



Low Level Design Document

Introduction

This Low Level Design (LLD) document details the implementation plan for **PolyReg - Polynomial Regression Visualizer**, a Jupyter-based educational tool for interactively exploring polynomial regression, visualizing model fits, and comparing regression metrics on synthetic or user-uploaded datasets.

1. System Components

Component	Description	Key Responsibilities
DataManager	Handles data loading, validation, and preprocessing	Load/upload data, generate synthetic data
RegressionEngine	Performs polynomial regression and computes metrics	Fit model, predict, calculate MAE/MSE/R ²
Visualizer	Generates interactive plots and displays metrics	Plot data, regression curve, metrics
UIController	Manages Jupyter widgets and user interactions	Degree selection, file upload, trigger update

2. Class/Interface Overview

Class/Interface	Key Methods/Attributes	Relationships
DataManager	<code>load_data()</code> , <code>generate_synthetic()</code> , <code>get_data()</code>	Used by RegressionEngine, Visualizer
RegressionEngine	<code>fit(degree)</code> , <code>predict(X)</code> , <code>compute_metrics()</code>	Uses DataManager data
Visualizer	<code>plot_data()</code> , <code>plot_fit()</code> , <code>display_metrics()</code>	Uses RegressionEngine outputs
UIController	<code>setup_widgets()</code> , <code>on_change()</code> , <code>update_view()</code>	Orchestrates all components

3. Data Structure Overview

Data Model	Structure/Type	Description
Dataset	<code>pandas.DataFrame</code> with columns <code>X</code> , <code>y</code>	Input data for regression
Model Coefficients	<code>numpy.ndarray</code>	Polynomial regression coefficients
Metrics	<code>dict</code> with keys: <code>mae</code> , <code>mse</code> , <code>r2</code>	Regression performance metrics



4. Algorithms/Logic

Polynomial Regression Flow:

```
# Pseudocode
data = DataManager.load_data() # or generate_synthetic()
X, y = data['X'], data['y']
model = RegressionEngine.fit(X, y, degree)
y_pred = model.predict(X)
metrics = RegressionEngine.compute_metrics(y, y_pred)
Visualizer.plot_data(X, y)
Visualizer.plot_fit(X, y_pred)
Visualizer.display_metrics(metrics)
```

5. Error Handling

Scenario	Handling Approach
Invalid file format/upload	Show error message, prompt re-upload
Missing/NaN values in data	Warn user, offer to clean or impute
Degree out of valid range	Restrict input, show validation error
Regression fit failure	Display error, suggest lower degree
Widget interaction errors	Log error, reset affected UI components

End of Document