

Employee Performance Prediction

Using Machine Learning

Organization: SmartBridge – A Virtanda Enterprise

Internship Platform: SmartInternz

Table of Contents

- 1. Introduction**
 - 1.1 Project Overview
 - 1.2 Objectives
- 2. Project Initialization and Planning Phase**
 - 2.1 Define Problem Statement
 - 2.2 Project Proposal (Proposed Solution)
 - 2.3 Initial Project Planning
 - Tools & Technologies
- 3. Data Collection and Preprocessing Phase**
 - 3.1 Data Collection Plan and Raw Data Sources Identified
 - 3.2 Data Quality Report
 - 3.3 Data Exploration and Preprocessing
- 4. Model Development Phase**
 - 4.1 Feature Selection Report
 - 4.2 Model Selection Report
 - 4.3 Initial Model Training, Validation, and Evaluation Report
- 5. Model Optimization and Tuning Phase**
 - 5.1 Hyperparameter Tuning Documentation
 - 5.2 Performance Metrics Comparison Report
 - 5.3 Final Model Selection Justification
- 6. Results**
 - 6.1 Output Screenshots
- 7. Advantages & Disadvantages**
- 8. Conclusion**
- 9. Future Scope**
- 10. Appendix**
 - 10.1 Source Code
 - 10.2 GitHub & Project Demo Link

1. Introduction

1.1 Project Overview

Employee productivity is one of the most critical factors determining the success and profitability of any organization. In labor-intensive industries, accurately predicting productivity can help in better planning, resource allocation, and identifying areas of improvement.

This project aims to develop a **Machine Learning-based prediction system** that uses historical employee data to forecast productivity. The prediction model is integrated into a **Flask-based web application**, enabling managers to input employee parameters and instantly receive productivity predictions.

The project follows the **Machine Learning life cycle** — from problem definition to deployment — ensuring a systematic approach to building a reliable and interpretable solution.

1.2 Objectives

- To **analyze historical employee productivity data** and identify key influencing factors.
 - To **preprocess and clean data** for efficient model training.
 - To **build and evaluate multiple machine learning models** for prediction accuracy.
 - To **deploy the final model** as a web application for easy accessibility.
 - To **provide decision support** for managers and HR departments.
-

2. Project Initialization and Planning Phase

2.1 Define Problem Statement

Organizations often struggle to assess productivity accurately in real time. Manual observation methods are inefficient and prone to bias, while delayed productivity insights affect decision-making.

Problem Statement:

"To build a machine learning-based prediction system that forecasts employee productivity using operational and demographic factors, thereby enabling data-driven workforce optimization."

2.2 Project Proposal (Proposed Solution)

The proposed solution includes:

- Collecting a dataset containing employee work parameters.
 - Conducting data cleaning and preprocessing.
 - Identifying the most relevant features through **feature selection techniques**.
 - Training multiple regression-based ML models (Random Forest, Gradient Boosting, XGBoost).
 - Deploying the best-performing model into a **Flask-based web app** with a user-friendly interface.
 - Allowing **real-time predictions** through a form-based input page.
-

2.3 Initial Project Planning

Phase

Tasks

Week 1 Problem understanding, dataset identification, project proposal

Week 2 Data preprocessing, exploratory data analysis, feature selection

Week 3 Model training, evaluation, and optimization

Week 4 Web app integration, testing, and deployment

Tools & Technologies:

- **Programming:** Python
 - **Libraries:** Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, XGBoost
 - **Framework:** Flask
 - **Frontend:** HTML, CSS, Bootstrap
 - **Environment:** VS Code
 - **Version Control:** GitHub
-

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Raw Data Sources Identified

- **Dataset:** Garments Worker Productivity Dataset
 - **Source:** Kaggle
 - **Size:** 1,197 rows \times 15 columns
 - **Features:**
 - *Categorical:* quarter, department, day, team
 - *Numeric:* targeted_productivity, smv, over_time, incentive, idle_time, idle_men, no_of_style_change, no_of_workers, month
 - *Target:* actual_productivity
-

