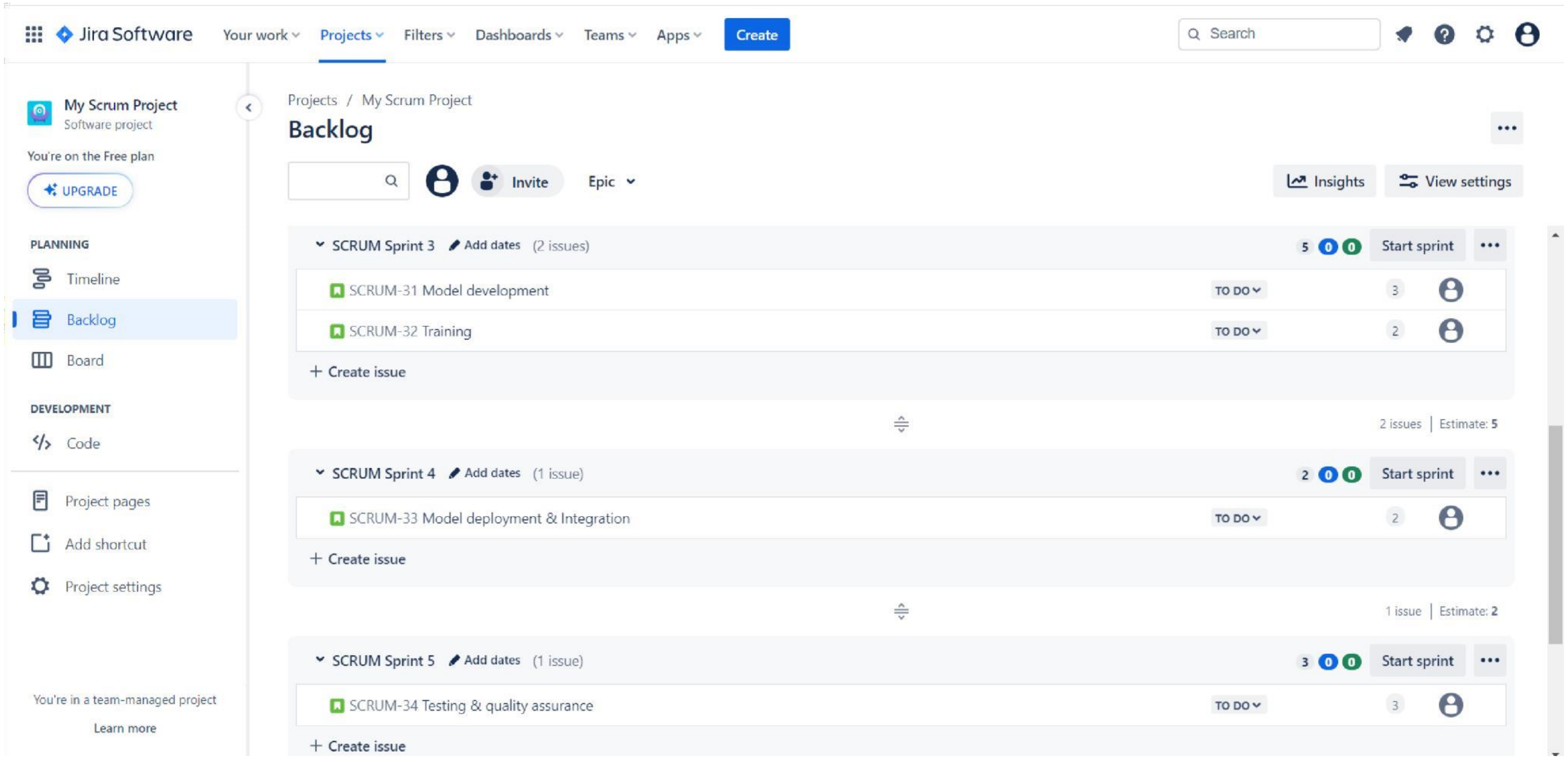
SCRUM-27 Project Setup and Infrastructure	TO DO	3	
SCRUM-28 Development Environment	TO DO	2	

+ Create issue

SCRUM Sprint 2 Add dates (2 issues) 5 0 0 Start sprint

SCRUM-29 Data collection	TO DO	2	
SCRUM-30 Data preprocessing	TO DO	3	

+ Create issue



Jira Software · Your work · Projects · Filters · Dashboards · Teams · Apps · Create · Search

My Scrum Project
Software project

You're on the Free plan
UPGRADE

PLANNING
Timeline
Backlog
Board

DEVELOPMENT
Code

Project pages
Add shortcut
Project settings

You're in a team-managed project
Learn more

Projects / My Scrum Project

Backlog

Invite · Epic · Insights · View settings

SCRUM Sprint 3 · Add dates (2 issues) · 5 0 0 · Start sprint

Issue	Status	Estimate
SCRUM-31 Model development	TO DO	3
SCRUM-32 Training	TO DO	2

+ Create issue

2 issues | Estimate: 5

SCRUM Sprint 4 · Add dates (1 issue) · 2 0 0 · Start sprint

Issue	Status	Estimate
SCRUM-33 Model deployment & Integration	TO DO	2

+ Create issue

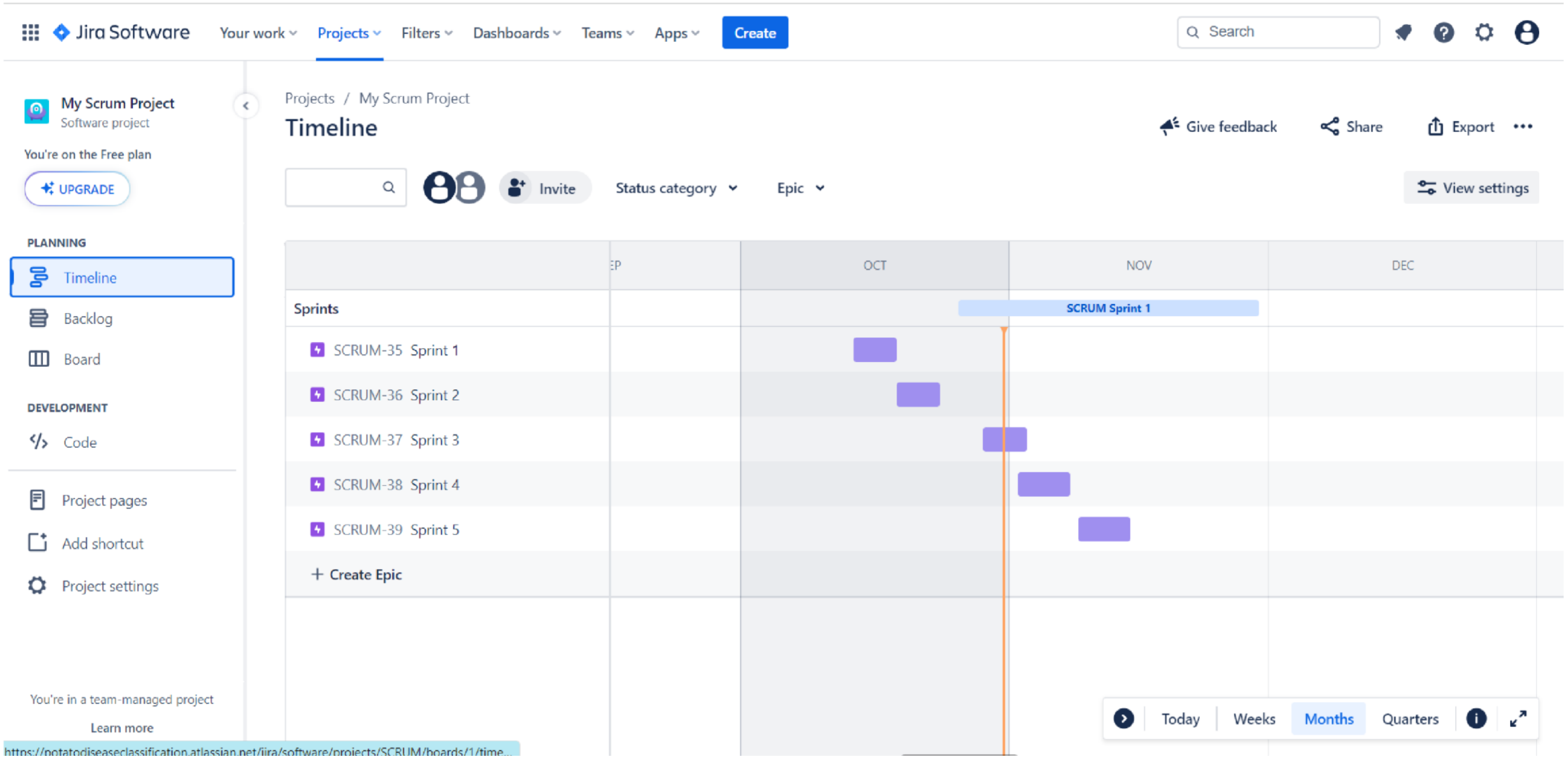
1 issue | Estimate: 2

SCRUM Sprint 5 · Add dates (1 issue) · 3 0 0 · Start sprint

Issue	Status	Estimate
SCRUM-34 Testing & quality assurance	TO DO	3

+ Create issue

Time Line



Timeline

	EP	OCT	NOV	DEC
Sprints			SCRUM Sprint 1	
SCRUM-35 Sprint 1				
SCRUM-36 Sprint 2				
SCRUM-37 Sprint 3				
SCRUM-38 Sprint 4				
SCRUM-39 Sprint 5				
+ Create Epic				

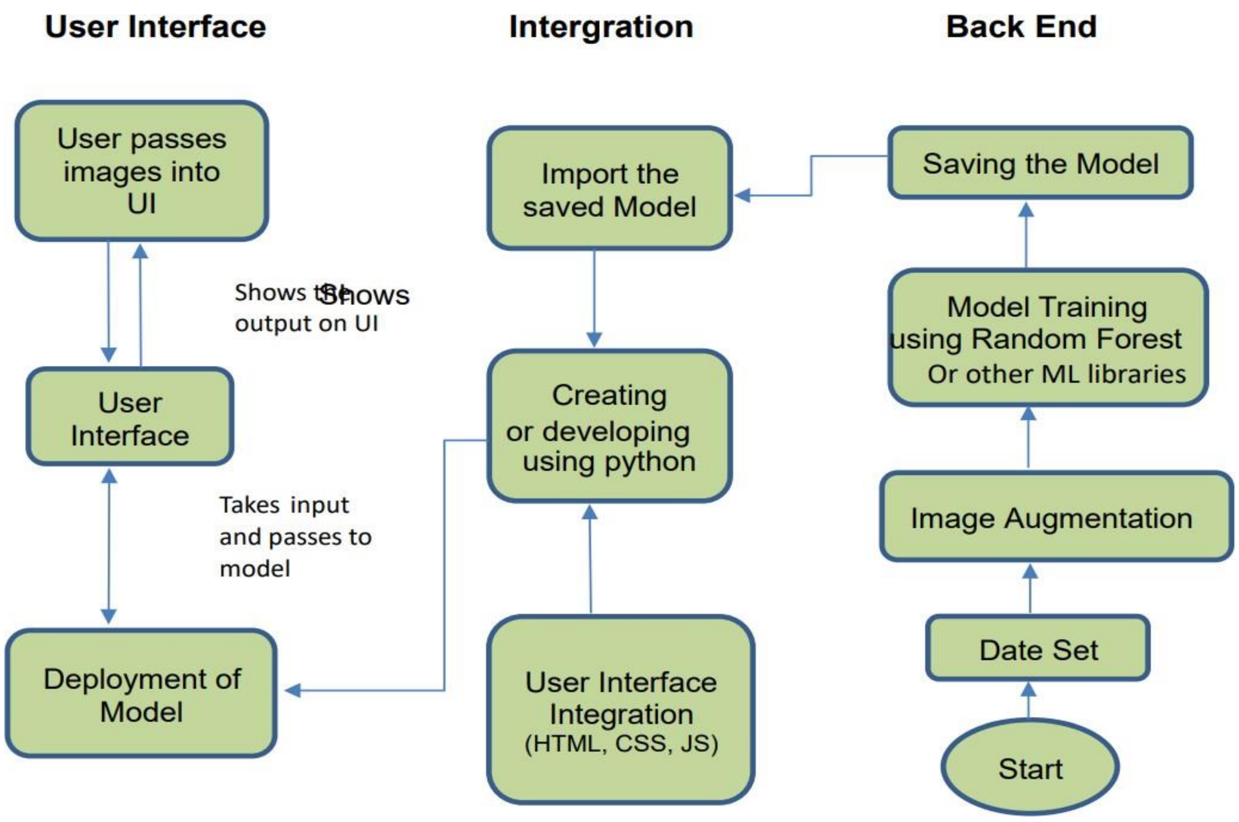
https://potatodiseaseclassification.atlassian.net/jira/software/projects/SCRUM/boards/1/time...

