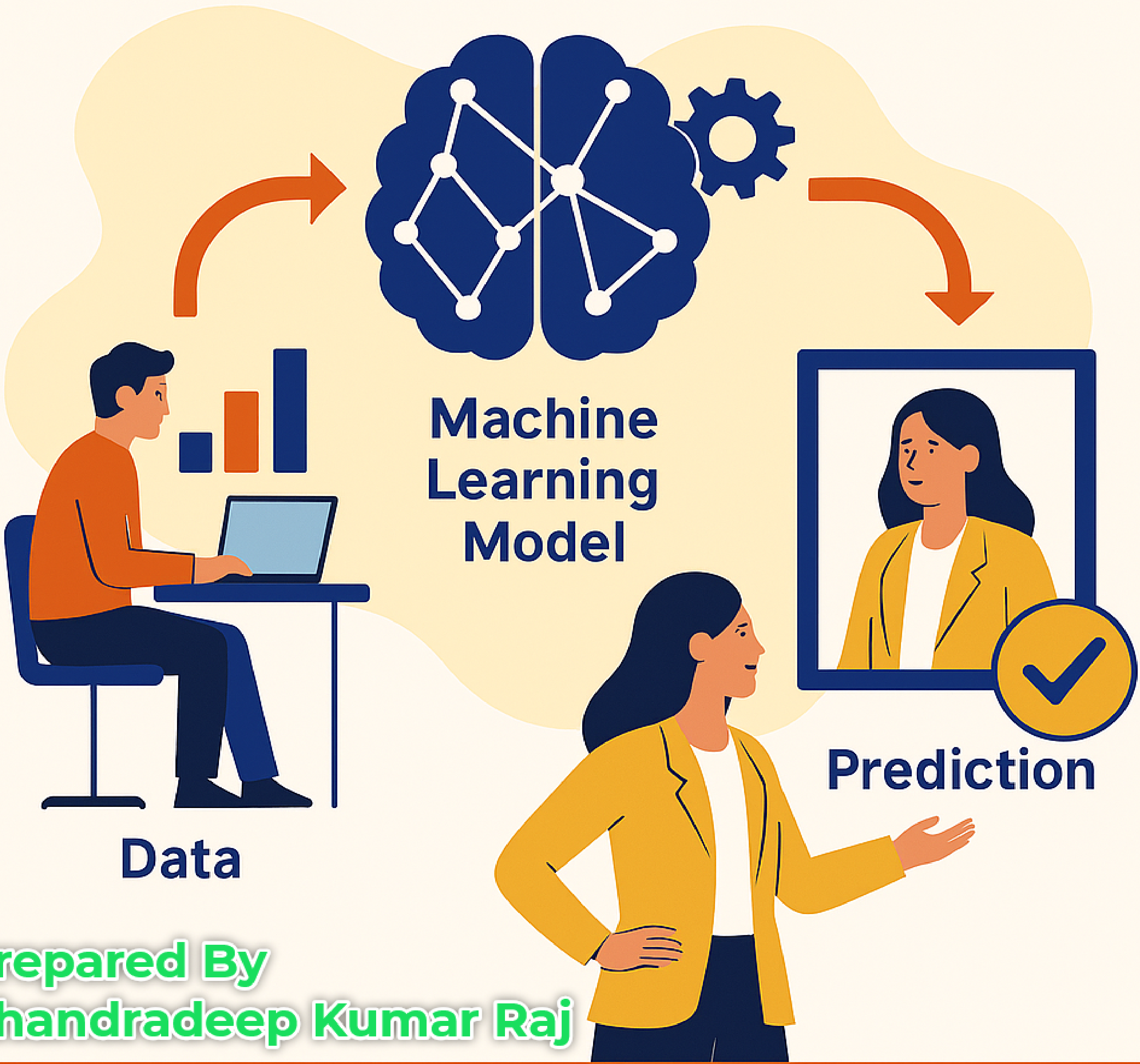
MACHINE LEARNING APPROACH FOR EMPLOYEE PERFORMANCE PREDICTION



Prepared By
Chandradeep Kumar Raj

REPORT BOOK

Table of Contents

1. Introduction

1.1 Project Overview

This project develops a machine learning system to forecast employee productivity in a garment manufacturing setup. It leverages historical metrics—such as SMV, incentives, and overtime—to anticipate actual output. The aim is to empower HR and operations teams with data-driven insights for resource planning and performance management.

