High Level Design (HLD)

Adult Census Income Prediction

Revision Number: 1.0

Last date of revision: 23/10/2022

Author: Dibyendu Biswas.

Contents:

1. Objective ---------- 03
2. Introduction ---------- 03
3. Project Architecture ---------- 03
4. Project Architecture Description ---------- 04
    I. Users Web Application ---------- 04
    II. Online Web Portal ---------- 04
    III. Transactional Databases ---------- 05
5. System Architecture ---------- 05 - 06
    I. Data Collection ---------- 06
    II. Exploratory Data Analysis (EDA) ---------- 06
    III. Data Pre-processing/ Features Engineering ---------- 06
    IV. Applying ML Algorithms ---------- 06
    V. Predictions ---------- 06
6. Technical Stack ---------- 07
7. Deployment Strategy ---------- 07

Objective

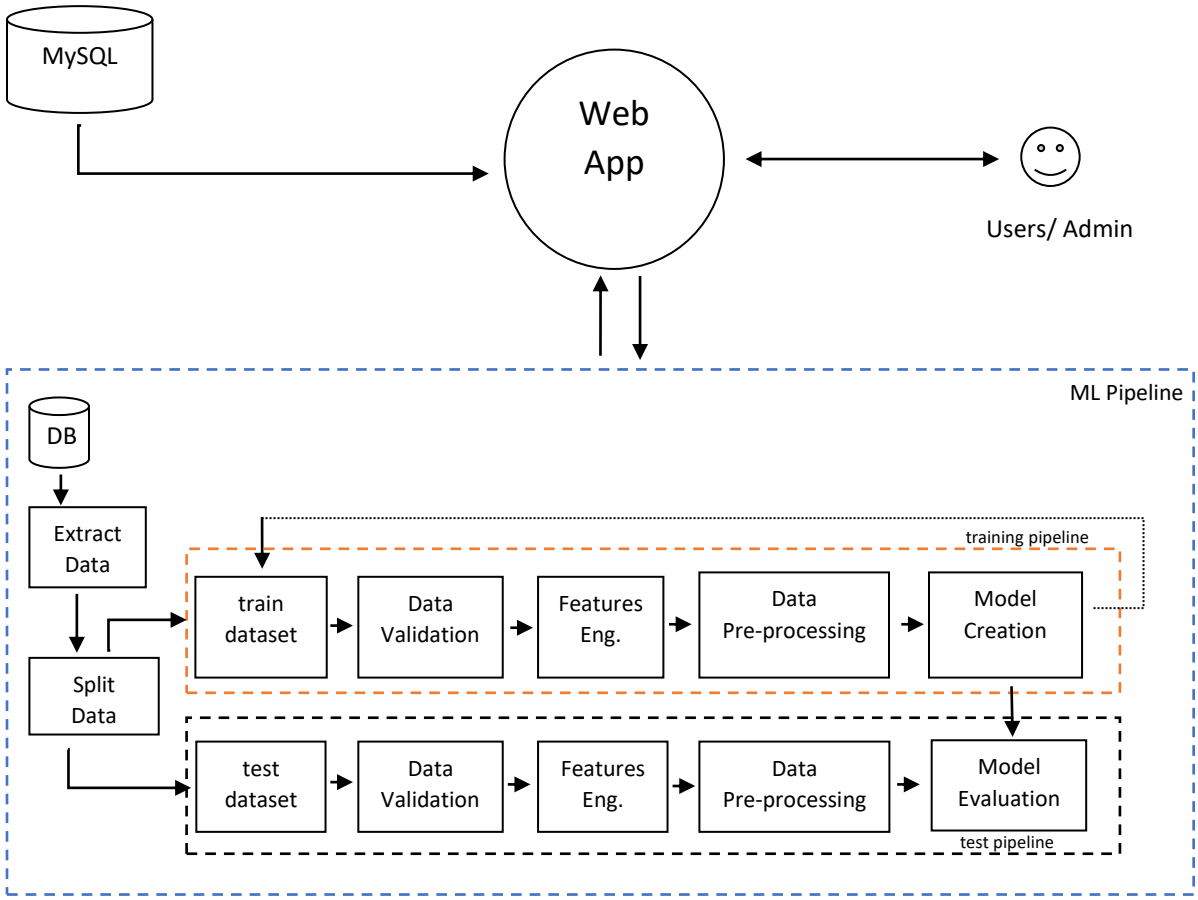
The objective of this document is to provide the high-level overview of the entire software solution for the **Adult Income Prediction** system. It describes end to end steps involved in designing web applications and CI/CD pipeline.

The main objective of this project is to find out whether a person has an income of more than 50k in a year or not.

Introduction

In this project, we will use a number of different supervised algorithms to precisely predict individuals' income using adult data Set collected from the UCI machine learning repository. We will then choose the best candidate algorithm from preliminary results and further optimize this algorithm to best model the data. Our goal with this implementation is to build a model that accurately predicts whether an individual makes more than $50,000.