Garment Worker Productivity Prediction using Machine Learning

The garment industry is one of the largest industries in the world, and garment worker productivity is a crucial factor in determining the success and profitability of a company. In this project, we aim to develop a machine learning model that predicts the productivity of garment workers based on a given set of features. Our dataset contains information on various attributes of garment production, including the quarter, department, day, team number, time allocated, unfinished items, over time, incentive, idle time, idle men, style change, number of workers, and actual productivity. We will use this dataset to train and evaluate our predictive model.

The development of an accurate garment worker productivity prediction model using machine learning can have significant implications in various domains, including manufacturing, human resources, and supply chain management. This model can help companies identify the factors that affect worker productivity and take corrective actions to improve efficiency, reduce costs, and enhance their competitiveness in the market.

Technical Architecture:


User Interface
Intergration
Back End
User passes images into UI
Shows the output on UI
Shows
User Interface
Takes input and passes to model
Deployment of Model
Import the saved Model
Creating or developing using python
User Interface Integration (HTML, CSS, JS)
Saving the Model
Model Training using Random Forest Or other ML libraries
Image Augmentation
Date Set
Start

Project Flow:

- User interacts with the UI to enter the input.
- Entered input is analyzed by the model which is integrated.
- Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

- Define Problem / Problem Understanding
    - Specify the business problem
    - Business requirements
    - Literature Survey
    - Social or Business Impact

- Data Collection & Preparation
    - Collect the dataset
    - Data Preparation

- Exploratory Data Analysis
    - Descriptive statistical
    - Visual Analysis

- Collect the dataset
    - Training the model in multiple algorithms
    - Testing the model

- Performance Testing & Hyperparameter Tuning
    - Testing model with multiple evaluation metrics
    - Comparing model accuracy before & after applying hyperparameter tuning

- Model Deployment
    - Save the best model
    - Integrate with Web Framework

- Project Demonstration & Documentation
    - Record explanation Video for project end to end solution
    - Project Documentation-Step by step project development procedure

Prior Knowledge:

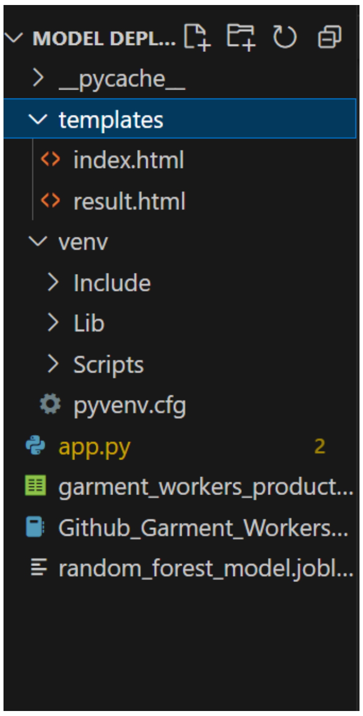
You must have prior knowledge of following topics to complete this project. ● ML Concepts

- Linear Regression: - https://www.javatpoint.com/linear-regression-in-machine-learning
- Decision Tree Regressor: - https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm
- Random Forest Regressor: - https://www.javatpoint.com/machine-learning-random-forest-algorithm
- Gradient Boosting Regressor: - https://www.javatpoint.com/gbm-in-machine-learning
- Xtreme Gradient Boost Regressor: - https://www.javatpoint.com/xgboost-ml-model-in-python
- Bagging Regressor: - https://www.javatpoint.com/bagging-vs-boosting o

Boosting Regressor: - https://www.javatpoint.com/bagging-vs-boosting ● Flask Basics: - https://www.youtube.com/watch?v=lj4I_CvBnt0

Project Structure:

Create a smart watch folder which contains files as shown below:


```
MODEL DEPL...
  > __pycache__
  v templates
      index.html
      result.html
  v venv
    > Include
    > Lib
    > Scripts
      pyvenv.cfg
    app.py  2
    garment_workers_product...
    Github_Garment_Workers...
    random_forest_model.jobl...
```

- We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.

- random_forest_model.joblib is where our machine learning model is saved. Further we will use this model for flask integration.

<u>Milestone 1: Define Problem / Problem Understanding</u>

Activity 1: Specify the business problem

Refer Project Description

Activity 2: Business requirements

To ensure that the garment worker productivity prediction model meets business requirements and can be deployed for public use, it should follow the following rules and requirements:

- Accuracy: The model should have a high level of accuracy in predicting worker productivity, with a low margin of error. This is crucial to ensure that the predictions are reliable and trustworthy.
- Privacy and security: The model should be developed in accordance with privacy and security regulations to protect user data. This includes ensuring that sensitive data is stored securely and implementing proper data access controls.
- Interpretability: The model should be interpretable, meaning that the predictions can be explained and understood by the end-users. This is important to build trust in the model and to allow users to make informed decisions based on the predictions.
- User interface: The model should have a user-friendly interface that is easy to use and understand. This is important to ensure that the model can be deployed for public use, even by individuals who may not have technical expertise.

Activity 3: Literature Survey

A literature survey for a garment worker productivity prediction project would involve researching and reviewing existing studies, articles, and other publications related to machine learning in the field of manufacturing and workforce management. The survey would aim to gather information on current methods for predicting productivity in the garment industry, including feature selection, data pre-processing, and machine learning algorithms used for prediction.

The survey would also examine any gaps in knowledge and research opportunities in the field of garment worker productivity prediction, including the use of new machine learning techniques, such as reinforcement learning, and the integration of other data sources, such as wearable technology and environmental sensors.

Activity 4: Social or Business Impact

Social Impact: The garment worker productivity prediction model has the potential to improve the lives of garment workers by promoting fair and efficient workforce management practices. The model can identify factors that affect worker productivity and suggest ways to improve working conditions, incentive schemes, and training programs. This can lead to higher job satisfaction, better pay, and improved working conditions for garment workers.

Business Impact: The garment worker productivity prediction model can have significant implications for the garment manufacturing industry. The model can help manufacturers optimize their workforce management strategies, reduce idle time, and increase productivity. This can lead to higher profits, lower production costs, and improved quality of goods produced. Additionally, the model can assist in identifying areas for process improvement, such as reducing rework and improving supply chain efficiency.

Milestone 2: Data collection & Preparation

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

Activity 1: Collect the dataset

There are many popular open sources for collecting the data. E.g., kaggle.com, UCI repository, etc.

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

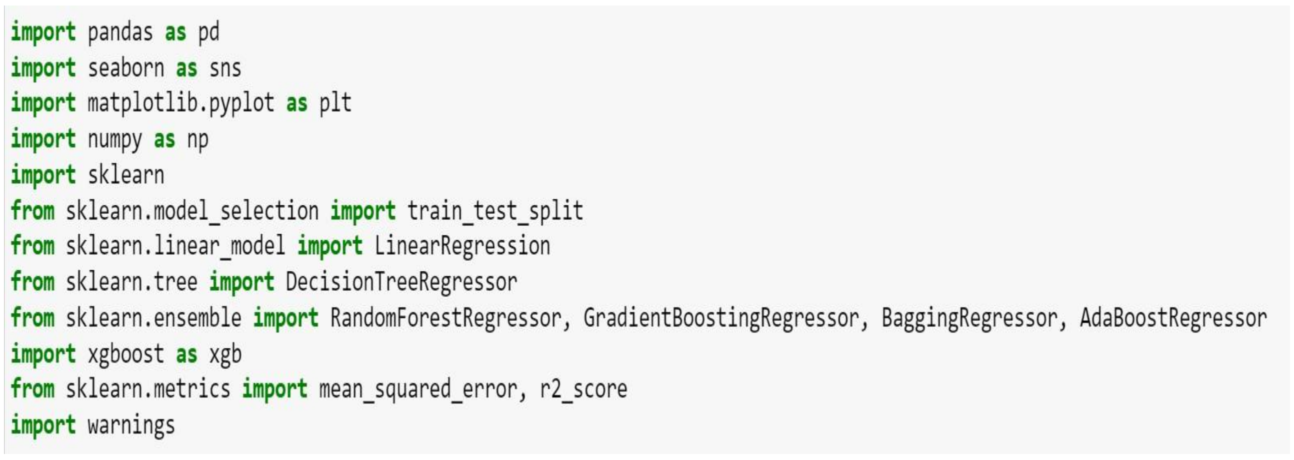
Link:
https://archive.ics.uci.edu/dataset/597/productivity+prediction+of+garment+employees

Download the dataset from the above link. Let us understand and analyze the dataset properly using visualization techniques.

Note: There are several approaches for understanding the data. But we have applied some of it here. You can also employ a variety of techniques.

Activity 1.1: Importing the libraries

Import the libraries required for this machine learning project, as shown in the image below.



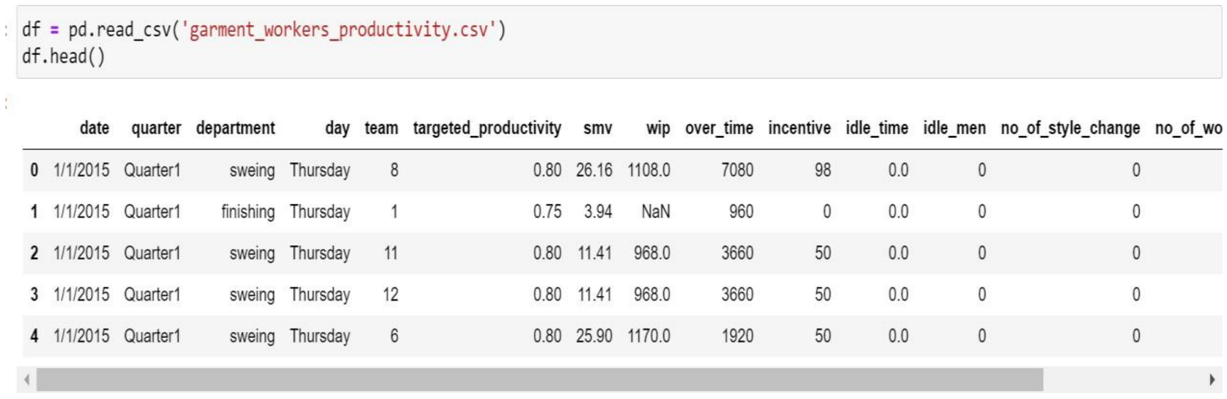
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor, BaggingRegressor, AdaBoostRegressor
import xgboost as xgb
from sklearn.metrics import mean_squared_error, r2_score
import warnings
```

Activity 1.2: Read the Dataset

Our dataset format could be in .csv, excel files,.txt, .json, and so on. With the help of pandas, we can read the dataset.

Since our dataset is a csv file, we use read_csv() which is pandas function to read the dataset. As a parameter we have to give the directory of the csv file.

df.head() will display first 5 rows of the dataset.



```python
df = pd.read_csv('garment_workers_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_wo
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
```

Activity 2: Data Preparation

Data preparation, also known as data preprocessing, is the process of cleaning, transforming, and organizing raw data before it can be used in a data analysis or machine learning model.

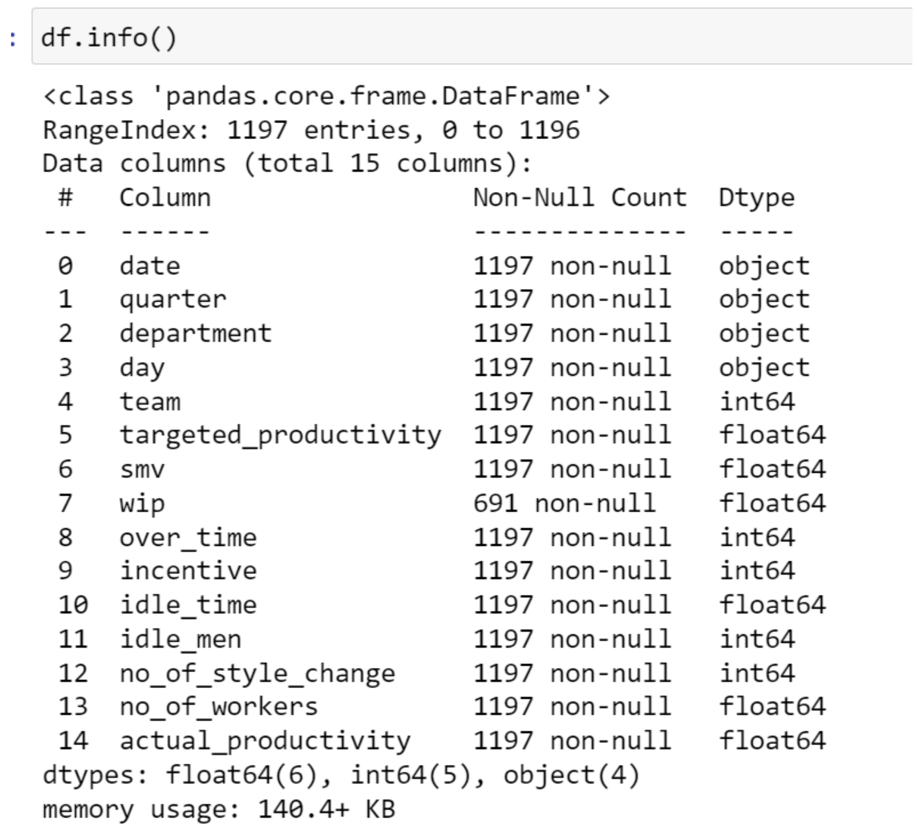
The activity include following steps:

- removing missing values
- handling outliers
- encoding categorical variables
- normalizing data

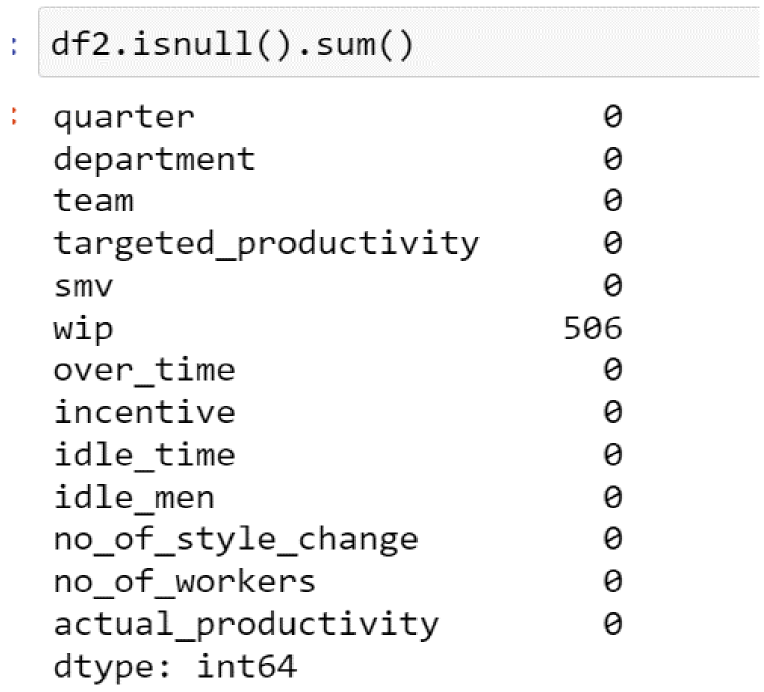
Note: These are general steps to take in pre-processing before feeding data to machine learning for training. The pre-processing steps differ depending on the dataset. Depending on the condition of your dataset, you may or may not have to go through all these steps.

Activity 2.1: Handling Missing Values

Let's first figure out what kind of data is in our columns by using df.info(). We may deduce from this that our columns include data of the types "object", "float64" and “int64”.



