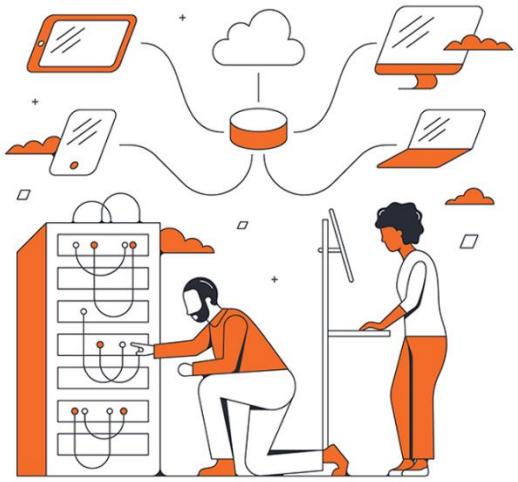
CREDIT CARD DEFAULT PREDICTION

: ARCHITECTURE DESIGN DOCUMENTATION :



CREATED BY

Rakesh

Document Version Control

Date	Version	Description	Author
25-July-2022	1.0	Abstract Introduction Architecture	Rakesh
26-July-2022	1.1	Architecture Design	Rakesh

Table of Contents

Abstract

Financial threats are displaying a trend about the credit risk of commercial banks as the incredible improvement in the financial industry has arisen. In this way, one of the biggest threats faced by commercial banks is the risk prediction of credit clients. The goal is to predict the probability of credit defaults based on credit card owner's characteristics and payment history.

1. Introduction:

1.1 Why this Architecture Design Document ?

The main objective of the Architecture design documentation is to provide the internal logic understanding of the Credit card defaults code. The Architecture design documentation is designed in such a way that the programmer can directly code after reading each module description in the documentation.

1.2 Scope

Architecture Design (AD) 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. And the complete workflow.

2. Dataset Information:

ID: ID of each client

LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary = credit)

SEX: Gender (1=male, 2=female)

EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown) MARRIAGE: Marital status (1=married, 2=single, 3=others)

AGE: Age in years

PAY_1: Repayment status in September 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)

PAY_2: Repayment status in August 2005 (scale same as above)

PAY_3: Repayment status in July 2005 (scale same as above)

PAY_4: Repayment status in June 2005 (scale same as above)

PAY_5: Repayment status in May 2005 (scale same as above)

PAY_6: Repayment status in April 2005 (scale same as above)

BILL_AMT1: Amount of bill statement in September 2005 (NT dollar)

BILL_AMT2: Amount of bill statement in August 2005 (NT dollar)

BILL_AMT3: Amount of bill statement in July 2005 (NT dollar)

BILL_AMT4: Amount of bill statement in June 2005 (NT dollar)

BILL_AMT5: Amount of bill statement in May 2005 (NT dollar)

BILL_AMT6: Amount of bill statement in April 2005 (NT dollar)

PAY_AMT1: Amount of previous payment in September 2005 (NT dollar)

PAY_AMT2: Amount of previous payment in August 2005 (NT dollar)

PAY_AMT3: Amount of previous payment in July 2005 (NT dollar)