Project Architecture

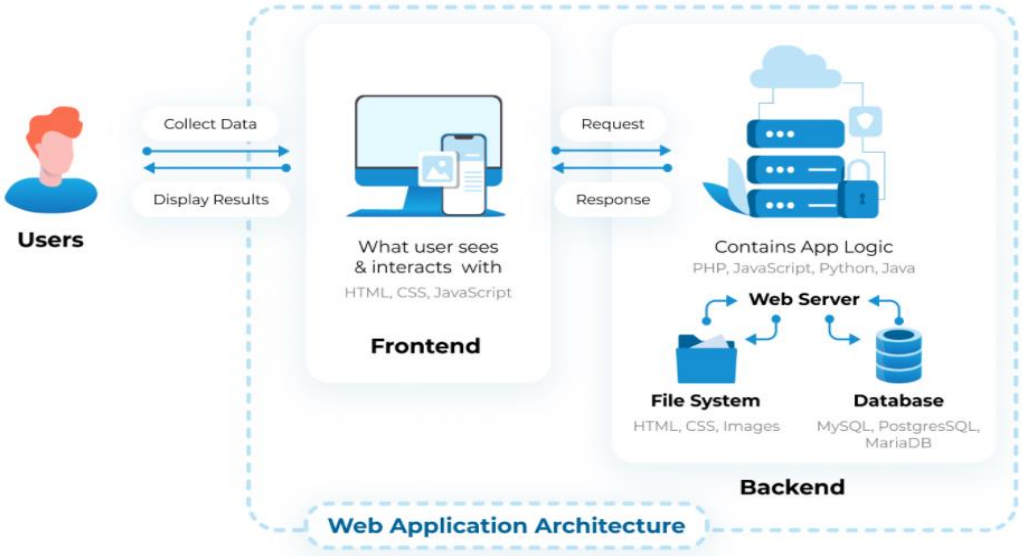


Project Architecture Description

The overall project is divided broadly into following section –

- Users Web Application
- Online Web Portal
- Transactional Databases

User Web Application & Online Web Portal

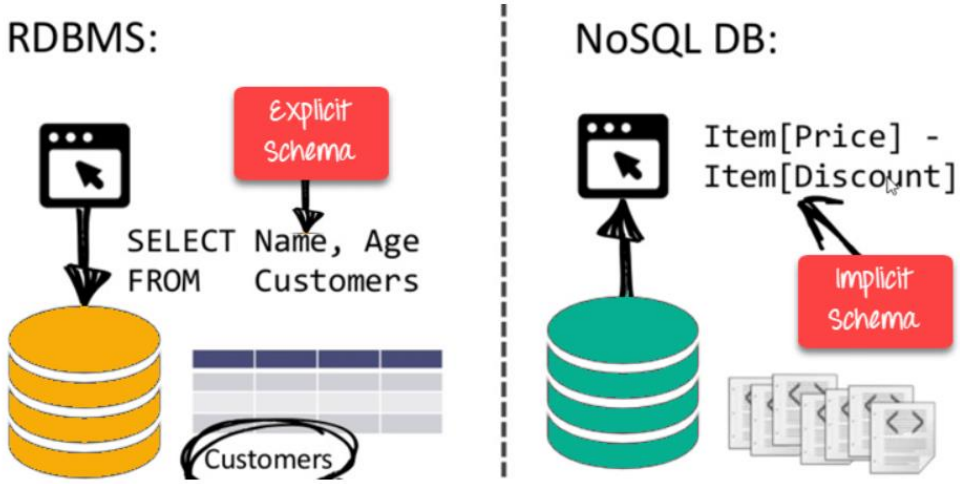


This section is a web portal which will be a front end for users.

Transactional Databases

The transactional database setup is a crucial step in the whole architecture. This is decided based on the nature of the data and content which we are going to store in the database and warehouse for analysis.

We could choose structure relational database, i.e., MySQL.



NoSQL is Schema-Free

System Architecture Description

- Data Collection
- Exploratory Data Analysis (EDA)
- Data Pre-processing/ Features Engineering
- Applying ML Algorithms
- Predictions

Data Collection

Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes.

To feed the data inside databases, we can use predefined dataset.

Link of dataset: https://www.kaggle.com/overload10/adult-census-dataset

Exploratory Data Analysis (EDA)

This is a very crucial step to perform to understand the nature of the data on which we are planning to build a model.

Here, we are going to perform both statistical and graphical approaches to understand the data into its depth.

We will do all sorts of analysis including univariate, bivariate and multivariate analysis. We will consider all numerical and categorical features.

An exploratory data analysis is an approach of analysing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modelling or hypothesis testing task.

Data Pre-processing/ Features Engineering

Feature engineering is the process of using domain knowledge to extract features (characteristics, properties, attributes) from raw data. Features are individual independent variables that act as inputs in a machine learning system. Features are properties of a problem for which we would like to predict results. In simplistic terms, one column of a data set can be one feature. In a more real-world scenario, you would obtain training features from existing features using a method known as “feature engineering.”

In this section, we are going to engineer our features so that it can be supplied to the next step in the ML pipeline. It includes dropping a few unwanted columns, deriving some useful features, merging multiple data sources.

Applying ML Algorithms

This is a Classification kind of model, so for this we apply Decision Tree Classifier, Random Forest Classifier, and Boosting & Bagging techniques.

Prediction

At this stage, prediction will be created from the best model and the processed output will display through portal.

Technical Stack

Below is the technical stack we are going to use for this project.

- **Programming Language:** Python (NumPy, pandas, scikit-learn, flask, matplotlib, seaborn, pyplot, plotly, etc)
- **Database:** MySQL
- **IDE:** PyCharm IDE
- **Visualization Tools:** jupyter notebook, power bi.
- **Front-End:** HTML, CSS, JavaScript.
- Git, GitHub, DVC, etc.

Deployment Strategy

The deployment of machine learning models is the process of making models available in production where web applications, enterprise software and APIs can consume the trained model by providing new data points and generating results/outcomes for business Decisions.

Deployment using a ML pipeline on Cloud (AWS/GCP/Azure)

These are 3 leading cloud platforms where we will deploy our web application. We will choose any of the below platforms for our use based on some factors.

---------- Thank You ----------