```
: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   date                   1197 non-null   object 
 1   quarter                1197 non-null   object 
 2   department             1197 non-null   object 
 3   day                    1197 non-null   object 
 4   team                   1197 non-null   int64  
 5   targeted_productivity  1197 non-null   float64
 6   smv                    1197 non-null   float64
 7   wip                    691 non-null    float64
 8   over_time              1197 non-null   int64  
 9   incentive              1197 non-null   int64  
 10  idle_time              1197 non-null   float64
 11  idle_men               1197 non-null   int64  
 12  no_of_style_change     1197 non-null   int64  
 13  no_of_workers          1197 non-null   float64
 14  actual_productivity    1197 non-null   float64
dtypes: float64(6), int64(5), object(4)
memory usage: 140.4+ KB
```



```
: df2.isnull().sum()

: quarter                    0
  department                 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
```

This line of code is used to count the number of missing or null values in a pandas DataFrame. It returns a list of the total number of missing values in each column of the DataFrame.

```
df3 = df2.fillna({
        'wip': 0,
        })
```

This line of code fills the missing values in the “unfinished_items” column with the column mean.

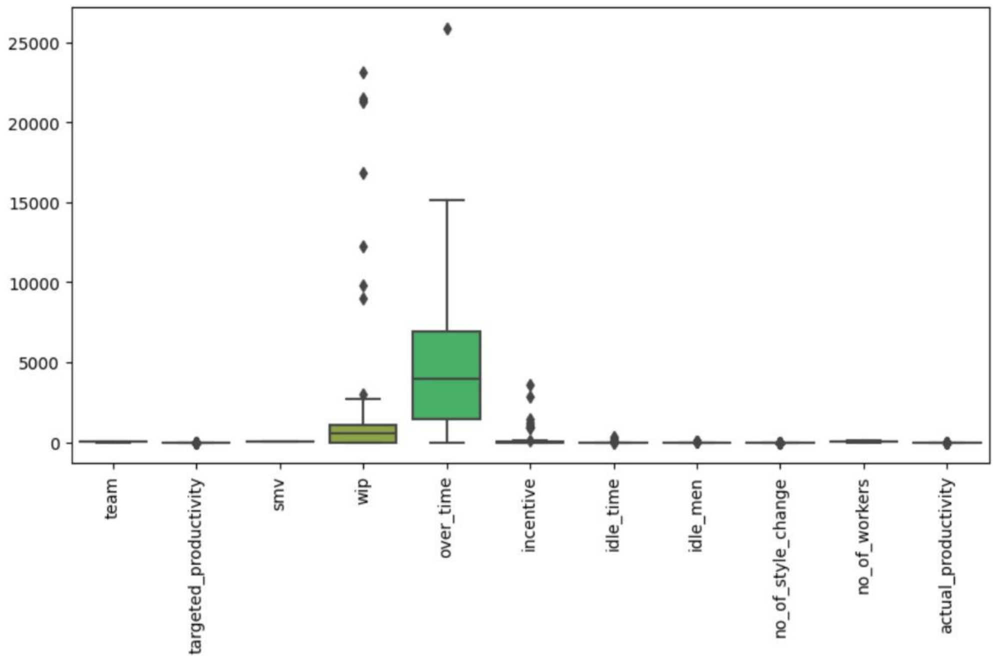
Activity 2.2: Outlier Detection and Removal

Outliers Detection

```
plt.figure(figsize=(10,5))
p = sns.boxplot(data = df3, orient ='v',width=0.8)
plt.xticks(rotation=90)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),
 [Text(0, 0, 'team'),
  Text(1, 0, 'targeted_productivity'),
  Text(2, 0, 'smv'),
  Text(3, 0, 'wip'),
  Text(4, 0, 'over_time'),
  Text(5, 0, 'incentive'),
  Text(6, 0, 'idle_time'),
  Text(7, 0, 'idle_men'),
  Text(8, 0, 'no_of_style_change'),
  Text(9, 0, 'no_of_workers'),
  Text(10, 0, 'actual_productivity')])
```

Plot before removing outliers



Removing Outliers for 'incentive' column

```python
Q1 = df3.incentive.quantile(0.25)
Q3 = df3.incentive.quantile(0.75)
Q1, Q3
```

```
(0.0, 50.0)
```

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

```
50.0
```

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

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

```python
df3[(df3.incentive<lower_limit)|(df3.incentive>upper_limit)]
```

	quarter	department	team	targeted_productivity	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers	actual_p
730	Quarter2	sweing	1	0.80	22.52	1397.0	0	138	0.0	0	0	57.0	
1128	Quarter2	finishing	11	0.80	2.90	0.0	0	960	0.0	0	0	8.0	
1129	Quarter2	finishing	12	0.80	4.60	0.0	0	1080	0.0	0	0	9.0	
1130	Quarter2	finishing	5	0.60	3.94	0.0	0	2880	0.0	0	0	12.0	
1133	Quarter2	finishing	9	0.75	2.90	0.0	0	3600	0.0	0	0	15.0	
1137	Quarter2	finishing	3	0.80	4.60	0.0	0	1440	0.0	0	0	12.0	
1138	Quarter2	finishing	4	0.75	3.94	0.0	0	960	0.0	0	0	8.0	
1139	Quarter2	finishing	1	0.75	3.94	0.0	0	960	0.0	0	0	8.0	
1143	Quarter2	finishing	2	0.70	3.90	0.0	0	1200	0.0	0	0	10.0	
1148	Quarter2	finishing	10	0.70	2.90	0.0	0	960	0.0	0	0	8.0	
1149	Quarter2	finishing	8	0.65	3.90	0.0	0	960	0.0	0	0	8.0	

```python
df4 = df3[(df3.incentive>lower_limit)&(df3.incentive<upper_limit)]
```

```python
df4.shape
```

```
(1186, 13)
```

This code snippet filters a DataFrame df3 by selecting rows where the 'incentive' column values are either less than a specified lower limit (lower_limit) or greater than a specified upper limit (upper_limit). The result is a DataFrame containing only the rows that meet this condition.

Removing Outliers for 'wip' column

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

```
(0.0, 1084.75)
```

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

```
1084.75
```

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

```
(-1627.125, 2711.875)
```

Removing Outliers for 'over_time' column

```
]: Q1 = df5.over_time.quantile(0.25)
   Q3 = df5.over_time.quantile(0.75)
   Q1, Q3
]: (1440.0, 6960.0)
```

```
]: IQR = Q3 - Q1
   IQR
]: 5520.0
```

```
]: lower_limit = Q1 - 1.5*IQR
   upper_limit = Q3 + 1.5*IQR
   lower_limit, upper_limit
]: (-6840.0, 15240.0)
```

```
]: df5[(df5.over_time<lower_limit)|(df5.over_time>upper_limit)]
]:
```

	quarter	department	team	targeted_productivity	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers	actual_produ
146	Quarter2	sweing	11	0.35	12.52	287.0	25920	38	0.0	0	0	54.0	0.

```
]: df6 = df5[(df5.over_time>lower_limit)&(df5.over_time<upper_limit)]
```

```
]: df6.shape
]: (1176, 13)
```

Activity 2.3: Handling Categorical Values

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

```
for i in range(0, df6.shape[1]):
    if df6.dtypes[i] == 'object':
        df6.loc[:, df6.columns[i]] = le.fit_transform(df6.loc[:, df6.columns[i]])
```

This code is encoding the values in a column named "department" by using a LabelEncoder() object to convert the original values into numerical encoded values. The original and encoded values are printed before and after the encoding is performed.

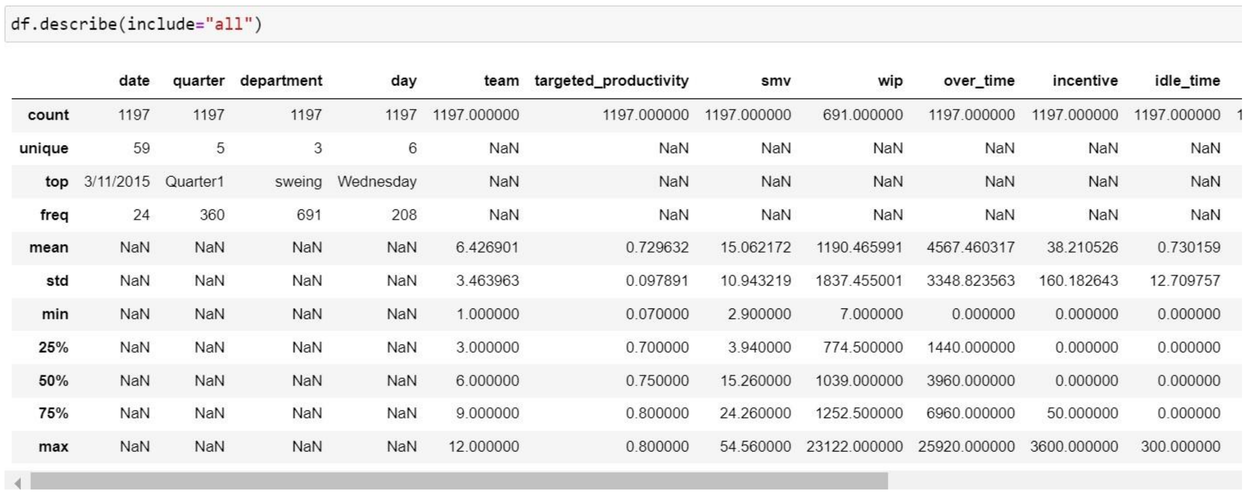
	quarter	department	team	targeted_productivity	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers
0	0	2	8	0.80	26.16	1108.0	7080	98	0.0	0	0	59.0
1	0	1	1	0.75	3.94	0.0	960	0	0.0	0	0	8.0
2	0	2	11	0.80	11.41	968.0	3660	50	0.0	0	0	30.5
3	0	2	12	0.80	11.41	968.0	3660	50	0.0	0	0	30.5
4	0	2	6	0.80	25.90	1170.0	1920	50	0.0	0	0	56.0

This code is encoding the values in a column named "day" by using a LabelEncoder() object to convert the original values into numerical encoded values. The original and encoded values are printed before and after the encoding is performed.

Milestone 3: Exploratory Data Analysis

Activity 1: Descriptive statistical

The purpose of descriptive analysis is to analyze the basic features of data using a statistical technique. In this case, Pandas has a useful function called describe. We can understand the unique, top, and frequent values of categorical features with this describe function. We can also get the count, mean, standard deviation, minimum, maximum, and percentile values of continuous features.