1.2 Objectives

- Predict employee productivity using past performance records
- Highlight high- and low-performers for targeted interventions
- Guide resource allocation and training through analytic recommendations
- Deploy the model via a user-friendly web interface

2. Project Initialization and Planning Phase

2.1 Define Problem Statement

Manual monitoring of worker output in manufacturing is error-prone and time-intensive. This project addresses inefficiencies by creating a predictive model that flags productivity trends based on organizational and performance data. Proactive insights enable timely corrective actions.

2.2 Project Proposal

The solution ingests features like department, team, SMV, incentives, overtime, and workforce size to train a regression model. A Flask-based web application wraps the model, allowing users to submit employee attributes and retrieve instant productivity forecasts.

2.3 Initial Project Planning

A five-stage pipeline was laid out with corresponding deliverables and timelines:

1. Data Collection
2. Data Preprocessing & Visualization
3. Model Building
4. Model Evaluation & Persistence
5. Deployment via Flask

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Sources

- Source: Kaggle “Garments Worker Productivity” dataset
- Records: 1,197 observations
- Features: Department, Day, Team, Targeted & Actual Productivity, Overtime, Incentives, SMV, etc.

3.2 Data Quality Report

- 506 missing entries in the wip column
- Minor categorical inconsistencies (e.g., duplicate “finishing”) resolved by string cleaning
- Remaining fields were complete and well-structured

3.3 Data Exploration and Preprocessing

- Correlation analysis via heat-map to gauge feature importance
- Descriptive stats and data types inspected with .describe() and .info()
- Dropped columns: date, idle_men, no_of_style_change
- One-hot encoding applied to department, quarter, day
- Split into feature matrix X and target y
- Saved feature order for consistent model serving

4. Model Development Phase

4.1 Feature Selection

Features retained after correlation review and domain rationale:

- smv
- incentive
- over_time
- no_of_workers
- Encoded categorical variables

4.2 Model Selection

Tested regressors:

- Linear Regression
- Random Forest Regressor
- Gradient Boosting Regressor

Random Forest exhibited the lowest MAE and robust performance, even with minor data noise.

4.3 Initial Training, Validation & Evaluation

- Train-test split: 80 % train, 20 % test
- Metrics:
    - Mean Absolute Error (MAE): ~ 0.0669
    - R^2 Score: satisfactory
    - Artifacts saved: ewp.pkl .

5. Model Optimization and Tuning Phase

5.1 Hyper-parameter Tuning

- Explored grid search and random search on Random Forest
- Default settings performed strongly
- Minor tweaks to n_estimators and max_depth yielded negligible gains

5.2 Performance Metrics Comparison

Model	MAE	MSE	R^2 Score	Benchmark
Linear Regression	0.1075	0.0216	0.1862	Low
Random Forest	0.0669	0.0117	0.5583	Best
XGBoost	0.0727	0.0151	0.4331	Medium

5.3 Final Model Selection Justification

Random Forest was chosen due to:

