High-Level Design (HLD) for Automated Machine Learning Solution

# Abstract

The increasing accessibility of machine learning (ML) and the proliferation of data have made automated ML solutions a key focus in modern technological advancements. However, many non-expert users struggle with implementing end-to-end machine learning workflows due to complexities in data preprocessing, model selection, and deployment. This project addresses these challenges by developing an automated machine learning system that empowers users to simply upload their datasets and specify the type of problem (classification or regression). The system performs comprehensive data analysis, hyperparameter tuning, and model training, delivering the best-performing model. In addition, users can access descriptive statistics and visualizations to gain insights into their data. To ensure real-world applicability, the system allows seamless deployment of models on cloud platforms such as AWS and GCP. This solution simplifies the ML pipeline, reduces barriers to entry, and enhances accessibility for non-expert users while maintaining scalability and accuracy.

Contents

[Abstract 1](#_heading=h.gjdgxs)

[Introduction 2](#_heading=h.30j0zll)

[Why this High-Level Design Document? 2](#_heading=h.1fob9te)

[The HLD will: 3](#_heading=h.3znysh7)

[Scope 3](#_heading=h.2et92p0)

[Definitions 3](#_heading=h.tyjcwt)

[General Description 5](#_heading=h.3dy6vkm)

[Product Perspective 5](#_heading=h.1t3h5sf)

[Problem Statement 5](#_heading=h.4d34og8)

[Proposed Solution 5](#_heading=h.2s8eyo1)

[Further Improvements 5](#_heading=h.17dp8vu)

[Technical Requirements 5](#_heading=h.3rdcrjn)

[Tools and Technologies 6](#_heading=h.26in1rg)

[Hardware Requirements 6](#_heading=h.lnxbz9)

[Constraints 6](#_heading=h.35nkun2)

[Assumptions 6](#_heading=h.1ksv4uv)

[Design Details 7](#_heading=h.44sinio)

[Event log 7](#_heading=h.2jxsxqh)

[Error Handling 7](#_heading=h.z337ya)

[Conclusion 8](#_heading=h.3j2qqm3)

# Introduction

## Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to provide a comprehensive framework for the development of an automated machine learning (AutoML) solution. It bridges the gap between the conceptual project description and the detailed implementation, ensuring a clear and cohesive pathway for coding and development. This document serves to identify potential inconsistencies early, streamline the development process, and act as a reference manual for the interaction of modules at a high level.

## The HLD will:

• Define all aspects of the system design in detail.

• Describe the user interface and its functionalities.

• Specify hardware and software interfaces.

• Outline performance and scalability requirements.

• Detail the design features and overall architecture of the project.

• Address non-functional attributes such as:

- Security

- Reliability

- Maintainability

- Portability

- Reusability

- Resource utilization

- Application compatibility

- Serviceability

# Scope

This HLD document provides an overview of the system's architecture, including:  
• Data Handling: Processes for data ingestion, preprocessing, and storage.  
• Application Flow: Navigation through the web application, from data upload to model deployment.  
• Technology Architecture: The tools, platforms, and frameworks employed in the development process.  
• Cloud Integration: Details of the deployment options available on AWS and GCP.

The document employs a blend of non-technical and mildly technical terminology, making it accessible to both developers and stakeholders responsible for system administration.

# Definitions

| Term | Description |
| --- | --- |
| AutoML Solution | An end-to-end platform for automating machine learning tasks. |
| Artifacts | Directory for storing intermediate files, analysis, and models. |
| AWS/GCP | Cloud platforms used for model deployment and scalability. |
| IDE | Integrated Development Environment used for coding and debugging. |

# General Description

## Product Perspective

The Automated Machine Learning (AutoML) solution is designed to simplify and streamline the machine learning workflow, enabling users with minimal technical expertise to build, analyze, and deploy ML models. The system automates critical steps, including data preprocessing, model selection, hyperparameter tuning, and deployment, while providing intuitive data visualizations and descriptive insights.

## Problem Statement

Developing machine learning solutions requires extensive domain knowledge and technical expertise, presenting a significant barrier for non-expert users. The objective of this project is to create an end-to-end AutoML platform with the following capabilities:  
- Allow users to upload datasets in standard formats and specify problem types (classification or regression).  
- Perform automated data preprocessing, exploratory data analysis, and model training.  
- Select and fine-tune the best-performing machine learning model.  
- Enable seamless deployment of the trained model to cloud platforms such as AWS or GCP.