```
df.describe(include="all")
```

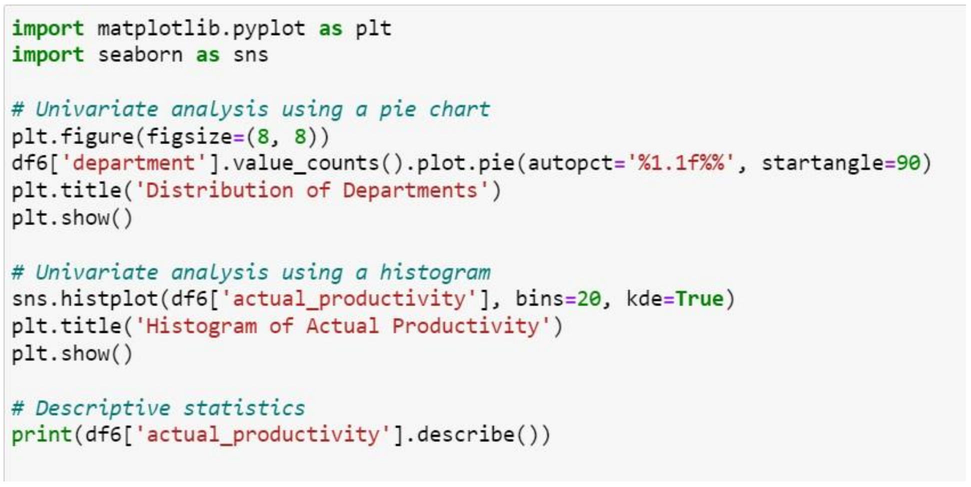
	date	quarter	department	day	team	targeted_productivity	smv	wip	over_time	incentive	idle_time
count	1197	1197	1197	1197	1197.000000	1197.000000	1197.000000	691.000000	1197.000000	1197.000000	1197.000000
unique	59	5	3	6	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	3/11/2015	Quarter1	sweing	Wednesday	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	24	360	691	208	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	6.426901	0.729632	15.062172	1190.465991	4567.460317	38.210526	0.730159
std	NaN	NaN	NaN	NaN	3.463963	0.097891	10.943219	1837.455001	3348.823563	160.182643	12.709757
min	NaN	NaN	NaN	NaN	1.000000	0.070000	2.900000	7.000000	0.000000	0.000000	0.000000
25%	NaN	NaN	NaN	NaN	3.000000	0.700000	3.940000	774.500000	1440.000000	0.000000	0.000000
50%	NaN	NaN	NaN	NaN	6.000000	0.750000	15.260000	1039.000000	3960.000000	0.000000	0.000000
75%	NaN	NaN	NaN	NaN	9.000000	0.800000	24.260000	1252.500000	6960.000000	50.000000	0.000000
max	NaN	NaN	NaN	NaN	12.000000	0.800000	54.560000	23122.000000	25920.000000	3600.000000	300.000000

Activity 2: Visual analysis

Visual analysis is the process of examining and understanding data via the use of visual representations such as charts, plots, and graphs. It is a method for quickly identifying patterns, trends, and outliers in data, which can aid in gaining insights and making sound decisions.

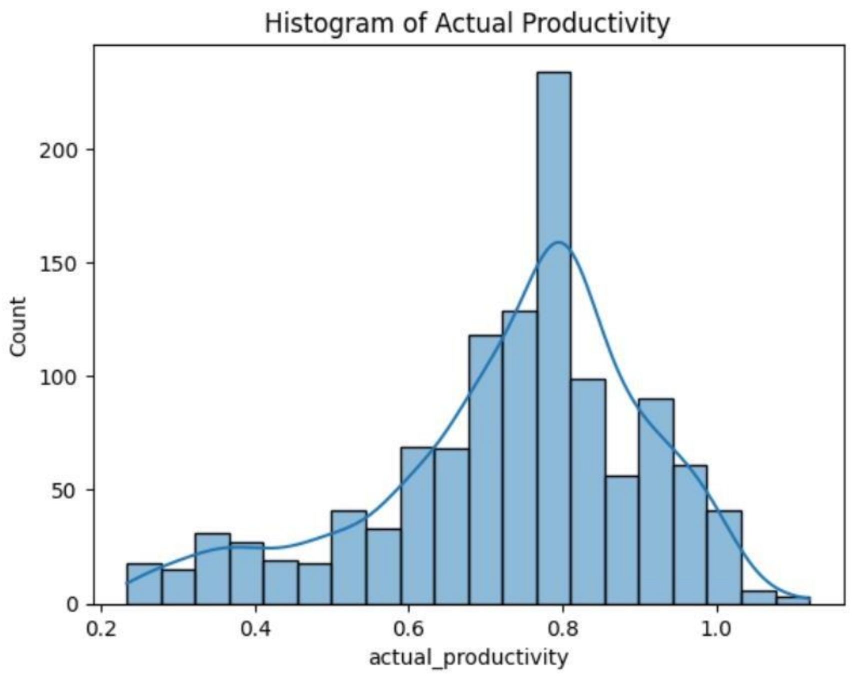
Activity 2.1: Univariate analysis

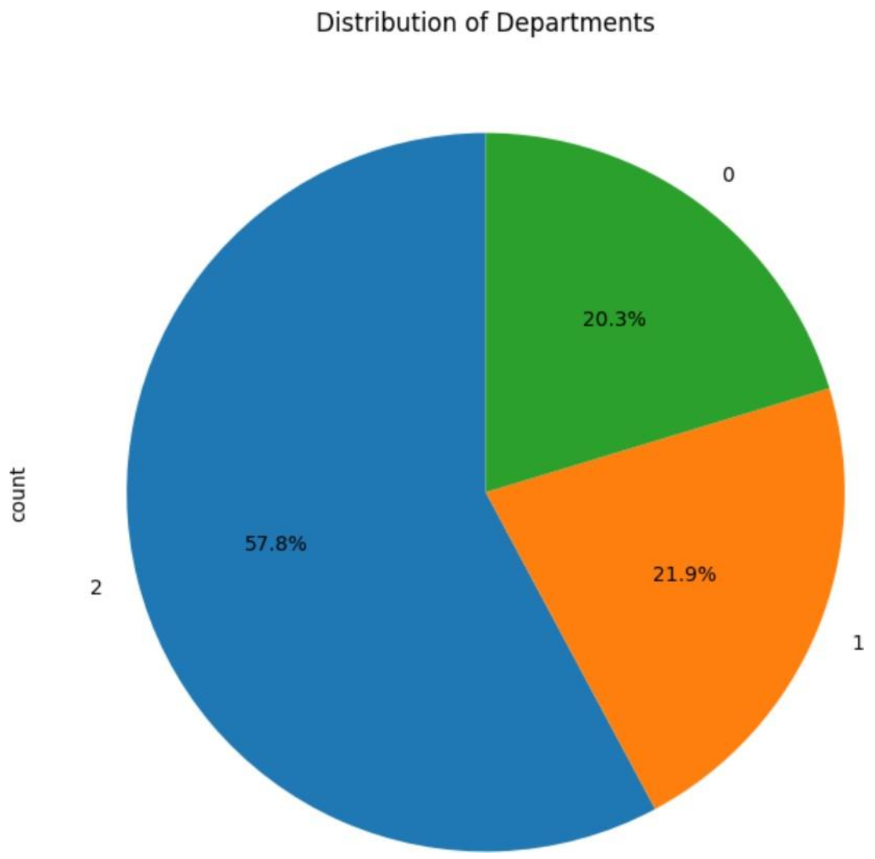
Univariate analysis is a statistical method used to analyse a single variable in a dataset. This analysis focuses on understanding the distribution, central tendency, and dispersion of a single variable.



```python
import matplotlib.pyplot as plt
import seaborn as sns

# Univariate analysis using a pie chart
plt.figure(figsize=(8, 8))
df6['department'].value_counts().plot.pie(autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Departments')
plt.show()

# Univariate analysis using a histogram
sns.histplot(df6['actual_productivity'], bins=20, kde=True)
plt.title('Histogram of Actual Productivity')
plt.show()

# Descriptive statistics
print(df6['actual_productivity'].describe())
```

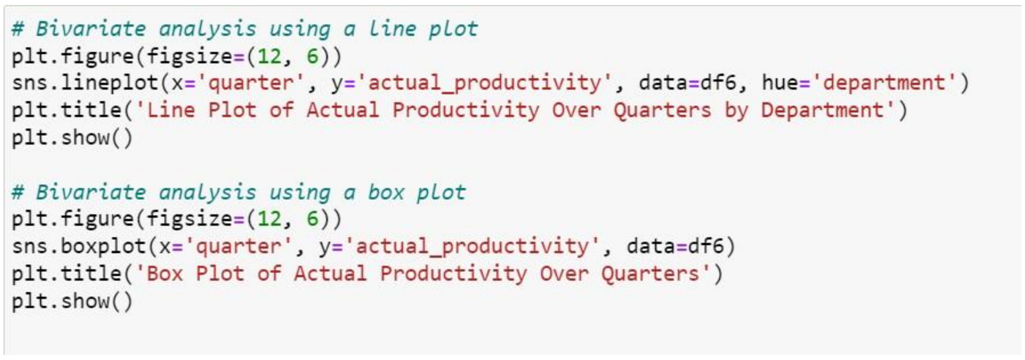
This code creates a histogram and piechart using the Seaborn library to show the number of occurrences of each unique value in the "department" column of a Pandas DataFrame. The x-axis represents the unique values of the "department" column, and the y-axis represents the number of times each unique value appears in the column. The plt.xlabel(), plt.ylabel(), and plt.title() functions are used to add labels and a title to the plot. Finally, plt.show() is used to display the plot.





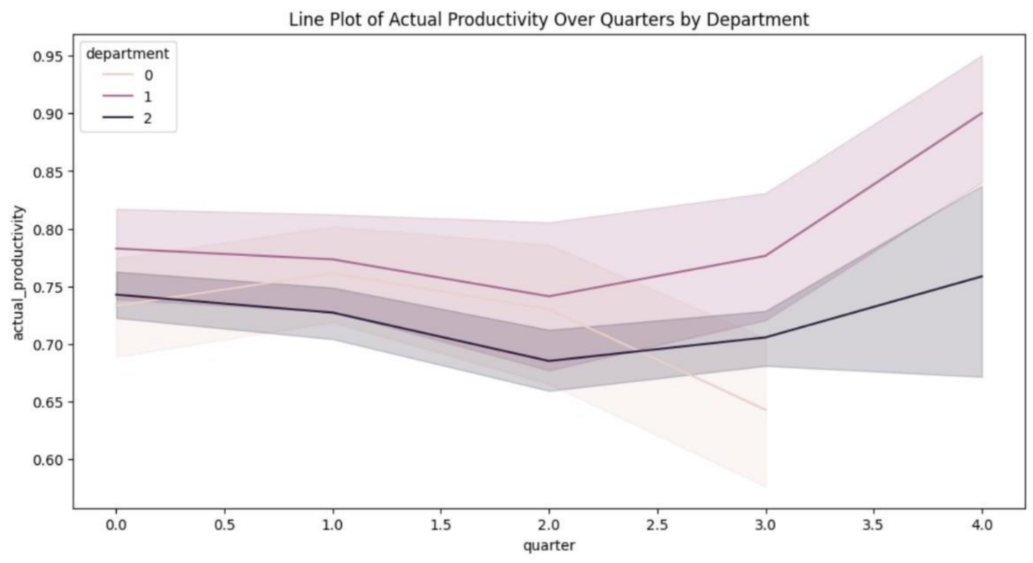
Activity 2.2: Bivariate analysis

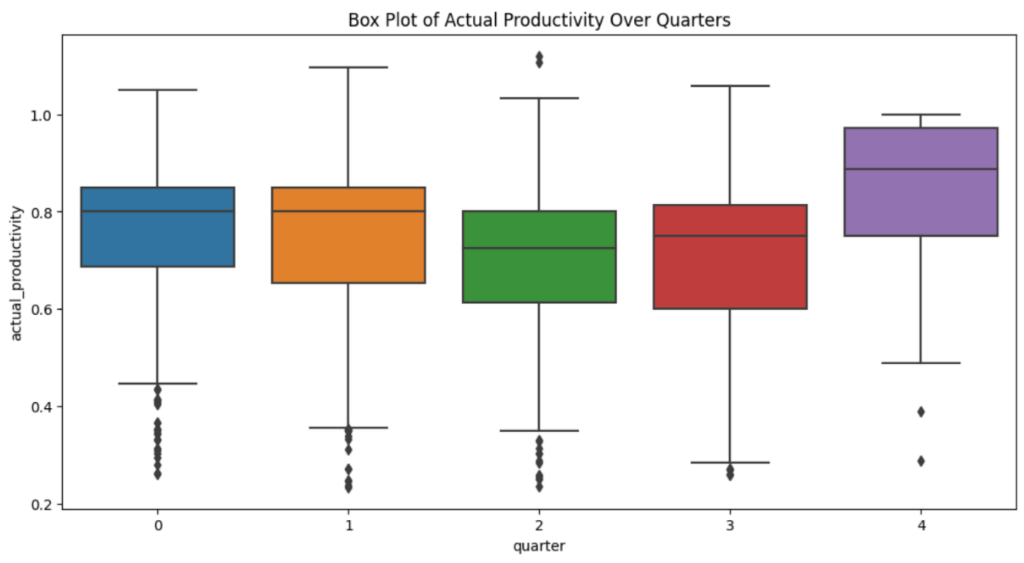
Bivariate analysis is a statistical method used to analyse the relationship between two variables in a dataset. This analysis focuses on examining how changes in one variable are related to changes in another variable.