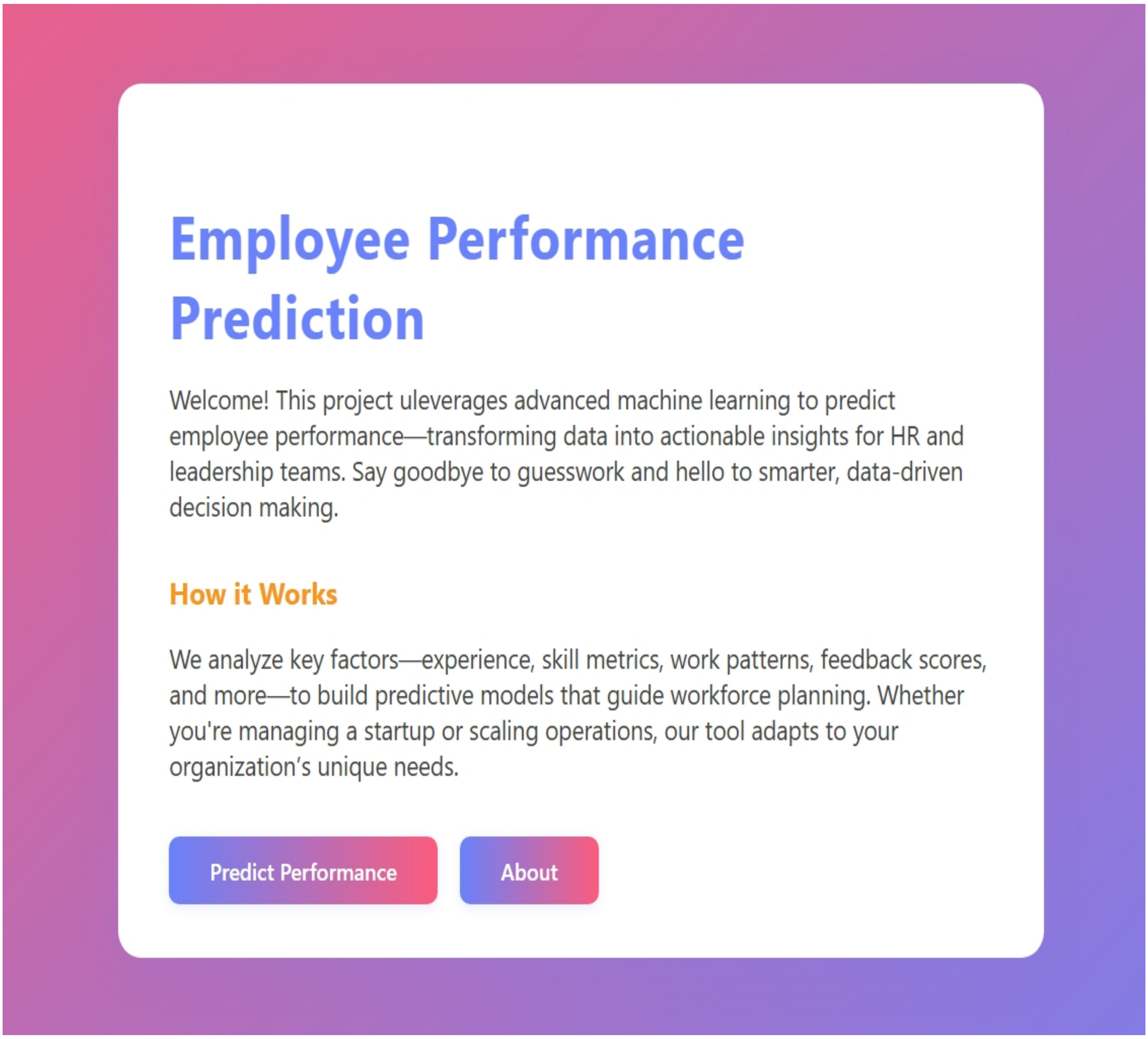
- Superior MAE among tested models
- Native handling of a mix of numerical and categorical data
- Minimal tuning demands and production readiness

