Employee Performance Prediction

Project Overview

This project aims to predict the productivity of garment industry workers using machine learning techniques. It includes data analysis, model building in a Jupyter notebook (`ML.ipynb`), and a user-friendly prediction interface built with Streamlit (`app.py`).

Files & Structure

- ***ML.ipynb***

  Main notebook for data exploration, preprocessing, model training, and evaluation.

- ***app.py***

  Streamlit web app for making predictions using the trained model.

- ***garments_worker_productivity.csv***

  Dataset containing employee productivity records.

- ***model.pkl***

  Saved machine learning model for deployment.

- ***scaler.pkl****

Saved feature scaler for preprocessing input data.

Workflow

1. **Data Analysis & Model Building (`ML.ipynb`)**

- Load and explore the dataset.
- Clean data: drop irrelevant columns, handle missing values.
- Visualize relationships between features and productivity.
- Select relevant features for modeling.
- Split data into training and test sets.
- Scale features using `StandardScaler`.
- Train models (Linear Regression, KNN, SVR).
- Evaluate models using Mean Squared Error and R^2 Score.
- Save the best model and scaler for deployment.

2. Prediction Web App (`app.py`)
- Load the trained model and scaler.
- Provide a web interface for users to input employee data.
- Preprocess user input and predict productivity.
- Display prediction results.

How to Run

1. Jupyter Notebook
- Open `ML.ipynb` in Jupyter or VS Code.
- Run cells to perform data analysis and train models.

2. Streamlit App
- Make sure `model.pkl` and `scaler.pkl` are present.
- Run the app:
  streamlit run app.py
- Enter employee details to get productivity predictions.

Requirements

- Python 3.7+
- Libraries: numpy, pandas, matplotlib, scikit-learn, joblib, streamlit

---

References

- Dataset: garments_worker_productivity.csv
- Main notebook: ML.ipynb
- Web app: [app.py](app.py)

For details, see the code in `ML.ipynb` and `app.py`.