```
# Bivariate analysis using a line plot
plt.figure(figsize=(12, 6))
sns.lineplot(x='quarter', y='actual_productivity', data=df6, hue='department')
plt.title('Line Plot of Actual Productivity Over Quarters by Department')
plt.show()

# Bivariate analysis using a box plot
plt.figure(figsize=(12, 6))
sns.boxplot(x='quarter', y='actual_productivity', data=df6)
plt.title('Box Plot of Actual Productivity Over Quarters')
plt.show()
```

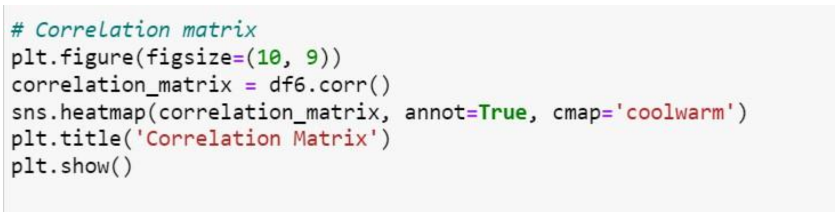
This code generates a line plot and box plot using the Seaborn library to show the relationship between team number and the number of unfinished items. The line plot shows how the number of unfinished items varies across different teams. The x-axis represents the team number, and the y-axis represents the number of unfinished items. The title of the line plot is "Line Plot of Unfinished Items by Team Number".





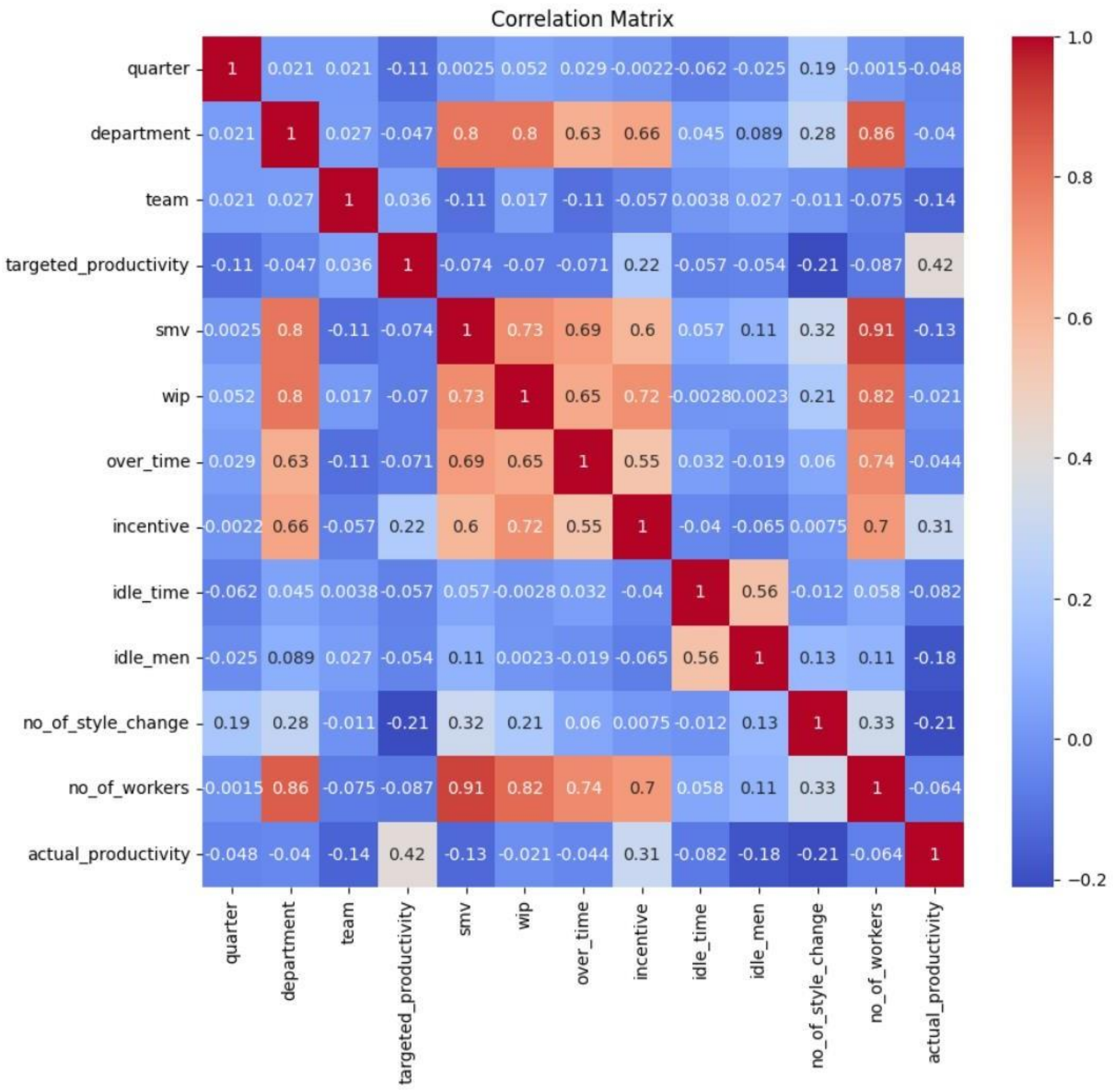
Activity 2.3: Multivariate analysis

Multivariate analysis is a statistical technique used to analyse data that involves more than two variables. It aims to understand the relationships between multiple variables in a dataset by examining how they are related to each other and how they contribute to a particular outcome or phenomenon.



```python
# Correlation matrix
plt.figure(figsize=(10, 9))
correlation_matrix = df6.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

This code creates a heatmap to show the correlation between the numerical variables in the dataset. The darker the color of the square, the stronger the correlation between the variables. The annotated values in the square show the correlation coefficients.



Splitting data into train and test

Now let us split the Dataset into train and test sets. First split the dataset into X and y and then split the data set. The ‘X’ corresponds to independent features and ‘y’ corresponds to the target variable.

```
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)
```

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

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

Applying Standard Scalar

We apply standard scaling to independent variables before training because it helps to normalize the data and brings all the features to a similar scale. This is important because some machine learning algorithms are sensitive to the scale of the input features, and having features on vastly different scales can lead to suboptimal performance.

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

```
scaler = StandardScaler()
```

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

```
x.head()
```

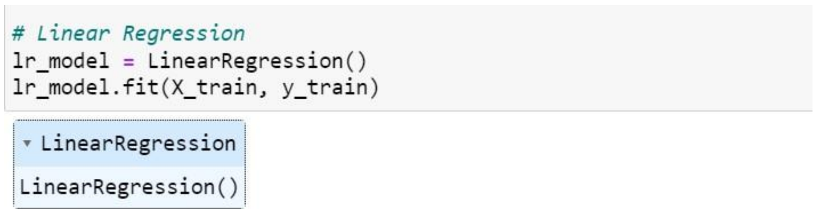
	quarter	department	team	targeted_productivity	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers
0	-1.160117	0.780739	0.455616	0.720481	1.008608	0.926217	0.760786	2.413020	-0.057985	-0.114022	-0.355135	1.097504
1	-1.160117	-0.468444	-1.571751	0.209020	-1.018120	-1.019391	-1.102067	-0.847073	-0.057985	-0.114022	-0.355135	-1.201499
2	-1.160117	0.780739	1.324487	0.720481	-0.336767	0.680382	-0.280220	0.816240	-0.057985	-0.114022	-0.355135	-0.187233
3	-1.160117	0.780739	1.614110	0.720481	-0.336767	0.680382	-0.280220	0.816240	-0.057985	-0.114022	-0.355135	-0.187233
4	-1.160117	0.780739	-0.123632	0.720481	0.984893	1.035087	-0.809855	0.816240	-0.057985	-0.114022	-0.355135	0.962269

Milestone 4: Model Building

Activity 1: Training the model in multiple algorithms

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying seven regression algorithms. The best model is saved based on its performance.

Activity 1.1: Linear Regression Model



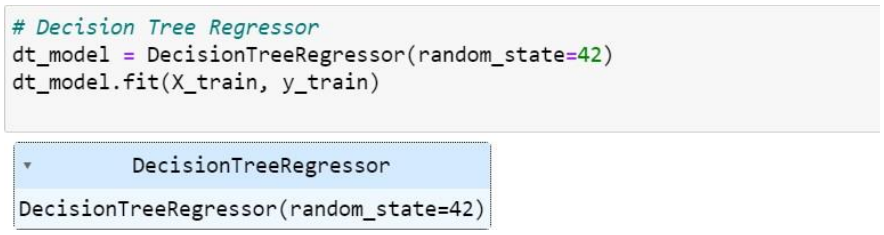
```
# Linear Regression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

This code performs linear regression modeling using the scikit-learn library.

It creates a LinearRegression object named "lr". The "fit" method is then called on the training data X_train and y_train. This method trains the model by finding the coefficients for the linear equation that best fits the data.

Activity 1.2: Decision Tree Regressor Model

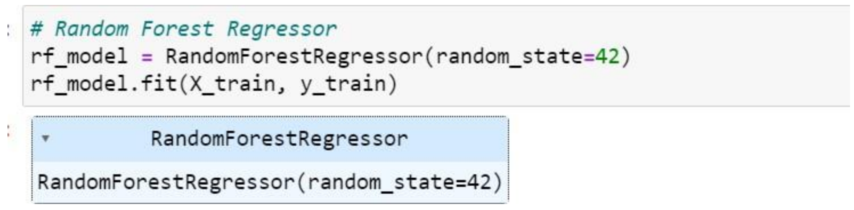


```
# Decision Tree Regressor
dt_model = DecisionTreeRegressor(random_state=42)
dt_model.fit(X_train, y_train)
```

```
▾        DecisionTreeRegressor
DecisionTreeRegressor(random_state=42)
```

This code is training a decision tree regression model using the training data (X_train and y_train) with the specified hyperparameters (max_depth=4, min_samples_split=3, min_samples_leaf=2). The model is stored in the variable 'dtr'. After training, the model will be able to make predictions on new data based on the relationships learned from the training data.

Activity 1.3: Random Forest Regressor Model



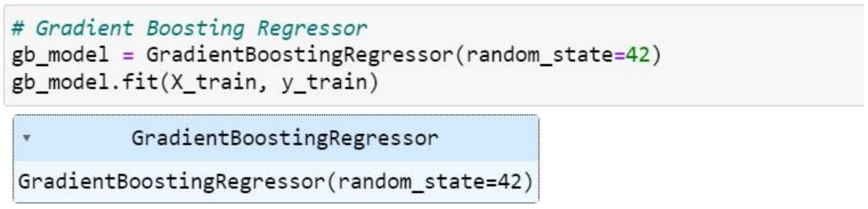
```
# Random Forest Regressor
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train, y_train)
```

```
▾        RandomForestRegressor
RandomForestRegressor(random_state=42)
```

The code creates an instance of the RandomForestRegressor class with certain hyperparameters set. The hyperparameters specify the number of trees to use in the random forest, the maximum depth of each tree, the minimum weight fraction required to be at a leaf node, the maximum number of features to consider when splitting a node, and a random state to ensure reproducibility of results. The rfr object is then trained on the training data X_train and y_train using the fit method.

The trained model can then be used to make predictions on new data.

Activity 1.4: Gradient Boosting Regressor Model



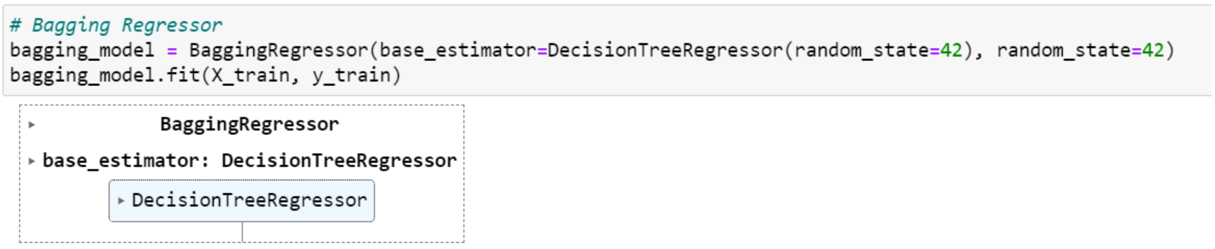
```
# Gradient Boosting Regressor
gb_model = GradientBoostingRegressor(random_state=42)
gb_model.fit(X_train, y_train)
```

```
▾          GradientBoostingRegressor
GradientBoostingRegressor(random_state=42)
```

The code is fitting a Gradient Boosting Regressor model to the training data X_train and y_train using n_estimators equal to 100 (the number of trees in the forest), learning_rate equal to 0.1 (the step size shrinkage used to prevent overfitting), max_depth equal to 1 (the maximum depth of the individual regression estimators), and random_state equal to 42 (to ensure reproducibility of results).

The fitted model can then be used to make predictions on new data using the predict() method.

Activity 1.5: Bagging Regressor Model