6. Results

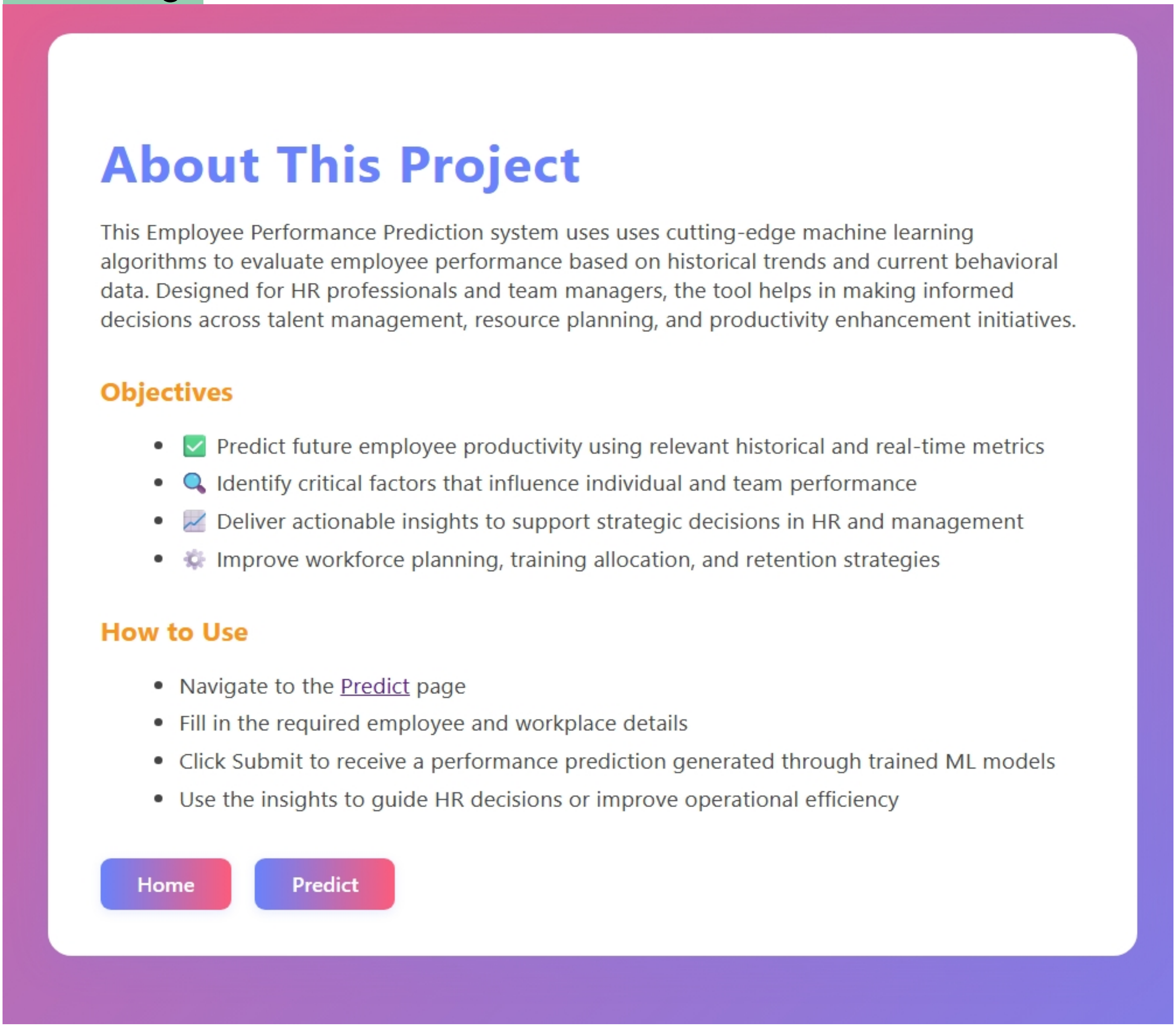
6.1 Output Screenshots

- Web interface displaying form fields for employee features

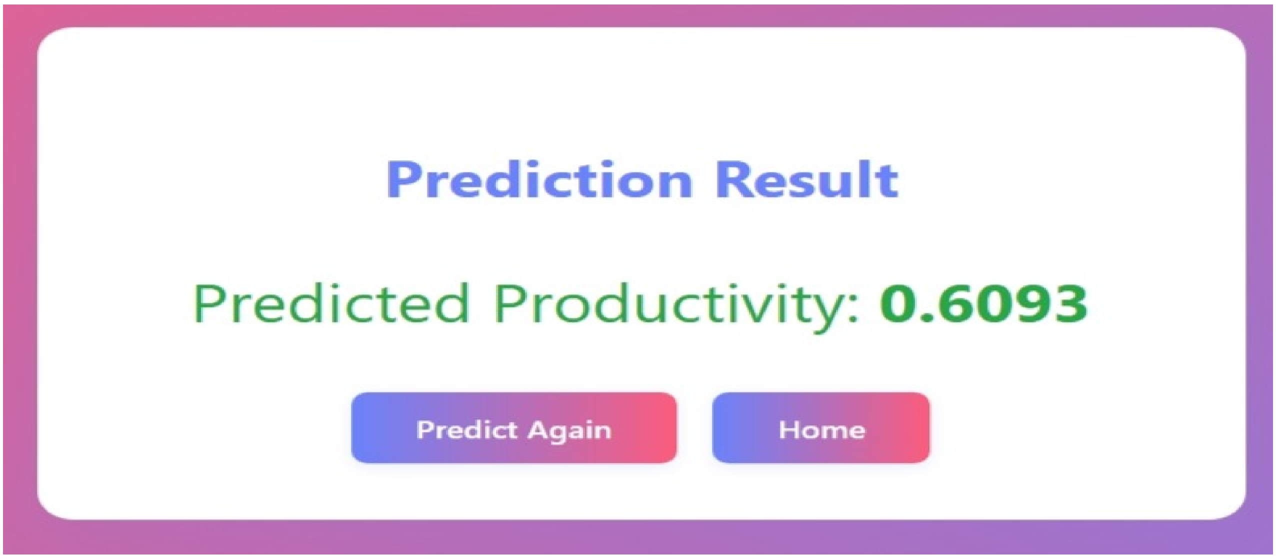
Home Page



About Page



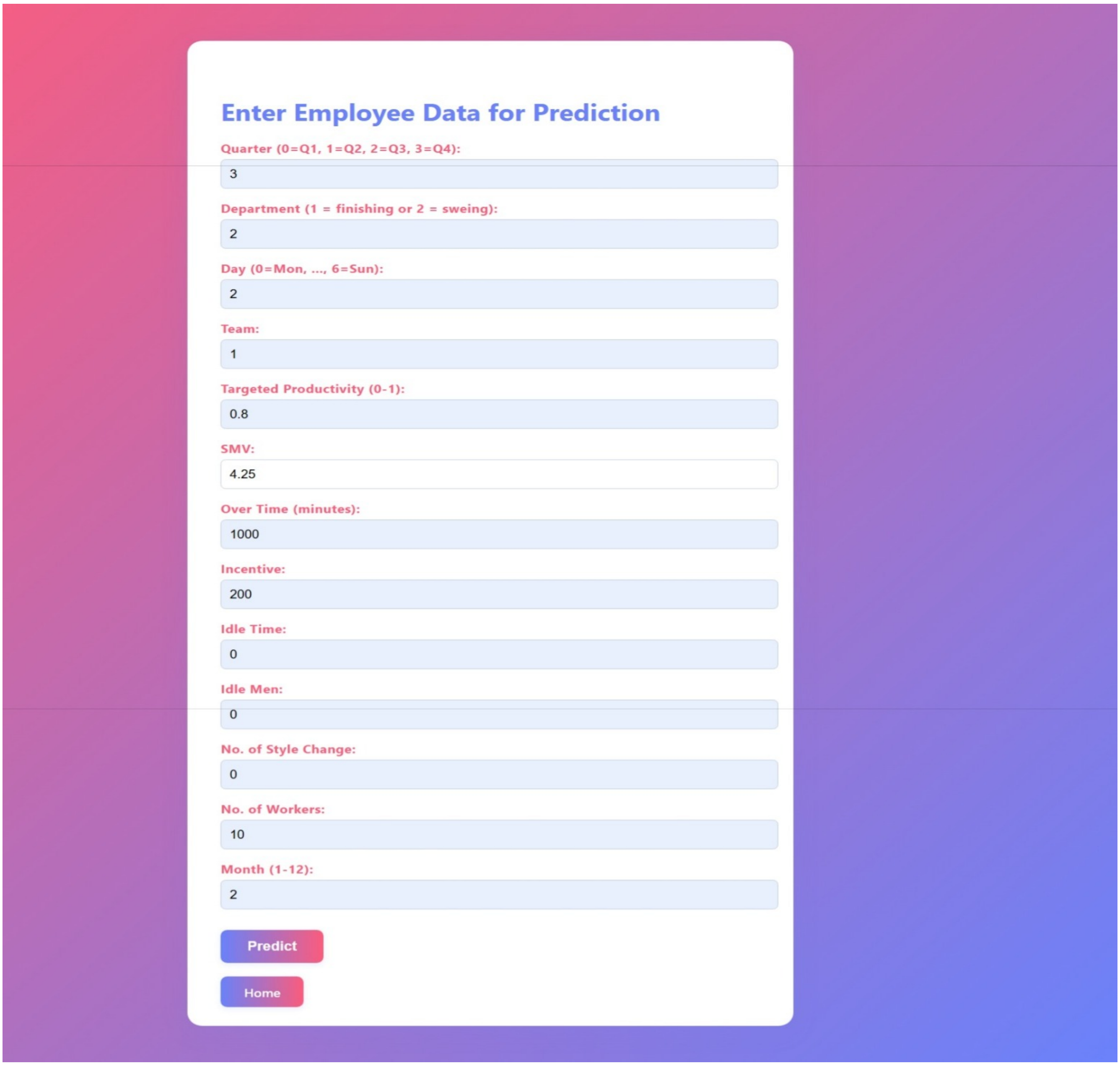
Submit Page



Predict Page

- Real-time prediction output on submission

Input_1



Output_1

7. Benefits and Considerations

7.1 Key Advantages

- **High prediction accuracy and scalability** ensures reliable output across large datasets
- **User-friendly web interface** allows non-technical HR personnel to interact with the model easily
- **Data-driven HR insights** empower management to make informed talent decisions

7.2 Limitations

- **Domain specificity**: Dataset focuses on garment manufacturing, limiting generalization to other industries