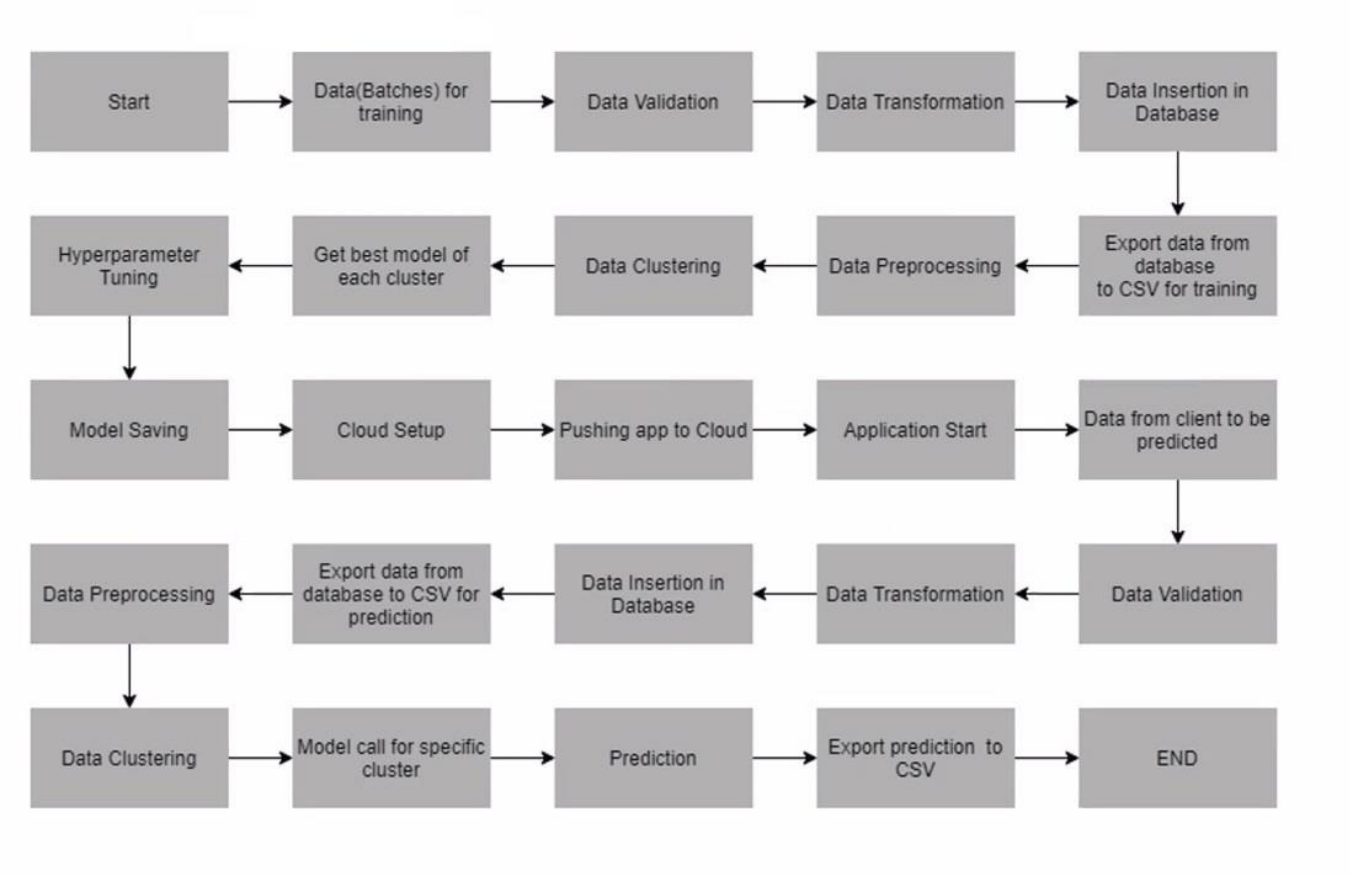
PAY_AMT4: Amount of previous payment in June 2005 (NT dollar)

PAY_AMT5: Amount of previous payment in May 2005 (NT dollar)

PAY_AMT6: Amount of previous payment in April 2005 (NT dollar)

Default payment next month: Default payment (1=yes, 0=no)

3. Architecture:



4. Architecture Design

4.1 DATA INGESTION

4.1.1 Data Validation

In this step we will do validation of data based on discussion with stack holder. In validation we will perform basic operations like file name validation, no of rows, no of columns in dataset etc.….

4.1.2 Data Transformation

In this step we need to transform data in order to store in our database. Basic operations like replacing nan values with null etc.…

4.1.3 Data insertion

After completion of data validation and transformation we need to insert our data into database. This step is required because we receive multiple batch files from user so that we need to consolidate those files.

4.2 CORE ML PIPELINE

4.2.1 Importing csv file

After insertion of data into database, now we need retire the data as csv file from database.

4.2.2 Data Preprocessing

After importing csv file we need to start our machine leaning pipeline. As we know first step in ml pipeline is data preprocessing. In this we will convert raw data into good data.

4.2.3 Data Clustering

This step is required because when we train our data with single model, we have chance of getting high error. Instead of that creating multiple models will give better accuracy and less error compare with one model.

4.2.4 Hyper parameter tunning

As we know that we perform hyper parameter tunning to file best parameters for model. This step is required to create good model.

4.2.5 Model Saving

This is final step of ml pipeline. We need to save our model for future predictions.

4.3 DEPLOYMENT

4.3.1 Pushing code to github

To deploy our application on cloud first we need to push our local code into github repository**.**

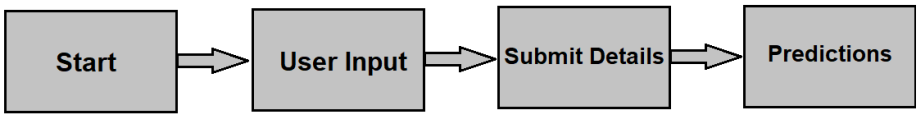
4.3.2 Deploy using cloud

After pushing code repository into github. We can use any cloud deploy platforms to deploy our application.

4.4 PREDICTIONS

After successful deployment of application. When ever the data comes form user, we need to perform all steps that are mentioned above and finally get predictions by using trained model.

5. User Input / Output Flow:



6. Conclusion:

The project is designed in the flask; hence it is accessible to everyone. The above design process will help banks and loan lenders predict whether customers will default the credit card payment or not, so the bank or respective departments can take necessary action, based on the model's predictions. The UI is made to be user-friendly so that the user will not need much knowledge of any tools but will just need the information for results

THE END