```
# Bagging Regressor
bagging_model = BaggingRegressor(base_estimator=DecisionTreeRegressor(random_state=42), random_state=42)
bagging_model.fit(X_train, y_train)
```

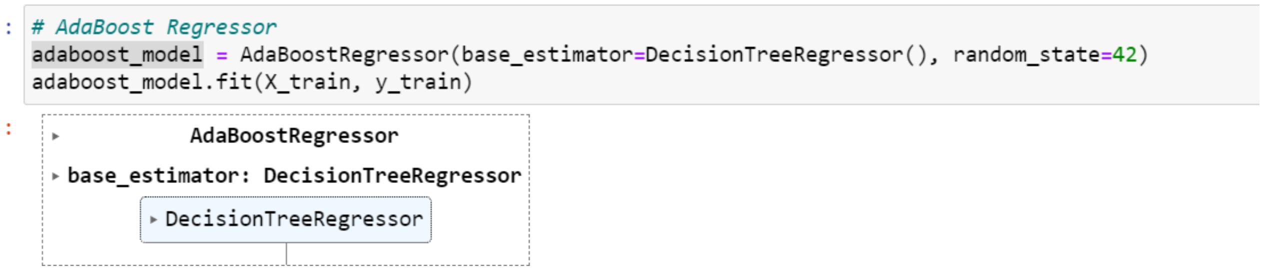
```
▸           BaggingRegressor
▸ base_estimator: DecisionTreeRegressor
        ▸ DecisionTreeRegressor
```

This code creates a machine learning model using the XGBoost algorithm. The model is designed to predict a numeric value (the target variable) based on a set of input variables.

To improve the accuracy of the model, the Bagging technique is used. This involves creating multiple versions of the model, each with slightly different training data, and combining their predictions to create a final prediction.

Finally, the model is trained using a set of input data (X_train) and the corresponding target values (y_train). This involves adjusting the weights of the various components of the model until it can accurately predict the target values based on the input data.

Activity 1.6: Boosting Regressor Model



```
# AdaBoost Regressor
adaboost_model = AdaBoostRegressor(base_estimator=DecisionTreeRegressor(), random_state=42)
adaboost_model.fit(X_train, y_train)
```

```
▸           AdaBoostRegressor
▸ base_estimator: DecisionTreeRegressor
        ▸ DecisionTreeRegressor
```

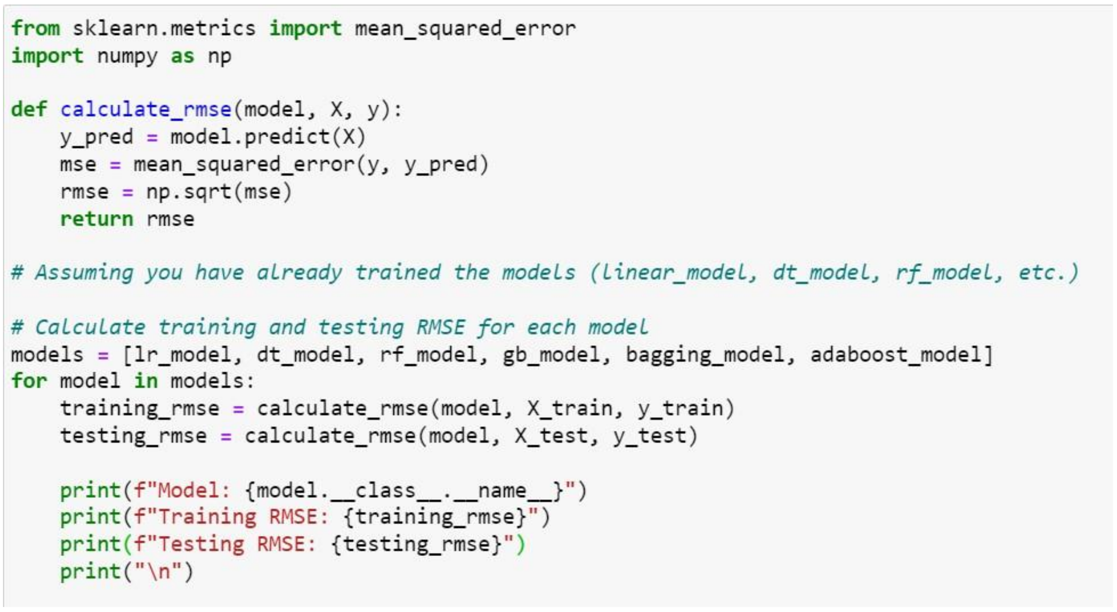
To improve the accuracy of the model, the AdaBoost technique is used. AdaBoost stands for Adaptive Boosting and works by creating multiple versions of the model, each with slightly different training data, and weighting the predictions of each model based on its accuracy.

Finally, the model is trained using a set of input data (X_train) and the corresponding target values (y_train). This involves adjusting the weights of the various components of the model until it can accurately predict the target values based on the input data.

Milestone 5: Performance Testing

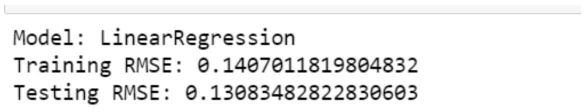
Activity 1: Check model performance on test and train data for each model

To check the model performance on test and train data, we calculate the RMSE score. The RMSE score tells us how far off the predictions of the model are, on average, from the actual values.



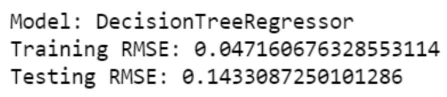
```
from sklearn.metrics import mean_squared_error
import numpy as np

def calculate_rmse(model, X, y):
    y_pred = model.predict(X)
    mse = mean_squared_error(y, y_pred)
    rmse = np.sqrt(mse)
    return rmse

# Assuming you have already trained the models (linear_model, dt_model, rf_model, etc.)

# Calculate training and testing RMSE for each model
models = [lr_model, dt_model, rf_model, gb_model, bagging_model, adaboost_model]
for model in models:
    training_rmse = calculate_rmse(model, X_train, y_train)
    testing_rmse = calculate_rmse(model, X_test, y_test)

    print(f"Model: {model.__class__.__name__}")
    print(f"Training RMSE: {training_rmse}")
    print(f"Testing RMSE: {testing_rmse}")
    print("\n")
```

Activity 1.1: Linear Regression Model



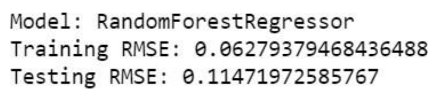
```
Model: LinearRegression
Training RMSE: 0.1407011819804832
Testing RMSE: 0.13083482822830603
```

Activity 1.2: Decision Tree Regressor Model



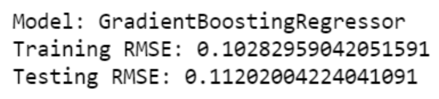
```
Model: DecisionTreeRegressor
Training RMSE: 0.047160676328553114
Testing RMSE: 0.1433087250101286
```

Activity 1.3: Random Forest Regressor Model



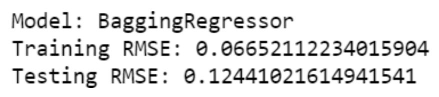
```
Model: RandomForestRegressor
Training RMSE: 0.06279379468436488
Testing RMSE: 0.11471972585767
```

Activity 1.4: Gradient Boosting Regressor Model



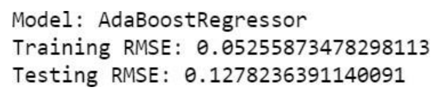
```
Model: GradientBoostingRegressor
Training RMSE: 0.10282959042051591
Testing RMSE: 0.11202004224041091
```

Activity 1.5: Bagging Regressor Model



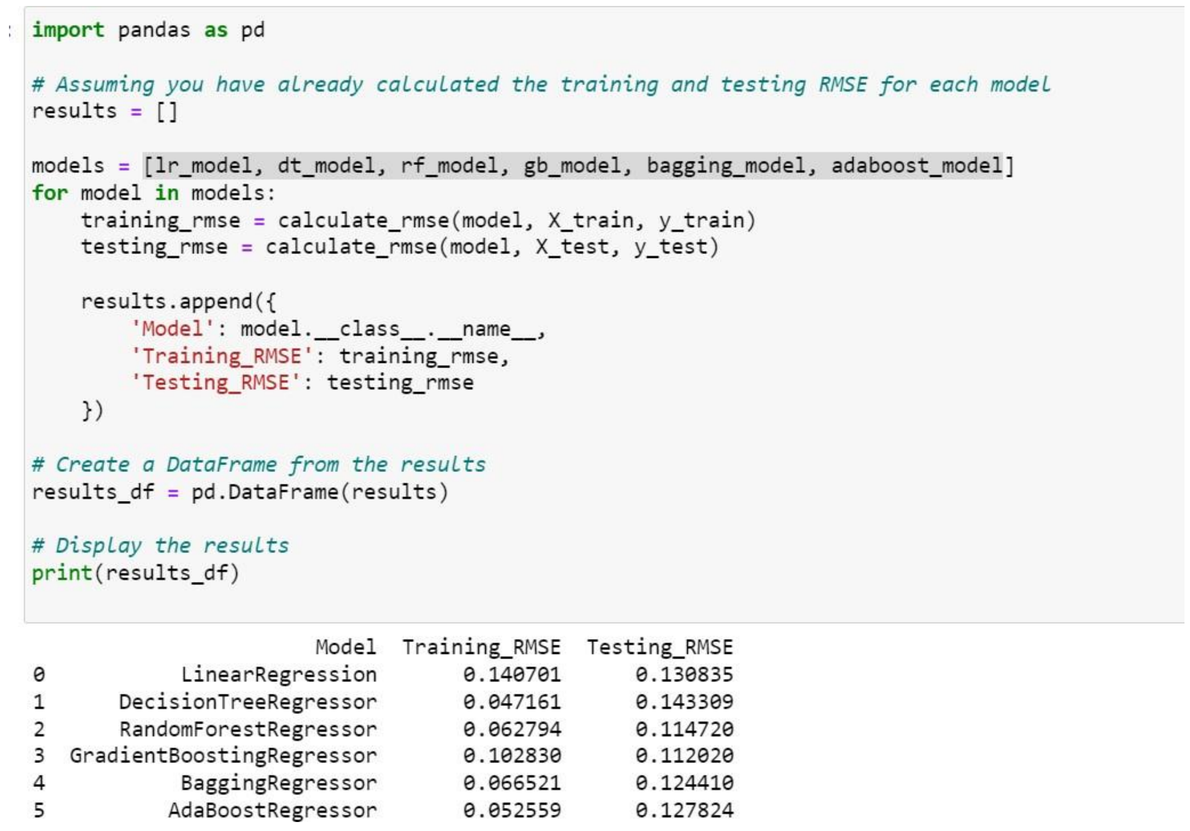
```
Model: BaggingRegressor
Training RMSE: 0.06652112234015904
Testing RMSE: 0.12441021614941541
```

Activity 1.6: Boosting Regressor Model



```
Model: AdaBoostRegressor
Training RMSE: 0.05255873478298113
Testing RMSE: 0.1278236391140091
```

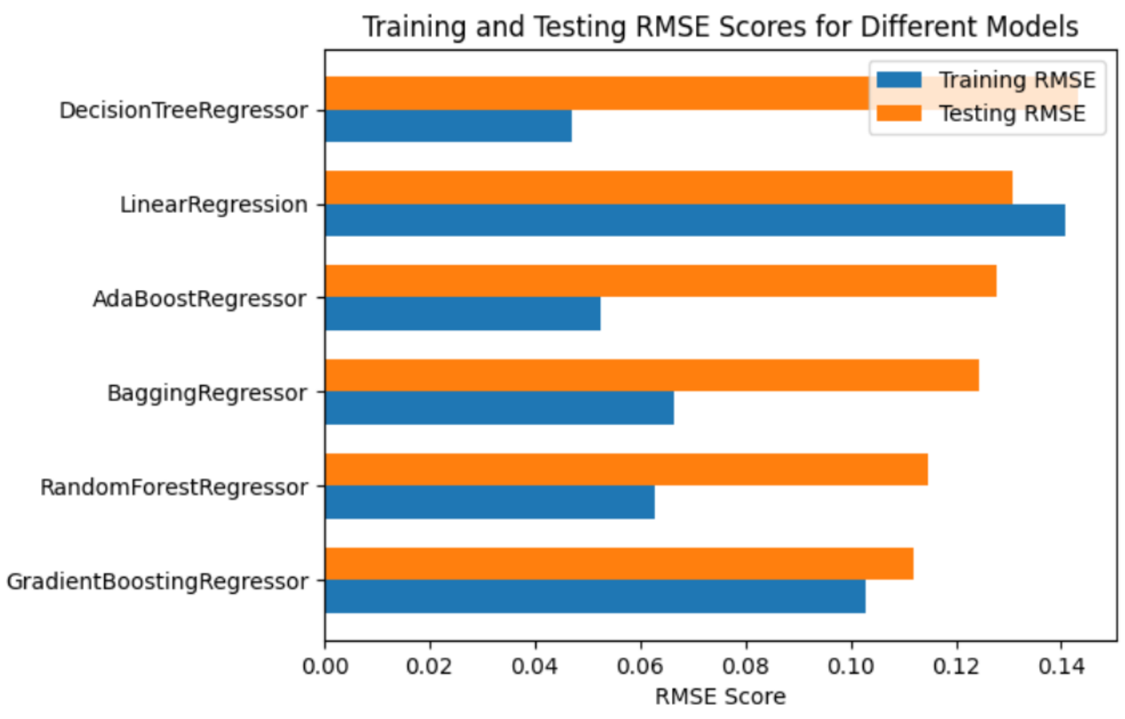
Activity 2: Comparing models



