High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **PolyReg - Polynomial Regression Visualizer**. PolyReg is a Jupyter-based educational tool enabling users to interactively explore polynomial regression, adjust model degrees, visualize fits, and compare regression metrics (MAE, MSE, R^2) on synthetic or uploaded datasets.

---

1. System Architecture Overview

Architecture Description:
PolyReg is implemented as a modular Jupyter Notebook application. It leverages Python libraries for data handling, machine learning, and visualization. The system is composed of distinct modules for data input, preprocessing, model training, visualization, and metrics computation.

Main System Modules

Module	Role
Data Input	Load synthetic or user-uploaded datasets (CSV).
Data Preprocessing	Clean, validate, and prepare data for modeling.
Model Training	Fit polynomial regression models of user-selected degrees.
Visualization	Plot data points, regression curves, and residuals interactively.
Metrics Computation	Calculate and display MAE, MSE, and R^2 for model evaluation.
User Interface (UI)	Jupyter widgets for user interaction (degree selection, file upload).

---

2. Component Interactions

Sequence Step	Interaction Description
1. User Input	User uploads data or selects synthetic dataset via UI widgets.
2. Data Preprocessing	Data is validated and preprocessed for modeling.
3. Model Training	Polynomial regression model is trained with selected degree.
4. Metrics Computation	Model predictions are evaluated; MAE, MSE, R^2 are computed.
5. Visualization	Data, regression curve, and metrics are visualized interactively.
6. User Iteration	User adjusts parameters (e.g., degree), triggering re-training and updates.

---

3. Data Flow Overview

Source	Data/Action	Destination	Purpose
User	Dataset selection/upload	Data Input Module	Provide data for analysis
Data Input	Raw data	Preprocessing	Clean and validate data
Preprocessing	Processed data	Model Training	Prepare data for regression
Model Training	Trained model, predictions	Metrics, Visualization	Evaluate and display results
Metrics	Computed metrics	Visualization, UI	Show model performance
UI	User parameter changes	All modules	Trigger updates and re-computation

---

4. Technology Stack

Layer/Function	Technology/Framework
Notebook Platform	Jupyter Notebook
Programming Language	Python 3.x
Data Handling	pandas, numpy
Machine Learning	scikit-learn
Visualization	matplotlib, seaborn, ipywidgets
User Interaction	ipywidgets

---

5. Scalability & Reliability

- **Scalability:** Designed for interactive, single-user educational use within Jupyter; not intended for large-scale or concurrent usage.
- **Reliability:** Relies on mature, widely-used Python libraries; input validation and error handling ensure robustness.
- **Security:** Operates within the user's Jupyter environment; no external data transmission.

End of Document