

Final Project Report

1. Title Page

Project Name: Employee Productivity Prediction Using XGBoost

Prepared by: [Keshav Sharma]

Date: [08-08-2025]

2. Introduction

- This project aims to predict employee productivity using machine learning—specifically, the XGBoost regressor.
- Predicting performance helps organizations optimize resource allocation, improve output, and enhance workforce efficiency.

3. Objectives

- Develop an accurate regression model to forecast productivity.
- Evaluate model using MSE, MAE, and R^2 metrics.
- Deploy the trained model with a Flask web application.
- Provide a user-friendly front-end for input and prediction delivery.

4. Dataset and Features

You use 13 predictor features:

1. **quarter**
2. **department**
3. **day**
4. **team**
5. **targeted_productivity**
6. **smv**
7. **over_time**
8. **incentive**
9. **idle_time**
10. **idle_men**
11. **no_of_style_change**
12. **no_of_workers**
13. **month**

Target variable: Actual productivity values.

5. Methodology

- **Data Preprocessing:** Handle missing values, convert data types, and potentially scale features or encode categoricals.
- **Model:** XGBoost Regressor with `n_estimators=100`, `learning_rate=0.1`, `max_depth=5`.
- **Training & Evaluation:** Split data into training/testing, train the model, and compute MSE, MAE, R^2 .
- **Web Deployment:** Build a Flask backend with a front-end form using Bootstrap (for a professional UI), accept user inputs, run predictions, and display results.

6. Results (Sample)

test_mse: 0.02021150707977389

test_mae: 0.09780533089943558

R2_score: 0.3332611345864712

Include plots:

- Actual vs Predicted productivity
- Feature importance from XGBoost

7. Web App Description

- **app.py**: Flask app loading the `gwp.pkl` model, handling routes, and returning prediction results.
- **Templates**:
 - `home.html`: Landing page with navigation.
 - `about.html`: Overview of the project.
 - `predict.html`: Bootstrap form for input.
 - `submit.html`: Displays prediction.
- **Model**: `gwp.pkl` (trained with XGBoost).
- **Requirements**: `requirements.txt` includes Flask, NumPy, XGBoost, scikit-learn.
- Optionally include additional assets (CSS/images) under a `static/` folder.

8. Conclusion

- XGBoost regression can effectively model employee productivity when trained properly.
- The Flask-based web app provides an accessible interface for users to get real-time predictions.
- Future enhancements:
 - More features (e.g., skill levels, weather, time of year)
 - Model comparison with Random Forest, LightGBM
 - Deploy on platforms like Heroku or Render

Reference Reports and Related Research

Here are several relevant studies and reports for your references:

1. **Forecasting Office Workers' Productivity**

A Machine Learning model using physiological, behavioral, and psychological features; XGBoost achieved $R^2 = 0.60$ and MAE = 10.52

[MDPI](#)

2. **Data Analytics for Optimizing and Predicting Employee Performance**

Explores preprocessing, feature selection, model tuning for employee performance prediction models

[MDPI](#)

3. **Machine Learning-Based Prediction of Employee Performance Using Lifestyle Indicators**

Framework to predict daily productivity based on nutrition and lifestyle data

[ResearchGate](#)

4. **Comparative Analysis of CatBoost vs. XGBoost for Employee Turnover Prediction**

Highlights that CatBoost can outperform XGBoost under certain conditions

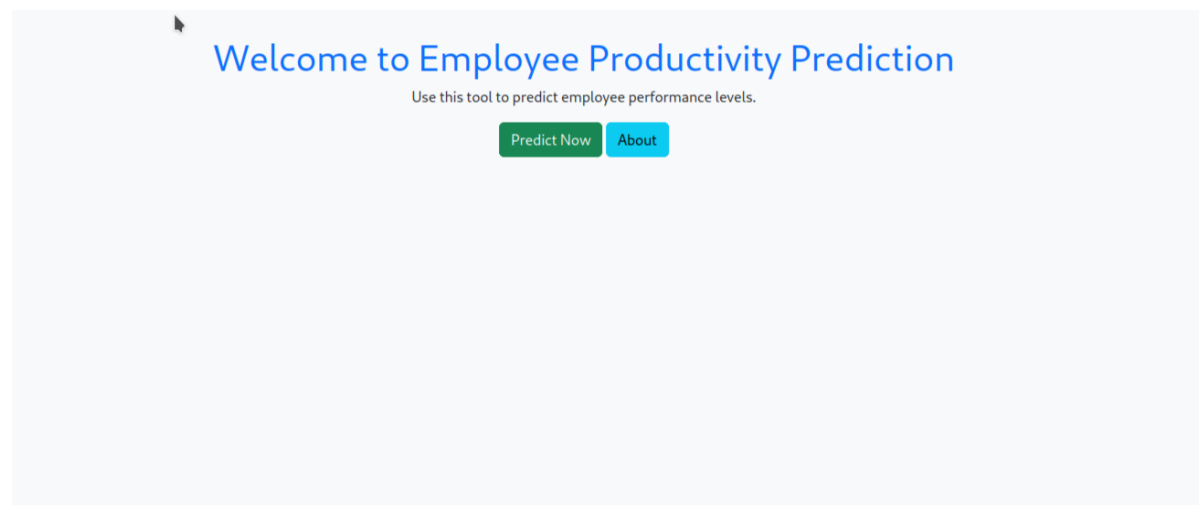
[ResearchGateDergiPark](#)

5. **Predicting Employee Turnover Using XGBoost (IJSET)**

Demonstrates the robustness of XGBoost for attrition prediction in HR contexts

[ijiset.com](#)

Output



About This Project

This application uses a machine learning model (XGBoost) to predict employee productivity levels based on various input factors.

[Back to Home](#)

Enter Employee Details

Enter Employee Details

Prediction Result

The employee is medium productive.

[Back to Home](#)