```python
import pandas as pd

# Assuming you have already calculated the training and testing RMSE for each model
results = []

models = [lr_model, dt_model, rf_model, gb_model, bagging_model, adaboost_model]
for model in models:
    training_rmse = calculate_rmse(model, X_train, y_train)
    testing_rmse = calculate_rmse(model, X_test, y_test)

    results.append({
        'Model': model.__class__.__name__,
        'Training_RMSE': training_rmse,
        'Testing_RMSE': testing_rmse
    })

# Create a DataFrame from the results
results_df = pd.DataFrame(results)

# Display the results
print(results_df)
```

```
                       Model  Training_RMSE  Testing_RMSE
0           LinearRegression       0.140701      0.130835
1      DecisionTreeRegressor       0.047161      0.143309
2      RandomForestRegressor       0.062794      0.114720
3  GradientBoostingRegressor       0.102830      0.112020
4           BaggingRegressor       0.066521      0.124410
5          AdaBoostRegressor       0.052559      0.127824
```

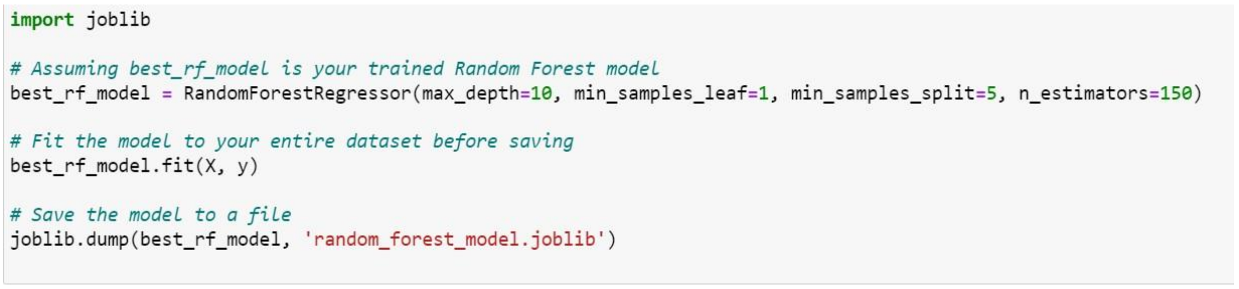
This code creates a Pandas Data Frame named "results" that contains the model names and root mean squared errors (RMSE) for both the training and testing data for each of the seven regression models: Linear Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, XG Boost Regressor, Bagging Regressor, and Boosting Regressor.



To determine which model is best, we should look for the model with the lowest RMSE value on the test data. This suggests that the model predicts value closer to the actual values. In this out of the seven models selected Boosting Regressor satisfies the conditions and hence selected.

Milestone 6: Model Deployment

Activity 1: Save the best model



```
import joblib

# Assuming best_rf_model is your trained Random Forest model
best_rf_model = RandomForestRegressor(max_depth=10, min_samples_leaf=1, min_samples_split=5, n_estimators=150)

# Fit the model to your entire dataset before saving
best_rf_model.fit(X, y)

# Save the model to a file
joblib.dump(best_rf_model, 'random_forest_model.joblib')
```

This code uses the "joblib" library in Python to save the trained random forest model named "random forest" as a file named "random_forest_model.joblib".

The "dump" method from the pickle library is used to save the model object in a serialized form that can be used again later. The "wb" parameter indicates that the file should be opened in binary mode to write data to it.

Activity 2: Integrate with Web Framework

In this section, we will be building a web application that would help us integrate the machine learning model we have built and trained.
A user interface is provided for the users to enter the values for predictions. The entered values are fed into the saved model, and the prediction is displayed on the UI.

The section has following task:

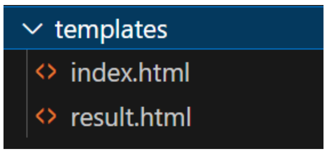
- Building HTML pages
- Building server side script
- Run the web application

Activity 2.1: Building Html Pages:

For this project we create three html files:

- index.html
- predict.html
- productivity.html

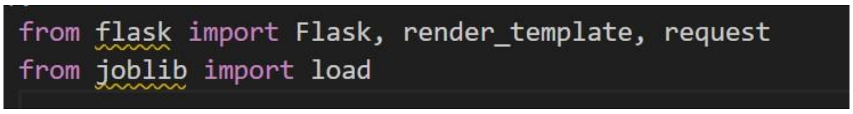
and save these html files in the templates folder.



```
templates
  index.html
  result.html
```

Activity 2.2: Build Python code:

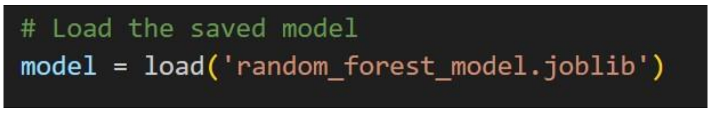
Importing the libraries



```
from flask import Flask, render_template, request
from joblib import load
```

This code first loads the saved Boosting Regressor model from the "random_forest_model.joblib" file using the "joblib.load()" method. The "rb" parameter indicates that the file should be opened in binary mode to read data from it.

After loading the model, the code creates a new Flask web application object named "app" using the Flask constructor. The "name" argument tells Flask to use the current module as the name for the application.

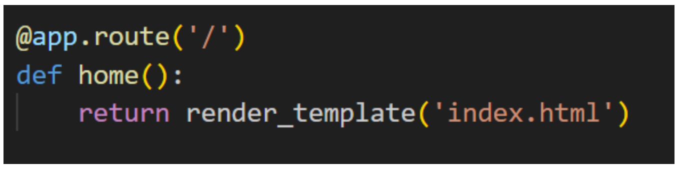


```
# Load the saved model
model = load('random_forest_model.joblib')
```

This code sets up a new route for the Flask web application using the "@app.route()" decorator. The route in this case is the root route "/", which is the default route when the website is accessed.

The function "home()" is then associated with this route. When a user accesses the root route of the website, this function is called.

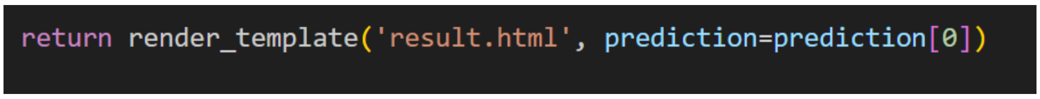
The "render_template()" method is used to render an HTML template named "index.html". The “index.html” is the home page.



```python
@app.route('/')
def home():
    return render_template('index.html')
```

The route in this case is "/predict". When a user accesses the "/predict" route of the website, this function “index()” is called.

The "render_template()" method is used to render an HTML template named "predict.html".



```python
return render_template('result.html', prediction=prediction[0])
```

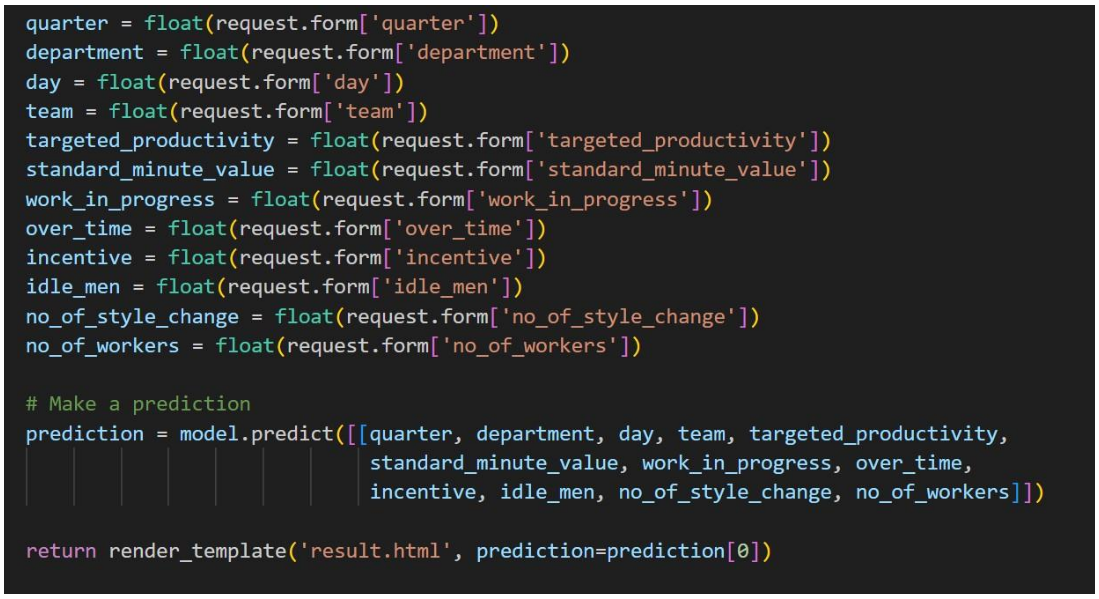
This code sets up another route for the Flask web application using the "@app.route()" decorator. The route in this case is "/data_predict", and the method is set to GET and POST.The function "predict()" is then associated with this route.

The input data is collected from an HTML form and includes information such as the quarter, department, day of the week, team number, time allocated, unfinished items, overtime, incentive, idle time, idle men, style change, and number of workers.

The code converts some of the input data into a format that can be used by the machine learning model. For example, the department input is converted from a string to a binary value (1 for sewing, 0 for finishing), and the day of the week input is converted to a numerical value (0-6).

The model is then used to make a prediction based on the input data. The prediction is rounded to 4 decimal places and multiplied by 100 to convert it to a percentage.

The prediction value is passed to the HTML template 'productivity.html' where it is displayed as a text message. The message informs the user of the predicted productivity based on the input data.



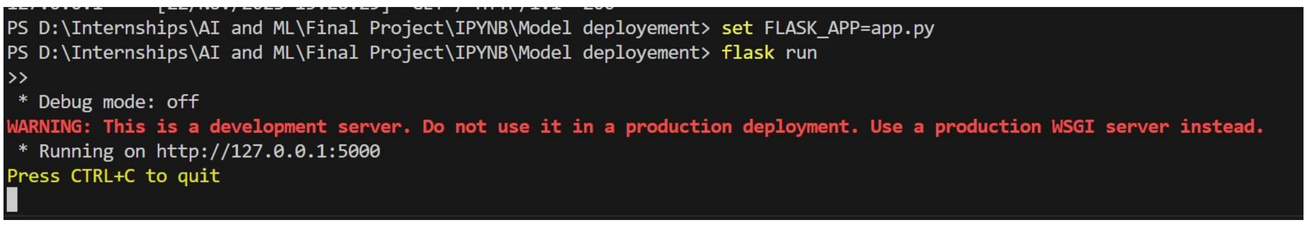
```python
quarter = float(request.form['quarter'])
department = float(request.form['department'])
day = float(request.form['day'])
team = float(request.form['team'])
targeted_productivity = float(request.form['targeted_productivity'])
standard_minute_value = float(request.form['standard_minute_value'])
work_in_progress = float(request.form['work_in_progress'])
over_time = float(request.form['over_time'])
incentive = float(request.form['incentive'])
idle_men = float(request.form['idle_men'])
no_of_style_change = float(request.form['no_of_style_change'])
no_of_workers = float(request.form['no_of_workers'])

