Low Level Design

Adult Census Income Prediction

Written By	Sachin Kumar
Document version	0.1
Last Revised date	08-06-2022

Document Control

Change Record:

Version	Date	Author	Comments
0.1	08-06-2022	Sachin Kumar	Introduction & Architecture Description appended and updated

Contents

1. Introduction

1.1. What is Low-Level design document?

1.2. Scope

2. Architecture

3. Architecture Description

3.1. Data Description

3.2 Data Validation

3.2. Data Transformation

3.4. Data Insertion into Database

3.5. Export Data from Database

3.6. Data Pre-processing

3.7. Data Clustering

3.10. Model Building

3.11. Hyper parameter tuning

3.12. Model saving

3.13. cloud setup

3.14. Pushing application to cloud

3.15. Model Call for Specific Cluster

3.16. Deployment

1. Introduction

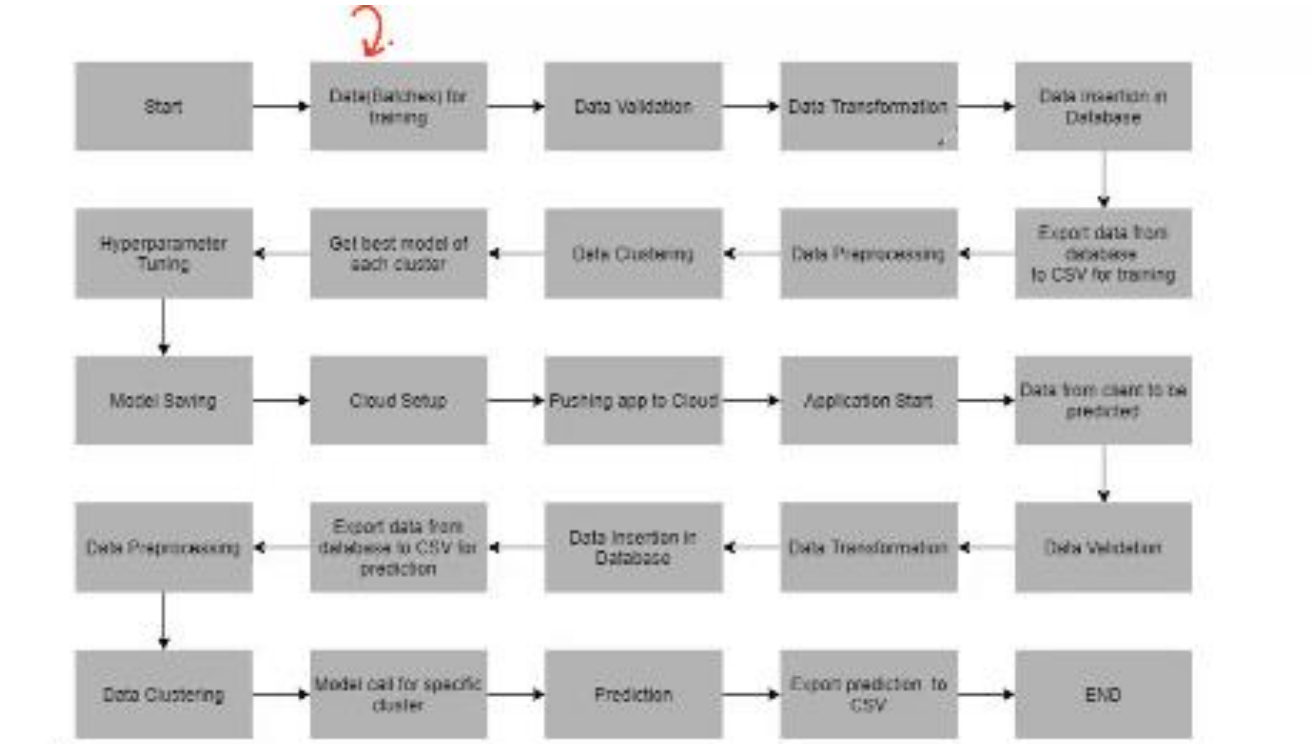
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Adult Census Income Prediction 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 Income Prediction dataset will be provided from client side , data will be kept over their shared location and from that shared location we are going to read those files. The dataset contains 1 label column and 14 different features. Also, the dataset contains 32562 records which is sufficient for making a model.

3.2. Data Validation

Here Data Validation will be done, given by the client. We accept the data as per client said agreement.

We check filename convention, no of columns etc and make sure the data received, is in correct format as per the agreement else data will be rejected and moved to bad data folder.

3.3 Data Transformation

Once we have filtered out the correct format of data sent by the client, will do a data transformation.

There are certain things we need to do before inserting the data in the local database – Ex – Missing values should be replaced by Null, categorical values in single quotes will be converted into double quotes, etc.

3.4. Data Insertion into Database

- Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
- Table creation in the database.
- Insertion of files in the table

3.5. Export Data from Database

Data Export from Database - The data in a stored database is exported as a CSV file to be used for Data Pre-processing and Model Training.

3.6. Data Pre-processing

Data Pre-processing steps that could be used are Null value handling, conversion of categorical values to numerical values, balance and imbalance dataset handling, can perform standard normal distribution, punctuation removal and several other steps also can be performed for cleaning the data.

3.7. Data Clustering.

K-Means algorithm will be used to create clusters in the pre-processed data. The optimum number of clusters is selected by plotting the elbow plot. The idea behind clustering is to implement different algorithms to train data in different clusters. The K-means model is trained over pre-processed data and the model is saved for further use in prediction.

3.8 Model Building

After clusters are created, we will find the best model for each cluster. For each cluster, algorithms will be passed with the best parameters derived from Grid-Search. We will calculate the AUC scores for the models and select the model with the best score. Similarly, the models will be selected for each cluster. All the models for every cluster will be saved for use in Recommendation.

3.9 Hyper parameter tuning

Using hyper parameter tuning we are going to enhance the performance of the model which we already selected and those models which perform best for a given cluster by comparing their AUC scores. After that will get a model for individual clusters which performs the best.

3.10 Model saving

Once we get the best model will save it to our local system.

3.11 cloud setup

Once we have selected our model and done with tuning part we start cloud setup process. Here, before pushing the application to cloud we need to do certain changes, we need to add some files which are required for our application to correctly run over cloud platform. For ex- run time.txt,

requirements.txt etc.

3.12 Pushing app to cloud

Once we are done with cloud setup steps an application is created and pushed to cloud. Post cloud platform itself will start the application and keep the application running.

Once the cloud deployment is done, again we can use our model for prediction. The same steps will be followed which were followed during training. The client will keep their data in a shared location and from that shared location we are going to read those files, after we have read the files again we need to perform the same sets of validations that were done in case of training and after the files pass the validation will perform certain transformations the same we did in training approach like missing values handling, single quotes replaced by double quotes and so on.

Post we will insert the data into the prediction database, so basically, we will be having 2 databases one for training and the other for prediction. Now from the prediction database again we will export the data to a CSV file and this CSV file will act as an input for our actual prediction.

Once we have the CSV file will perform a data cleaning approach whatever we did for cleaning in case of training same will be applied in case of prediction. Once our data is cleaned will pass it to the clustering algorithm that we have saved and it will give the cluster numbers to which the data belongs according to the row, which means which row numbers belong to which cluster number this approach will tell us.

Now we are going to segregate all the data based on their cluster numbers and after this whatever model we had prepared for a certain cluster that model will be used to do the prediction for that particular cluster data. So now the prediction will be done and will keep all the predictions in a CSV file. This CSV file will be saved in a shared drive from where the client will be able to read it and perform operations based on the requirement using this prediction data.