Project Report - Phase 1: Initialization and Planning

Author: Abhay **Date:** August 5, 2025 **Project:** Employee Productivity Prediction

1.1. Introduction and Project Vision

This document marks the commencement of the Employee Productivity Prediction project. The primary vision is to leverage machine learning to create a predictive tool that can accurately forecast the productivity of garment industry workers. In a sector where operational efficiency is a key driver of success, such a tool can provide invaluable insights for management, aiding in strategic planning, resource allocation, and performance optimization. This initial phase outlines the foundational steps taken to define the project's scope, objectives, and technical roadmap.

1.2. Problem Statement and Business Case

Problem: The garment manufacturing industry operates on tight margins and demanding schedules. The inability to reliably predict workforce productivity can lead to missed deadlines, inefficient resource allocation, and reduced profitability. This project addresses the critical need for a data-driven approach to forecast the actual_productivity of employees.

Business Case: By developing a predictive model, the project aims to deliver tangible business value, including:

- **Improved Production Planning:** More accurate forecasting allows for better scheduling and target setting.
- **Enhanced Resource Management:** Identifying factors that influence productivity can help in optimizing team composition and workflow.
- **Proactive Performance Management:** The tool can help identify teams or individuals who may require additional support or incentives.

1.3. Project Objectives and Scope

The objectives for this project are as follows:

- To conduct a thorough analysis of the provided garments_worker_productivity.csv dataset.
- To develop and compare multiple machine learning models to find the most accurate predictor.
- To build a functional web application that allows users to interact with the predictive model.
- To deploy this application to a public cloud platform, making it accessible for real-world use.

Scope: The project will encompass the entire machine learning lifecycle, from data exploration and preprocessing to model training, evaluation, and deployment as a web service.

1.4. Technology Stack Selection

The following technologies have been selected for this project:

- **Programming Language:** Python 3
- **Core Libraries:** Pandas, NumPy, Scikit-learn, XGBoost
- **Web Framework:** Flask
- **Deployment:** Render, with Gunicorn as the web server.

This stack was chosen for its robustness, extensive community support, and suitability for both machine learning and web development tasks.

Project Report - Phase 2: Data Collection and Preprocessing

Author: Abhay Date: August 6, 2025 Project: Employee Productivity Prediction

2.1. Data Sourcing and Initial Exploration

The foundation of this project is the garments_worker_productivity.csv dataset. This dataset contains 1,197 records and 15 distinct attributes related to the work of employees in a garment factory. An

initial exploratory data analysis (EDA) was conducted to understand the structure, data types, and statistical properties of the dataset. This involved examining the distribution of key variables and identifying any immediate data quality issues.

2.2. Data Cleaning and Transformation

To prepare the data for machine learning, several cleaning and transformation steps were performed:

- **Standardization of Categorical Data: The department column contained inconsistencies such as "sweing" and "finishing ", which were corrected to "sewing" and "finishing" respectively. This ensures that the model treats these categories correctly.**
- **Handling of Missing Values: The wip (work in progress) column was identified as having a significant number of missing values. For the final model, this column was dropped from the dataset to maintain data integrity.**
- **Feature Engineering: The date column was leveraged to create a new, more useful month feature. This was done to capture any potential seasonal effects on productivity. After this, the original date colusmn was removed.**

2.3. Categorical Data Encoding

Machine learning models require all input features to be numerical. Therefore, the categorical columns (quarter, department, and day) were converted into a numerical format. The Label Encoding technique was used for this purpose, as implemented in the training notebook. This method assigns a unique integer to each category within a feature.

2.4. Final Dataset Preparation

After the preprocessing steps, the dataset was finalized and prepared for the model development phase. The data was split into features (X) and the target variable (y, which is actual_productivity). This clean and structured dataset formed the basis for training and evaluating the machine learning models in the next phase.

Project Report - Phase 3: Model Development

Author: Abhay **Date:** August 8, 2025 **Project:** Employee Productivity Prediction

3.1. Model Selection and Rationale

For this regression task, three different machine learning models were selected for training and comparison:

- **Linear Regression:** Chosen as a simple baseline model to establish a benchmark for performance.
- **Random Forest Regressor:** An ensemble learning method that is known for its high accuracy and

ability to handle complex relationships in data.

- **XGBoost Regressor:** A highly optimized and powerful gradient boosting algorithm, which is often a top performer in machine learning competitions.

This selection provides a good range of models, from a simple baseline to more complex and powerful ensemble methods.

3.2. Training and Testing Methodology

The preprocessed dataset was split into a training set (80% of the data) and a testing set (20% of the data). This separation is crucial to ensure that the models are evaluated on data they have not seen before, providing an unbiased assessment of their performance. Each of the three models was trained on the same training data.

3.3. Model Performance Evaluation

The performance of each trained model was evaluated on the test set using the following standard regression metrics:

- **Mean Absolute Error (MAE):** This metric provides a straightforward measure of the average error of the model's predictions.
- **Mean Squared Error (MSE):** This metric penalizes larger errors more heavily.
- **R-squared (R^2) Score:** This is a key metric that indicates the proportion of the variance in the target variable that is predictable from the features. A higher R^2 score indicates a better fit.

3.4. Model Selection and Finalization

After comparing the evaluation metrics for all three models, the **XGBoost Regressor** was identified as the best-performing model, primarily due to its superior R^2 score. This indicates that it was the most accurate and reliable model for this particular dataset. The trained XGBoost model was then saved as a pickle file (best_model.pkl) so that it could be easily loaded and used in the web application without the need for retraining.

Project Report - Phase 4: Application Development and Deployment

Author: Abhay **Date:** August 9, 2025 **Project:** Employee Productivity Prediction

4.1. Web Application Development

A web application was developed using the **Flask** framework in Python to provide a user-friendly interface for the machine learning model. The application consists of a single-page interface with a form where users can input the various attributes of an employee.

- **Frontend:** The user interface was built with HTML and styled using **Tailwind CSS** to create a modern and responsive design.
- **Backend:** The Flask backend handles the form submission, preprocesses the user's input to

match the format expected by the model, and then uses the loaded .pkl model to make a prediction.

4.2. Application Logic and Flow

1. The user navigates to the application's URL.
2. They fill out the web form with the required employee data.
3. Upon clicking "Predict Productivity," the data is sent to the Flask server.
4. The server's Python code applies the same **Label Encoding** transformations to the categorical data as was done during model training.
5. The processed data is then passed to the XGBoost model for prediction.
6. The model returns a productivity score, which is then categorized as "High," "Medium," or "Low" and displayed back to the user on the web page.

4.3. Deployment to Render

The final application was deployed to the **Render** cloud platform to make it publicly accessible. The deployment process involved:

1. Pushing all project files (app.py, best_model.pkl, requirements.txt) to a GitHub repository.
2. Creating a new Web Service on Render and linking it to the GitHub repository.
3. Configuring the build command (pip install -r requirements.txt) and the start command (gunicorn app:app).

The use of a requirements.txt file ensures that all necessary Python libraries are installed in the deployment environment, and Gunicorn is used as a production-ready web server.

4.4. Conclusion and Future Work

This project successfully demonstrates the end-to-end process of building and deploying a machine learning application. The final deployed web service provides a valuable and accessible tool for predicting employee productivity.

Future enhancements could include:

- **Hyperparameter Tuning:** To further improve the model's accuracy.
- **Data Visualization:** Adding charts and graphs to the web interface to provide more insights.
- **Batch Predictions:** Allowing users to upload a CSV file to get predictions for multiple employees at once.