# Make a prediction
prediction = model.predict([[quarter, department, day, team, targeted_productivity,
                             standard_minute_value, work_in_progress, over_time,
                             incentive, idle_men, no_of_style_change, no_of_workers]])

return render_template('result.html', prediction=prediction[0])
```

Main Function:

This code sets the entry point of the Flask application. The function "app.run()" is called, which starts the Flask development server.

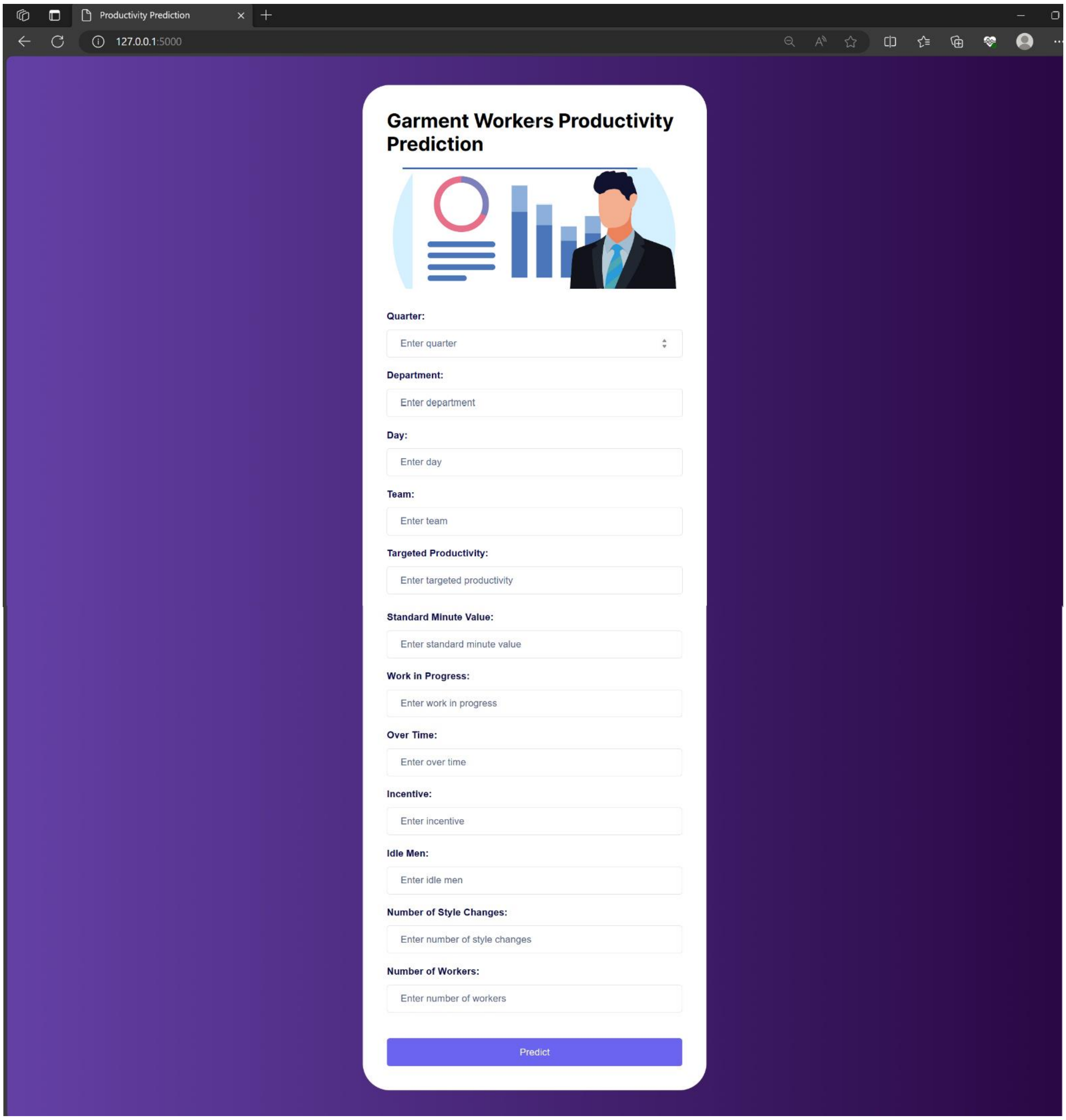
Activity 2.3: Run the web application

When you run the “app.py” file this window will open in the output terminal. Copy the http://127.0.0.1:5000 and paste this link in your browser.



```
PS D:\Internships\AI and ML\Final Project\IPYNB\Model deployement> set FLASK_APP=app.py
PS D:\Internships\AI and ML\Final Project\IPYNB\Model deployement> flask run
>>
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Website Screenshots:



Garment Workers Productivity Prediction

Quarter: Enter quarter

Department: Enter department

Day: Enter day

Team: Enter team

Targeted Productivity: Enter targeted productivity

Standard Minute Value: Enter standard minute value

Work in Progress: Enter work in progress

Over Time: Enter over time

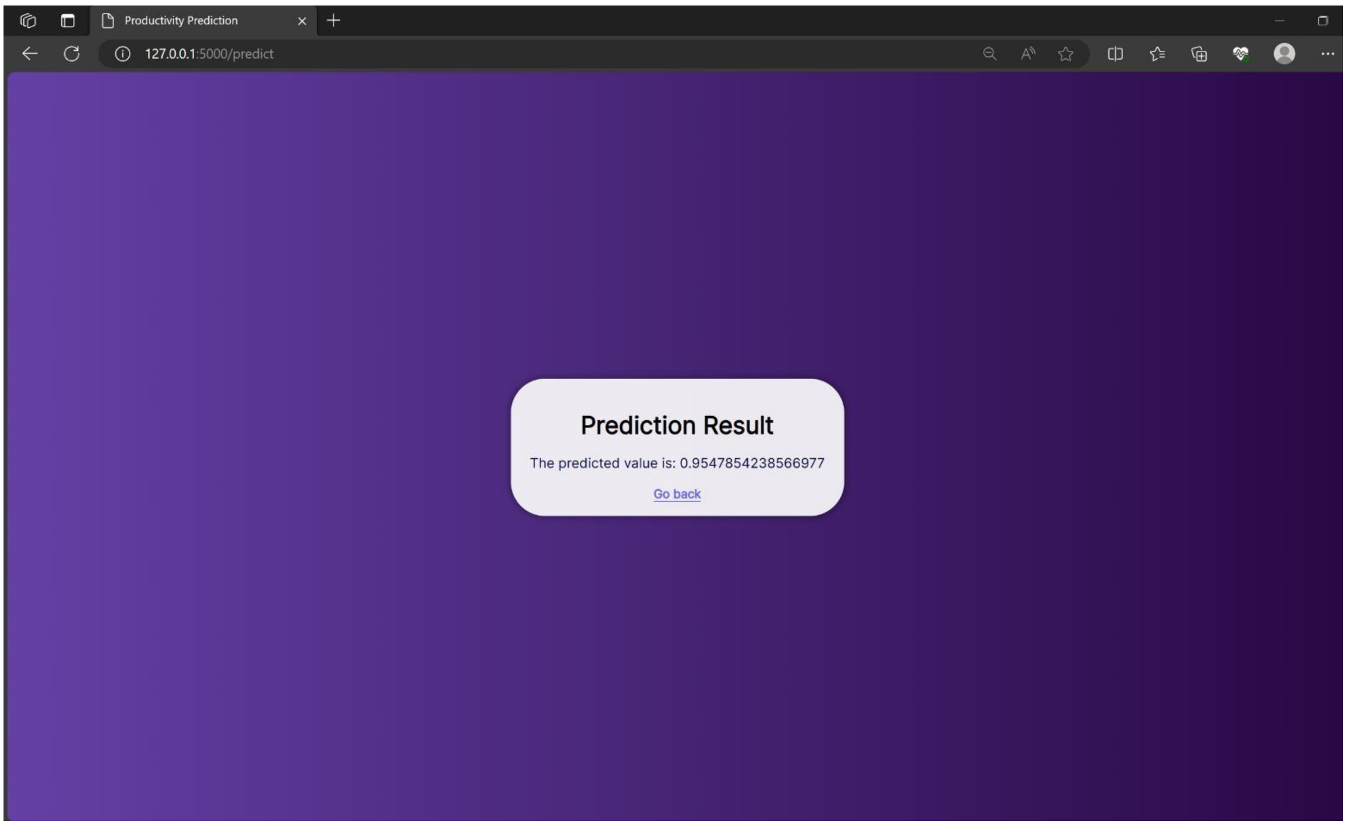
Incentive: Enter incentive

Idle Men: Enter idle men

Number of Style Changes: Enter number of style changes

Number of Workers: Enter number of workers

Predict



Prediction Result

The predicted value is: 0.9547854238566977

10 . **ADVANTAGES AND DISADVANTGES**

Advantages

Optimized Resource Allocation:
A predictive model can assist in optimizing resource allocation by providing insights into factors that influence garment worker productivity. This can lead to more efficient use of manpower, time, and other resources.

Cost Reduction:
By identifying and addressing productivity bottlenecks, companies can reduce operational costs and improve overall cost-effectiveness in garment production.

Data-Driven Decision-Making:
The model enables data-driven decision-making by providing actionable insights into the relationships between various production factors and worker productivity.

Enhanced Competitiveness:
Companies that leverage predictive modeling can adapt more effectively to market demands, enhancing their competitiveness in the dynamic garment industry.

Improved Human Resource Management:
The model can contribute to better human resource management by understanding and addressing factors that impact worker satisfaction, engagement, and productivity.

Strategic Planning:
Predictive modeling allows for more informed strategic planning, enabling companies to anticipate challenges and proactively implement measures to improve efficiency.

Disadvantages

Data Quality Issues:
The effectiveness of the predictive model is highly dependent on the quality of the data used for training. Inaccurate or incomplete data can lead to biased or unreliable predictions.

Complex Implementation:
Developing and implementing a predictive model can be a complex process, requiring specialized skills in data science and machine learning. This complexity may pose challenges for companies with limited technical expertise.

Model Overfitting:
There is a risk of overfitting, where the model performs well on the training data but fails to generalize effectively to new, unseen data. Overfitting can result in inaccurate predictions.

Ethical Considerations:
Predictive models may inadvertently reinforce biases present in the training data, leading to unfair or discriminatory outcomes. Ethical considerations must be carefully addressed to ensure fair treatment of workers.

Resource Intensiveness:
Training and maintaining a predictive model may require significant computational resources, particularly for large datasets. This can lead to increased costs and technical infrastructure requirements.

Resistance to Change:
Implementing a predictive model may face resistance from workers or management who are not accustomed to data-driven decision-making. Overcoming this resistance requires effective communication and change management strategies.

Model Interpretability:
Some complex machine learning models lack interpretability, making it challenging for stakeholders to understand and trust the reasoning behind the model's predictions.

Continuous Model Maintenance:
Predictive models require ongoing maintenance and updates to remain effective as new data becomes available and business conditions change. This necessitates a commitment to continuous monitoring and improvement.

11. CONCLUSION

The development of a predictive model for garment worker productivity holds immense potential for revolutionizing decision-making processes within the garment industry. The advantages, including optimized resource allocation, cost reduction, and enhanced competitiveness, underscore the transformative impact of leveraging data-driven insights. However, it is essential to acknowledge and address challenges such as data quality issues, ethical considerations, and the need for ongoing maintenance. Striking a balance between the advantages and challenges is key to realizing the full benefits of predictive modeling, empowering the industry to adapt efficiently, improve operational efficiency, and ultimately thrive in a rapidly evolving market landscape. Embracing this technological advancement offers an opportunity not only to enhance productivity but also to foster a culture of innovation and resilience within the garment manufacturing sector.

12. FUTURE SCOPE

The future scope of predictive modeling for garment worker productivity is promising, with potential advancements and applications that can further transform the garment industry. Several key areas indicate the evolving landscape of this technology:

Integration of Advanced Technologies:
Future developments may involve the integration of advanced technologies such as artificial intelligence (AI) and machine learning (ML) algorithms to enhance the accuracy and predictive capabilities of models. This could lead to more sophisticated analyses of intricate relationships within the dataset.
Real-Time Predictions:

The evolution towards real-time predictive modeling is anticipated, enabling companies to make instantaneous decisions based on the most current data. This can significantly improve responsiveness to dynamic market conditions and sudden changes in production requirements.

Enhanced Interpretability:
Addressing the challenge of model interpretability, future advancements may focus on developing models that are not only accurate but also transparent and understandable to stakeholders. This could foster greater trust and acceptance of predictive insights.

Customization and Personalization:
Predictive models may evolve to offer more personalized insights, tailoring recommendations and strategies based on the unique characteristics of different garment production environments. This level of customization could lead to more targeted interventions and improvements.

Incorporation of IoT and Sensor Data:
The Internet of Things (IoT) and sensor technologies may be integrated into predictive models to capture real-time data from production floors. This can provide a more comprehensive and granular understanding of the factors influencing worker productivity, leading to more accurate predictions.

Expansion to Supply Chain Management:
The application of predictive modeling may extend beyond the confines of garment production facilities to encompass broader supply chain management. This could involve predicting demand fluctuations, optimizing inventory, and enhancing overall supply chain efficiency.

Emphasis on Ethical AI:
Future developments in predictive modeling will likely place a stronger emphasis on ethical considerations. Efforts may be directed towards mitigating biases, ensuring fairness, and addressing the ethical implications associated with the use of predictive models in workforce management.

Greater Collaboration and Industry Standards:
Collaboration among industry stakeholders and the establishment of standards for predictive modeling in garment production may become more prevalent. This could facilitate the sharing of best practices, data, and insights for mutual benefit across the industry.
As the garment industry continues to evolve, the future scope of predictive modeling is poised to play a pivotal role in shaping decision-making processes, improving operational efficiency, and promoting innovation. By staying at the forefront of technological advancements and addressing emerging challenges, the industry can harness the full potential of predictive modeling to thrive in a rapidly changing global landscape.

13. APPENDIX

SOURCE CODE :- https://drive.google.com/file/d/15-sfA72vXU4yiIGV0UXe7IhGvI7YtuOG/view?usp=sharing

GITHUB :- https://github.com/smartinternz02/SI-GuidedProject-609710-1698167863

PROJECT DEMO :- https://drive.google.com/file/d/1SkGK8rFDwuwyD4DbWWQjEYRFgDTtTPtS/view?usp=sharing