# Proposed Solution

The proposed AutoML solution simplifies the ML pipeline by automating data handling, analysis, and model optimization.  
1. Data Analysis: Automatically perform data validation, preprocessing, and visualization to ensure data readiness for modeling.  
2. Model Training: Identify the best algorithm based on the problem type, optimize hyperparameters, and train the model for maximum performance.  
3. Visualization: Present data insights and model performance metrics through graphical representations.  
4. Deployment: Offer one-click deployment of models on AWS or GCP for scalability and real-world application.

# Further Improvements

• Incorporating more advanced problem types, such as time-series forecasting or anomaly detection.  
• Expanding deployment options to include platforms like Azure or on-premises solutions.  
• Integrating real-time model monitoring and retraining pipelines for continuous improvement.

# Technical Requirements

• The system supports datasets of varying sizes, including large-scale datasets. No strict minimum or maximum number of rows or features is enforced; the platform is designed to handle datasets dynamically.  
• Handles missing values, outliers, and categorical data transformations automatically.  
• Acceptable formats include CSV and Excel for seamless ingestion.

# Tools and Technologies

• Programming Language: Python.  
• Libraries and Frameworks: NumPy, Pandas, Scikit-learn, TensorFlow, Matplotlib, Seaborn.  
• Cloud Platforms: AWS (S3, EC2) and GCP (Google Cloud Storage, Compute Engine).  
• IDE: PyCharm or Jupyter Notebook.  
• Database: MySQL or MongoDB.  
• Frontend: HTML, CSS, and JavaScript for the user interface.  
• Backend: Flask for API development.

# Hardware Requirements

• Reliable storage for handling large datasets and model artifacts.  
• Internet connection for cloud-based deployment and data fetching.

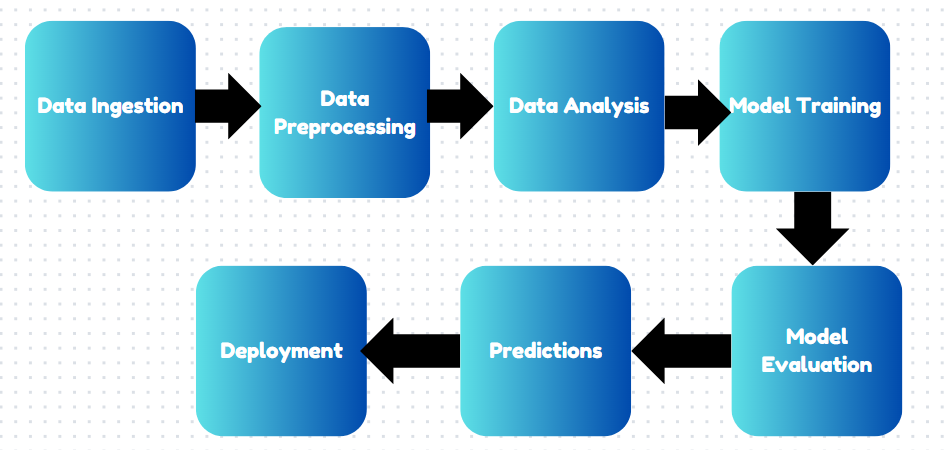
# Constraints

• The system must be user-friendly, requiring no coding skills from users.  
• Execution time for preprocessing and model training should remain within a reasonable limit.  
• Deployment pipelines should support real-time data processing in cloud environments.

# Assumptions

• The datasets provided by users are clean enough to support automated preprocessing.  
• Cloud platforms (AWS/GCP) are configured with the required credentials for seamless integration.  
• Users have basic understanding of problem types (classification or regression) to specify their requirements accurately

# Design Details



# Event log

The system should log every event so that the user will know what process is running internally.  
Initial Step-By-Step Description:  
1. The System identifies at what step logging required  
2. The System should be able to log each and every system flow.  
3. Developer can choose logging method. You can choose database logging/ File logging as well.  
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

# Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

# Conclusion

For the AutoML project, the designed system allows users to upload datasets, choose the problem type (regression or classification), and automatically receive the best hyper-tuned model tailored to their data. This end-to-end solution streamlines the model selection, training, and deployment process, making machine learning accessible to both technical and non-technical users. By providing insightful data analysis, visualization, and deployment options on cloud platforms like AWS or GCP, the system facilitates efficient and accurate predictions. Ultimately, this project aims to empower users to make data-driven decisions quickly and effectively, fostering a seamless machine learning experience.