Project Report

Adult Census Income Prediction

Written By **Document Version**	**Himanshu Kumar Jha,** 0.2
Last Revised Date	17-October-2021

Document Control

Change Record:

Version	Date	Author	Comments
0.1	08-Oct-2021	Himanshu Kumar Jha	Introduction and Architecture
0.2	10-Oct-2021	Himanshu Kumar Jha	Architecture & Architecture Description appended and updated
0.3	13-Oct-2021	Himanshu Kumar Jha	Model Deployment and Column Categorical Changes
0.4	15-Oct-2021		

Reviews:

Version	Date	Reviewer	Comments
0.1	17 Aug 2021	Himanshu	Document Content , Version Control and Unit Test Cases to be added

1. Introduction

1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2. Architecture

3. Architecture Description

3.1. Data Description

Adult Census Dataset is the data set of bank customers with various parameters

3.2. Project Study

Studying the project with understanding the parameters

3.3. Sorting Data

Finding out null values in the data set for removal and checking out the unique categories for every feature in the data set in order to find out any invalid entries for the given parameters

3.4. Cleansing Data

Removing the null values found in the data set and invalid symbols that have no meaning in the categories

3.5 In placing the Index

In placing, the index as the removal of null or invalid variable removed causes empty values in various rows, and categorizing the salary into two categories (<=50k and >50k) in order to make the data set sequential and ready for further steps.

3.6 Comparing the parameters with Salary

Comparing the parameters using cat-plot with salary in order to understand the relationship and the impact of parameters on Salary.

3.7 Dropping Insignificant Parameters

Dropping the parameters that do not have considerable amount of impact on salary, the lesser number of parameters it is going to have a better accuracy for the model.

3.8 Grouping of countries

As the number of categories in the country attribute is too much, so making some general categories accordingly in order to lesser the amount of categories

3.9 Mapping of categories

Mapping of the Education parameter in order to have a numeric values with respect to the increase in education qualification. These numeric values are required in order to run the model.

3.10 EDA

Exploratory Data Analysis of the whole data set comparing the data X attributes with the Y attribute, checking out relations and patterns.

3.11 Finding Outliers

Finding the outliers to check out the impact on the Mean, Median and Mode

3.12 Correlation

Checking out the Correlation values for all the parameters, to measure the strength of the relationship between two variables and direction of a linear relationship between two variables.

3.13 Data Scaling

Data Scaling or Data Preprocessing is used to normalize the range of independent variables or features of data. The collected data set contains features highly varying in magnitudes, units and range. If scaling is not done then algorithm only takes magnitude in account and not units hence incorrect modelling. To solve this, we do scaling to bring all the variables to the same level of magnitude.

3.14 Model training

Training model on various Algorithms to check for best accuracy and selection of a final Algorithm

3.15 Hyper Parameter Tuning

3.16 Model Selection

4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected results
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to user

Verify whether the user is able to sign up in the application	1. Application URL is accessible 2. Application is deployed	The application should load completely for the user when the URL is accessed
Verify whether the User is able to sign up in the application	1 Application is accessible	The user should be able to sign up in the application
Verify whether user is able to successfully login to the application	1. Application is accessible 2. User is signed up to the application	User should be able to successfully login to the application
Verify whether user is able to see input fields on logging in	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to see input fields on logging in
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance to the selections user made	1. Application is accessible 2. User is signed up to the application	The recommended results should be in accordance to the selections user made

	3. User is logged in to the application	
Verify whether user has options to filter the recommended results as well	1. Application is accessible	User should have options to filter the recommended results as well
	2. User is signed up to the application 3. User is logged in to the application	
Verify whether KPIs modify as per the user inputs for the user's health	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	The KPIs should indicate details of the suggested recipe