

Low Level Design

Customer Churn Prediction

Written By	Vishal Gujarathi
Date	2 Dec 2022

Document Control

Change Record:

Version	Date	Author	Comments

Reviews:

Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

Content	Page No
1. Introduction	1
1.1 What is Low-Level Design Document	1
1.2 Scope	1
2 Architecture	2
3.1 Architecture Description	3
3.2 Data Description	3
3.3 Data Gathering	3
3.4 Data Cleaning	3
3.5 Handling Missing Data	3
3.6 New Feature Generation	3
3.7 Feature Selection	4
3.8 Encoding Categorical Data	4
3.9 Parameter Tuning	4
3.10 Model Building	4
3.11 Model Saving	4
3.12 GitHub	4

1. Introduction

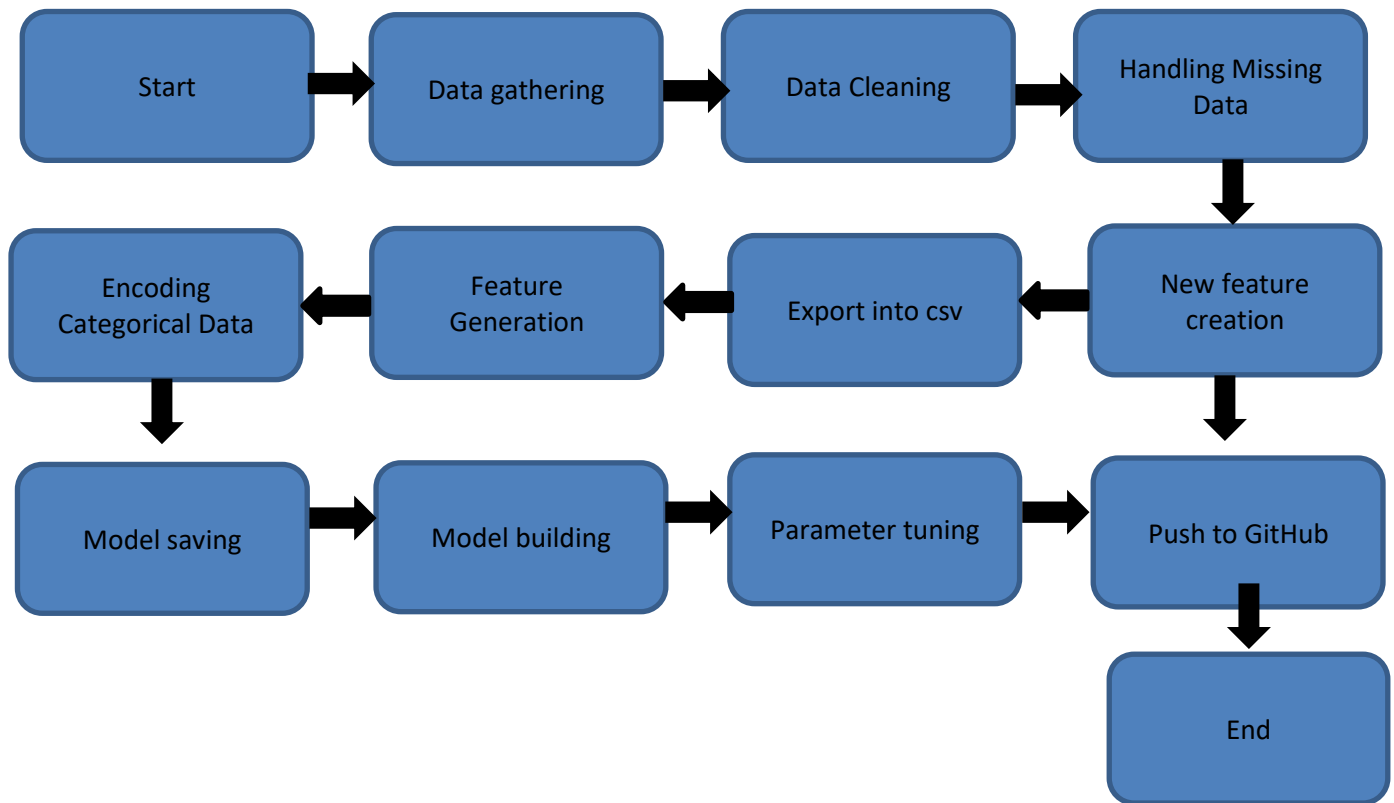
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Ad-Click Prediction. LLD describes the class diagrams with their 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

Independent variables:-

['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges']

Dependent variables:-

['Churn']

3.2. Data Gathering

Dataset link: - [Link](#)

We got csv files train. We use these files for the data preparation purpose and EDA.

3.3. Data Cleaning

I Observe that the datatypes of each column, number of rows present with non-null values, there are 2 int, 1 float, and remaining are string datatype columns.

Second, we check the description of the dataset, here we will only visible the num variables functionalities. we will use describe () method. Describe() method only describe the functionalities of a numerical variable. From this, we can easily conclude the parameters of each column. We drop customerID because it has no meaning in the dataset and we can easily differentiate each customer using indices of the rows. By dropping this column our dataset should be now ready to process.

3.4. Handling Missing Data

To handle null values we fill null values of the **TotalCharges** column with the mean of the **TotalCharges** column.

3.5. Feature Generation

We observed that there was numerical and categorical feature so that we encoded the categorical feature and create new feature in the dataset for better accuracy of the model.

3.6. Feature Selection

We included all the features for model except customerID.

3.7. Encoding Categorical Data

So, here we have to handle categorical columns, handle means we have to convert categorical values into numerical values because while the training model dataset contains all the numerical values categories won't be accepted.

On observing the count values of the dataset then we found that there are NO and YES are present, so we have to convert it into 1 and 0 which will be easy to process. For all categorical variables, we replace **Yes** with 1 and **No** with 0.

In the gender column, we replace **Male** with 1 and **Female** with 0.

Now we are importing LabelEncoder from the sklearn which will decode categorical values into numeric ones.

All the categorical columns are now typed cast into the numerical values.

The handling of categorical columns is over now we have to scale our data because there are some columns present where values are much larger which will affect the runtime of the program so we will convert bigger values into smaller ones.

3.8 Parameter Tuning

Parameters are tuned using Grid searchCV. The parameters are tuned on Gradient Boost model.

3.9 Model Building

In this project, we are building the neural network, we will use TensorFlow and Keras library for building the artificial neural net.

3.10 Model Saving

Model is saved using pickle library in ``.pkl`` format.

3.11 Github

The whole project directory will be pushed into the GitHub repository