

Project Report - Phase 4: Application Development and Deployment

Author: Aryan Rai **Date:** August 7, 2025 **Project:** Employee Performance Prediction using a Machine Learning Approach

4.1. Web Application Development

The web application development phase transforms the machine learning model into a practical, user-friendly tool. This process involves building both the frontend and backend components of the application.

- **Frontend:**
 - The frontend is the user interface of the application, built using HTML templates (as seen in the project's file structure).
 - It provides a simple webpage where users can input employee data into a form and submit it for a prediction.
 - This part of the application is what the user sees and interacts with.
- **Backend:**
 - The backend is the core logic that powers the application, primarily handled by the app.py file.
 - It uses the Flask framework to manage requests from the frontend.
 - The backend's main tasks include:
 1. **Loading the Model:** It loads the trained best_model.pkl file into memory.
 2. **Processing Data:** It takes the data submitted by the user from the frontend form.
 3. **Making Predictions:** It uses the loaded model to make a performance prediction based on the user's data.
 4. **Sending Response:** It sends the prediction result back to the frontend to be displayed to the user.
- **Deployment:**
 - The requirements.txt file ensures that the deployment environment (Render) has all the necessary libraries, including Flask and Gunicorn, to run the application correctly.
 - Gunicorn acts as the web server that hosts the backend logic, making the entire application accessible online.

4.2. Application Logic and Flow

- **Initialization:** The `app.py` script loads the `best_model.pkl` file when the application starts to prepare for predictions.
- **User Interface:** A Flask-rendered HTML template (e.g., `home.html`) provides a form for user input.
- **Data Processing:** The `app.py` script receives and processes the user's data after form submission, preparing it for the model.
- **Prediction:** The processed data is sent to the loaded machine learning model, which returns a predicted performance level.
- **Response:** The prediction result is displayed to the user via a new HTML page rendered by Flask.
- **Dependencies:** The `requirements.txt` file ensures all necessary libraries (Flask, etc.) are installed on the Render deployment platform.

4.3. Deployment to Render

- The project code, including the `best_model.pkl` file, is managed in a Git repository connected to **Render**.
- **requirements.txt** lists all dependencies (e.g., Flask, Gunicorn), which Render automatically installs.
- The "Start Command" uses **Gunicorn** to run the Flask application, making it accessible online.
- The entire process is automated, taking the project from code to a live, functional service.

4.4. Conclusion and Future Work

Conclusion

- This project successfully demonstrated the application of machine learning to predict employee performance, providing an objective alternative to traditional, subjective evaluation methods.
- The web application, built using Flask and deployed on Render with Gunicorn, successfully integrates the trained `best_model.pkl` to offer a practical and accessible tool for HR teams.
- The project serves as a proof of concept that data-driven insights can significantly enhance strategic decision-making in human resource management.

Future Work

- **Model Improvement:** Explore more advanced models, such as deep learning architectures, to potentially increase prediction accuracy.
- **Feature Engineering:** Integrate more dynamic data points, such as real-time project completion rates, team collaboration metrics, or sentiment analysis from internal communications.
- **Interactive Dashboard:** Develop a more comprehensive and interactive dashboard that allows HR managers to visualize performance trends, identify potential risks, and track the impact of interventions.
- **Expand Scope:** Extend the model to predict other key HR metrics, such as employee turnover risk, training effectiveness, or salary optimization.
- **Scalability:** Optimize the application for scalability to handle larger datasets and a greater number of simultaneous users, making it suitable for larger enterprises.