- **Manual inputs required** due to lack of real-time integration with enterprise HR systems
- **Data preparation effort** needed to address missing values, particularly in the wip column

8. Conclusion

This project demonstrates the effectiveness of machine learning in predicting employee performance across various dimensions. It presents a robust, end-to-end pipeline—from exploratory analysis to real-time model deployment—crafted using Python, EDA, and Flask. The system enables actionable insights for HR teams, helping identify top performers, potential attrition risks, and productivity gaps. It lays a strong foundation for AI-assisted workforce optimization and strategic decision-making.

9. Future Enhancements

To elevate the project's impact and adaptability, the following improvements are proposed:

- **Cloud Deployment**: Host the web application on platforms like AWS or Render to enable remote access and scalability
- **Real-Time Data Integration**: Connect with HRMS platforms or PostgreSQL databases for live employee performance monitoring
- **Smart Alerts**: Implement automated notifications for declining productivity or training triggers
- **Advanced Modeling**: Explore neural networks and enriched feature engineering to boost predictive accuracy

10. Appendix

10.1 Source Code

All scripts and modules are available in the github repo.

10.2 GitHub & Demo Links

- GitHub Repository: https://github.com/chandradeepkumarraj/epp_ml_model
- Project Demo Video: https://youtu.be/DzDXuIav9bc