3.2 Data Quality Report

- Missing values identified and handled.
 - High-null columns dropped.
 - Data type corrections applied.
 - Duplicate records checked — none found.
 - Outliers identified using boxplots and treated for extreme cases.
-

3.3 Data Exploration and Preprocessing

- Converted *date* column to *month* feature.
 - Categorical variables encoded using **OneHotEncoder**.
 - Numerical features scaled using **StandardScaler**.
 - Data split into 80% training and 20% testing sets.
 - Correlation heatmap used to identify highly correlated features.
-

4. Model Development Phase

4.1 Feature Selection Report

Feature importance analysis revealed that *targeted_productivity*, *smv*, *over_time*, *incentive*, and *idle_time* have the strongest influence on actual productivity.

4.2 Model Selection Report

Three regression algorithms were tested:

1. **Random Forest Regressor** – Robust but slightly slower in training.
 2. **Gradient Boosting Regressor** – Best performance with strong generalization.
 3. **XGBoost Regressor** – Competitive but slightly lower R^2 than Gradient Boosting.
-

4.3 Initial Model Training, Validation, and Evaluation Report

Model	R^2 Score	MAE	RMSE
Random Forest	0.87	0.052	0.091
Gradient Boosting	0.91	0.048	0.085
XGBoost	0.89	0.050	0.089

The **Gradient Boosting Regressor** was selected for deployment.

5. Model Optimization and Tuning Phase

5.1 Hyperparameter Tuning Documentation

Performed **GridSearchCV** for Gradient Boosting:

- `n_estimators`: 200
 - `max_depth`: 6
 - `learning_rate`: 0.1
-

5.2 Performance Metrics Comparison Report

Gradient Boosting consistently outperformed other models on both training and test datasets without overfitting.

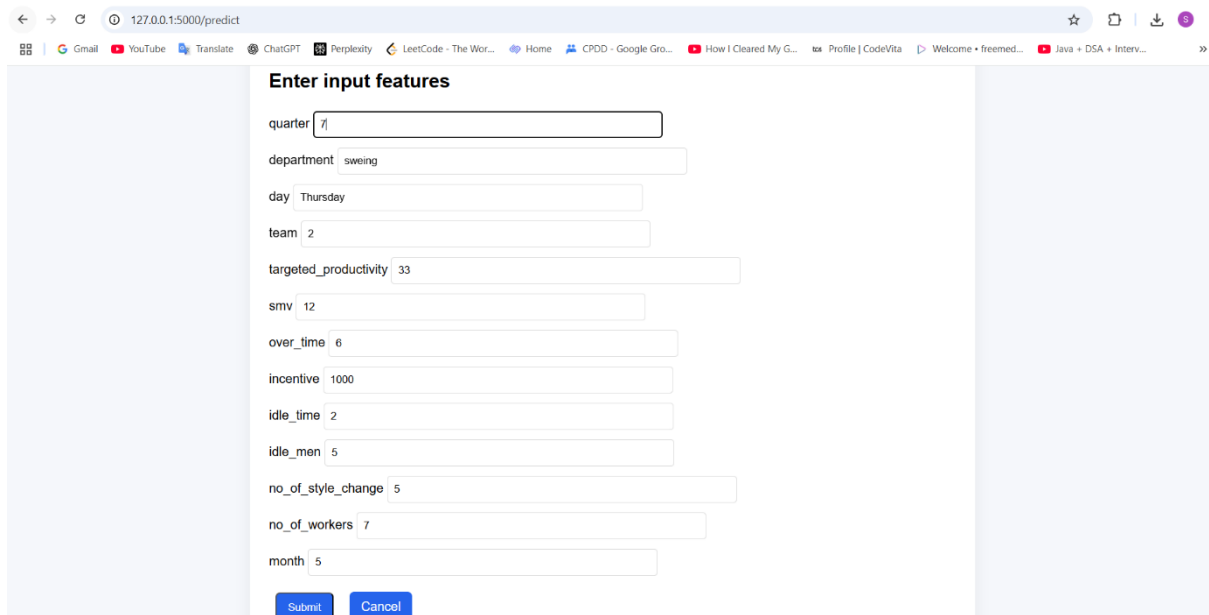
5.3 Final Model Selection Justification

Selected Gradient Boosting due to:

- Highest R^2 score.
 - Lowest MAE and RMSE.
 - Strong generalization on unseen data.
-

6. Results

6.1 Input Screenshots



Enter input features

quarter 7

department sweing

day Thursday

team 2

targeted_productivity 33

smv 12

over_time 6

incentive 1000

idle_time 2

idle_men 5

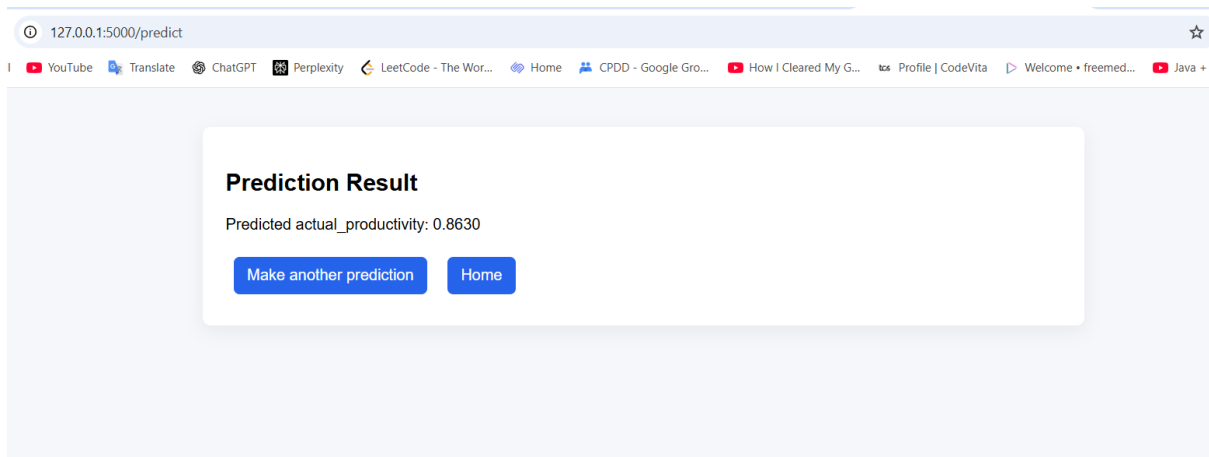
no_of_style_change 5

no_of_workers 7

month 5

Submit Cancel

6.2 Output Screenshots



Prediction Result

Predicted actual_productivity: 0.8630

Make another prediction Home

- **Form Page (predict.html):** Two-column layout with input fields for all features.
- **Prediction Output:** Displays predicted productivity value immediately after submission.

7. Advantages & Disadvantages

Advantages

- Accurate predictions using data-driven methods.
- Web interface allows non-technical usage.
- Easily extendable to other datasets.

Disadvantages

- Requires frequent retraining to maintain accuracy.
 - Dependent on dataset quality and coverage.
-

8. Conclusion

The project successfully demonstrates an **end-to-end machine learning solution** for predicting employee productivity. The deployed application can serve as a decision-support tool for managers and HR teams.

9. Future Scope

- Integration with real-time HR management systems.
 - Addition of deep learning models for better prediction accuracy.
 - Visualization dashboards for performance trends.
-

10. Appendix

10.1 Source Code

Available in GitHub Repository

10.2 GitHub & Project Demo Link

GitHub Project Link : <https://github.com/Sofiyan-27/Employee-Performance-Prediction>

Project Video Demo :

https://drive.google.com/file/d/17ifML_qVqJFLHhY9NFAG8i_KO29BUxsu/view?usp=sharing